International Journal of Industrial Engineering, 30(2), 336-349, 2023

ACO-GCN: A FAULT DETECTION FUSION ALGORITHM FOR WIRELESS SENSOR NETWORK NODES

Huamin Chen[1, *] and Limin Ren[2]

[1]School of Information Engineering
Nanyang Institute of Technology
Nanyang China

[2]School of Intelligent Manufacturing
Nanyang Institute of Technology
Nanyang, China
[*]Corresponding author's e-mail: hmchen1214@163.com

Wireless Sensor Network (WSN) has become a solution for real-time monitoring environments and is widely used in various fields. A substantial number of sensors in WSNs are prone to succumb to failures due to faulty attributes, complex working environments, and their hardware, resulting in transmission error data. To resolve the existing problem of fault detection in WSN, this paper presents a WSN node fault detection method based on ant colony optimization-graph convolutional network (ACO-GCN) models, which consists of an input layer, a space-time processing layer, and an output layer. First, the users apply the random search algorithm and the search strategy of the ant colony algorithm (ACO) to find the optimal path and locate the WSN node failures to grasp the overall situation. Then, the WSN fault node information obtained by the GCN model is learned. During the data training process, where the WSN fault node is used for error prediction, the weights and thresholds of the network are further adjusted to increase the accuracy of fault diagnosis. To evaluate the performance of the ACO-GCN model, the results show that the ACO-GCN model significantly improves the fault detection rate and reduces the false alarm rate compared with the benchmark algorithms. Moreover, the proposed ACO-GCN fusion algorithm can identify fault sensors more effectively, improve the service quality of WSN and enhance the stability of the system.

Keywords: Wireless Sensor Network; Fault Detection; Ant Colony Algorithm; Graph Convolution Network; Location

(Received on December 23, 2022; Accepted on March 15, 2023)

1. INTRODUCTION

With the never-ceasing development and progress of microprocessor technology, using wireless sensor networks (WSN) to monitor environmental information has become a new and popular paradigm. At present, wireless sensor network, as the data collection port of information processing systems, has been widely used in various fields such as military defense, disaster warning, and intelligent agriculture (Kandris *et al*., 2020; Priyadarshi *et al*., 2020). Since the sensor usually contacts the external environment directly, it is common that it cannot work normally. However, if the system fails to detect the faulty sensor nodes in time, information will be lost or misreported, resulting in unpredictable consequences such as decision-making errors, instrument damage, and so on. Therefore, it is very important to detect whether the wireless sensor network node works normally (Osman, 2022; Lo *et al*.,2013). Sensor failures can be divided into hard and soft failures (Wei *et al*., 2022). Hard failure refers to the failure in the hardware facilities of the sensor nodes, which makes them unable to obtain information or communicate with other nodes. Soft failure means that although the sensor is still functioning, the information obtained or transmitted by the nodes is inaccurate. Because the sensor of soft fault is in an unstable working state, fault detection is relatively challenging, so the author mainly carries out fault detection for the soft fault (Kane *et al*., 2022; Chen *et al*., 2022).

The WSN is a self-organizing monitoring system composed of myriads of sensor nodes. Its main function is to monitor and process the information of the target area in real-time. Its main function is to monitor and process the information of the target area in real-time. In WSNs, integrated sensors are usually laid out to monitor, perceive, collect, and process network information for an area so that it can be completely sent back to users (Jabbari *et al*., 2021; Jaiswal *et al*., 2022). When WSNs transmit and process data, most of the WSN nodes are randomly distributed. WSN nodes are often deployed in uncontrollable harsh environments, and it is not easy for users to perform maintenance on deployed nodes (Feng *et al*., 2019; Michaelides *et al*., 2020). As a result, when individual nodes crash, packet loss, and routing failure occur, it is difficult for users to handle,

DOI: 10.23055/ijietap.2023.30.2.8801 ISSN 1943-670X

causing errors in data transmission. This will increase the energy loss of network nodes and the share of network bandwidth, and the frequency of network failures will increase. Therefore, it is of paramount importance to research and studies an effective and applicable fault diagnosis algorithm for node modules (Lee *et al*., 2018). It has important practical significance for extending the working time of WSN and maintaining the stable operation of the system.

Therefore, the WSN fault node detection method is studied more deeply. By putting forward ACO, finding the best path to locate WSN fault nodes, and applying the GCN model, the learning accuracy is further improved. This not only increases the detection accuracy but also provides support for the stable operation of the network. Overall, the main contributions of this paper are listed as follows. (I) By using the ant colony algorithm, users can locate the location of WSN nodes by looking for an optimal path. Through this random search algorithm and the search strategy of the ant colony algorithm, users can generally grasp the location of the faulty WSN nodes. (II) Then, based on the GCN network model, the obtained WSN fault node information is further learned. In the process of data training, the error is predicted according to the WSN fault node, and the weight and threshold of the network are further adjusted to increase the accuracy of fault diagnosis. (III) Finally, in the spatial-temporal processing layer, the graph convolutional network is used to extract the spatial distribution characteristics of wireless sensor networks and the fault characteristics in the high-dimensional space, and the high-dimensional data constructed as the sequential sequence is used as the input of the gated cycle unit. Then, the temporal evolution characteristics and spatial evolution characteristics of the sensor network data are extracted and fused through the gated cycle unit.

2. RELATED WORK

WSN is a self-organizing network based on opulent sensor nodes deployed in a specific area. The information transmitted between the mid-end and the end of the whole network is based on the relay nodes and is implemented by multi-hop forwarding, which includes the features of flexibility, dynamic, and distributed. It is widely used in disaster relief and environmental monitoring (Zhang *et al*., 2020). However, sensor nodes are more vulnerable to damage from external environments because they are usually randomly deployed in environments that are difficult for getting long-time maintenance. In addition, the low cost of manufacturing sensor nodes, sensor aging, and low battery power also causes the nodes to collect wrong data, which reduces the accuracy of monitoring data. Therefore, the research on fault detection of the wireless sensor network node is very important to ensure the good operation of WSN (Jurado *et al*., 2021).

According to the different data processing methods, wireless sensor network fault detection can be divided into three types: centralized, distributed, and mixed (Aslam *et al*., 2019). A centralized fault detection algorithm has high detection accuracy. However, for large-scale wireless sensor networks, the applicability is lower, which is mainly because the communication between sensor nodes and sink nodes or base stations is implemented in a multi-hop way where a large amount of energy consumption will occur on the routing nodes. Besides, this method has a high diagnostic delay and cannot detect the status of the nodes in real-time. Distributed fault detection determines the working condition of a node by collecting and analyzing diagnostic response results from neighboring nodes for each sensor node. When information is transmitted between nodes, it carries a low amount of information, which enables the network to maintain well and prolong its life. It also achieves an effective reduction of data latency between the ends of the network. Mixed detection combines the above two algorithms to keep the whole network with low data latency and high detection accuracy by adding additional mobile nodes. Research and development on fault detection in wireless sensor networks have achieved some theoretical results, most of which use related algorithms in machine learning. Huang *et al.* (2020) proposed a fault detection method for industrial IOT sensor devices based on a restricted Boltzmann machine and an automatic encoder. This method extracts data features by constructing several Boltzmann machine modules, corrects parameters in the automatic coder, and constructs an optimal fault diagnosis model by inputting training data. Xiao *et al.* (2020) summarized the advantages of rough set theory and artificial neural network.

Then, a decision diagnosis table is obtained by data compression to extract the fault feature set and construct a back-end radial basis function neural network. Finally, the non-linear mapping relationship between fault types and fault features is constructed to further detect and distinguish the fault types. Zidi *et al.* (2018) pointed out that there is an improved distributed fault diagnosis method, Cross-Spatial Distributed Diagnosis (CS-DFD). In this way, a high-dimensional vector space model is built from the information collected at each node. At the same time, the cross-sliding window is established by the data in the sliding window and that of adjacent nodes, the horizontal fault weight is set, the anomaly vectors and threshold are detected to achieve fault diagnosis, and the accuracy of fault detection is improved. Zhang *et al.* (2018) used a Bayesian classifier to predict the probability of sensor nodes failing. By using boundary nodes to adjust the failure probability, the effect of failure nodes on detection accuracy is avoided.

With the iterative update of algorithms, metaheuristic algorithms have been gradually used in this field in recent years. Because of its non-linear solving method, this kind of algorithm has a good application effect in solving complex structure problems (Agarwal *et al*., 2022). Lin *et al.* (2021) put forward a method that combines an improved flower pollination algorithm (IFPA) with a support vector machine. To improve the global search ability of flower pollination algorithm, parallel operation is introduced in the paper. By dividing the population into groups, using improved mutation and cross-propagation strategies, better pollens are changed to refined pollens, and IFPA is used to optimize the parameters of the kernel function in support vector machines, thus achieving a more accurate detection result. Hoyingcharoen *et al.* (2019) proposed a fault detection algorithm that combines an improved multivariate cosmic optimization algorithm (MVO) with a feed-forward neural network (FNN). To enhance the population diversity of the MVO algorithm, the population is divided into subgroups for parallel operation, and the population diversity is enhanced by improving the communication mechanism. While the iteration reaches the threshold, the universe of different groups will exchange information, which further reduces the occurrence of search stagnation and increases the probability of jumping out of the local optimal solution. The improved MVO algorithm is used to optimize the parameters in FNN, which improves the classification accuracy of the FNN algorithm. Peng *et al.* (2022) synthesized the advantages of the algorithm and used the improved crow search algorithm and machine learning classifier to classify the failures. By comparing this algorithm with three different machine learning algorithms, it is found that the performance of this algorithm is better than other classifiers. Considering the characteristics of WSN node fault detection, a detection algorithm based on the combination of firefly optimization and extreme learning machine is presented in this paper. Firstly, based on the characteristics of the Levy algorithm in short-distance random walks and sudden long-distance jumps, the standard Firefly Optimization Algorithm (FA) is improved. It greatly reduces the probability that the objective function will fall into a local optimal solution. In addition, it improves the population convergence rate and achieves a balance between development and exploration capabilities. A target function for fault data detection in WSN is constructed using an extreme learning organization, and the target function is optimized using an optimized firefly optimization algorithm. In this way, the parameters of the extreme learning machine can be indirectly modified to achieve higher accuracy of fault data detection.

To the best of our knowledge, there are no multi-objective fault detection proposals that focus on the input layer, space-time processing layer and output layer, which are implemented in the WSN. Therefore, this paper presents fault detection method for the WSN via applying ACO-GCN models.

3. WIRELESS SENSOR NETWORK ARCHITECTURE

3.1 WSN modeling

Sensors in wireless sensor networks are usually distributed in an irregular network structure. Such a structure can be modeled as an unoriented graph $G = (\Phi, E, H)$ (Li *et al.*, 2021). The edges between sensors represent the connection between them.$\Phi = \{\varphi_1, \varphi_2, \cdots, \varphi_n\}$ represents the collection of all sensor nodes, $\varphi_i (i = 1,2, \cdots, n)$ represents the sensor node named with serial number i. $E = \{e_{i,j}\}$ represents the set of edges in the graph model and $e_{i,j}$ represents that there are edges connected between φ_i and φ_j. $H \in \Re^{n \times n}$ represents the adjacency weight matrix of a graph, where $H_{i,j}$ represents the edge weight between φ_i and φ_j. In addition, the graph can also represent $L = D - H$ by the Laplace matrix, where $D = \text{diag}\{d_i\}$ represents the degree matrix, $d_i = \sum_j H_{i,j}$. The normalized Laplacian matrix can be expressed as $L_s = \Pi - D^{-\frac{1}{2}} H D^{-\frac{1}{2}}$ where $\Pi \in \Re^{n \times n}$ represents the identity matrix. The eigenvalue matrix $\Lambda_s = \text{diag}\{\lambda_i\} (\lambda_1 \leq \lambda_2 \cdots \leq \lambda_n)$ and the eigenvector matrix U_s corresponding to the eigenvalue are obtained via eigen decomposition of $L_s = U_s \Lambda_s U_s^T$.

The shortest path between the node φ_i and φ_j is defined as geodesic distance ρ_{ij}. If $\varphi_i$has no access to φ_j, then $\rho_{ij} = \infty$. The k -order neighbor definition of φ_i can be obtained by geodesic distance $B(i,k) \equiv \{j | \rho_{ij} \leq k\}$, that is, the node set whose geodesic distance from φ_i is less than or equal to k. The data collected by the sensor network at different times have certain differences. Therefore, the data collected by the sensor network in T of consecutive moments can be expressed as $X = [x_1, x_2, \cdots x_T]$. Among them, $x_T = [x(\varphi_1), x(\varphi_2), \cdots, x(\varphi_n)]$ represents the collection of data collected by each node of the sensor network at the time of t. x_i^t represents the data collected by the sensor node φ_i at the time of t.

Through the construction graph model, the problem of wireless sensor network fault detection can be described in the following ways. A sensor network consisting of $n$sensors is deployed in a certain area. The data collected by the sensor network in T consecutive times is X. x_t represents the elements collected by the sensor network at the time t, that is, the set of signals to be detected. $[x_{t-T+1}, x_{t-T+2}, \cdots, x_{t-1}]$ represents the historical signal set of the sensor network. Analyze X and graph model$G$of the sensor network by building a fault detection model to detect whether fault nodes exist at x_t. If there is a fault node, the corresponding fault sensor will be further located.

3.2 WSN node fault detection system

This paper proposes a WSN node fault detection technology that combines ACO with the GCN model. It can detect WSN node fault information in an all-around way, which will make the detection of WSN node information more accurate and reliable. The framework of this scheme mainly includes a data layer, an information detection layer, and a system application layer. The system architecture is shown in Figure 1.

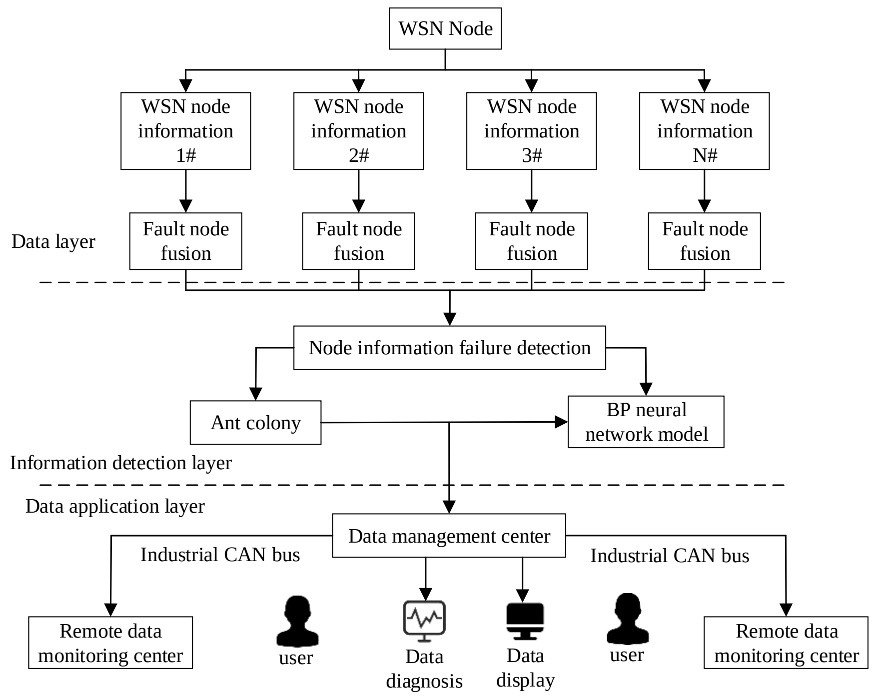

Figure 1. WSN Node Fault Detection System

In this system design, there are many sensor nodes in WSN. However, the more nodes there are, the greater the likelihood of failure. Therefore, it is necessary to fuse sensor node information before detecting node failures. Through the fusion of fault information, users can grasp the WSN node information more comprehensively and then calculate and process the fused WSN node information through the algorithm designed in this research. ACO is used to estimate and locate sensor nodes, and the GCN model is used to further study wireless sensor nodes. To make full use of the spatial and temporal characteristics of sensor networks, firstly, the spatial correlation of wireless sensor networks is depicted by the graph model, which is followed by the spatial characteristics of sensor networks extracted by GCN. Secondly, the spatial features extracted by the convolution network are fused with the data collected by the wireless sensor network and sent to the GRU. In GRU, the time characteristics of wireless sensor networks are extracted by making appropriate rounding of the information from the previous moment. Finally, the space-time characteristics are fitted to the fault detection results by the full connection layer. In this system design, the data obtained can be used in the data management center for user diagnosis and data display purposes. It can also be remotely transmitted to the remote data monitoring center through the industrial CAN bus for management by a higher-level monitoring center.

3.3 Problem description

WSN collects monitoring information under the auspice of a wide range of sensor nodes and studies the problem of WSN fault detection. The initial step is to detect and analyze whether the WSN node is functioning properly. Due to disturbances, if we detect data fluctuations with anomalies, we can conclude that the WSN node may fail. Define as follows to accomplish the above purpose.

$$A_n^m(t) = \begin{bmatrix} a_1^1(t) & a_1^2(t) & \cdots & a_1^n(t) \\ a_2^1(t) & a_2^2(t) & \cdots & a_2^n(t) \\ \vdots & \vdots & \vdots & \vdots \\ a_m^1(t) & a_m^2(t) & \cdots & a_m^n(t) \end{bmatrix} \tag{1}$$

$$y(t) = F\left[A_n^m(t), R\right], \tag{2}$$

where $A_n^m(t)$ defines the system input, $y(t)$ describes system output, $A_n^m(t)$ represents a data set used to detect whether there is a failure, and $y(t)$ defines the result of the failure detection.

The fault detection is mainly to analyze the data collected by WSN to determine whether the node is working properly. Users can identify possible flawed data based on the time-space correlation of WSN data. Due to noise disturbance when WSN collects monitoring data, this will make the collected data not fully trusted. This problem was solved by building BRB-r, where the function F is the process of converting the input dataset to the output detection result, and R is the set of parameters in the process of conversion (Hu *et al*., 2020).

4. FAULT DETECTION FUSION ALGORITHM

Figure 2 is a schematic diagram of a WSN node failure detection method. In this method, the ACO algorithm has great advantages in dealing with discrete combinatorial optimization problems. In this method, the ACO algorithm has great advantages in dealing with discrete combinatorial optimization problems. Therefore, after searching WSN nodes with ACO, the GCN model is used for detection, which can improve the accuracy of fault detection. When the fault node location is completed, the GCN model is used to detect the fault nodes in the wireless sensor network. As shown in Figure 2, the GCN model mainly consists of three parts: input layer, space-time processing layer and output layer. The input layer receives the sensor network data X and the adjacency matrix H of its graph model as an input of the space-time processing layer. The space-time processing layer consists of a GCN module and a GRU. First, the GCN module receives data from the input layer and extracts the spatial characteristic S of the sensor network data through a graph convolution operation. The data of S and X are joined together and infused into GRU, and the space-time feature Ω of sensor network data is extracted and transmitted to the full connection layer using GRU. Finally, the space-time feature Ω is fitted to the probability matrix Y through the full connection layer and transferred to the output layer, where the elements of the probability matrix Y represent the classification probability of the corresponding nodes. The output layer converts Y into fault detection results and outputs them.

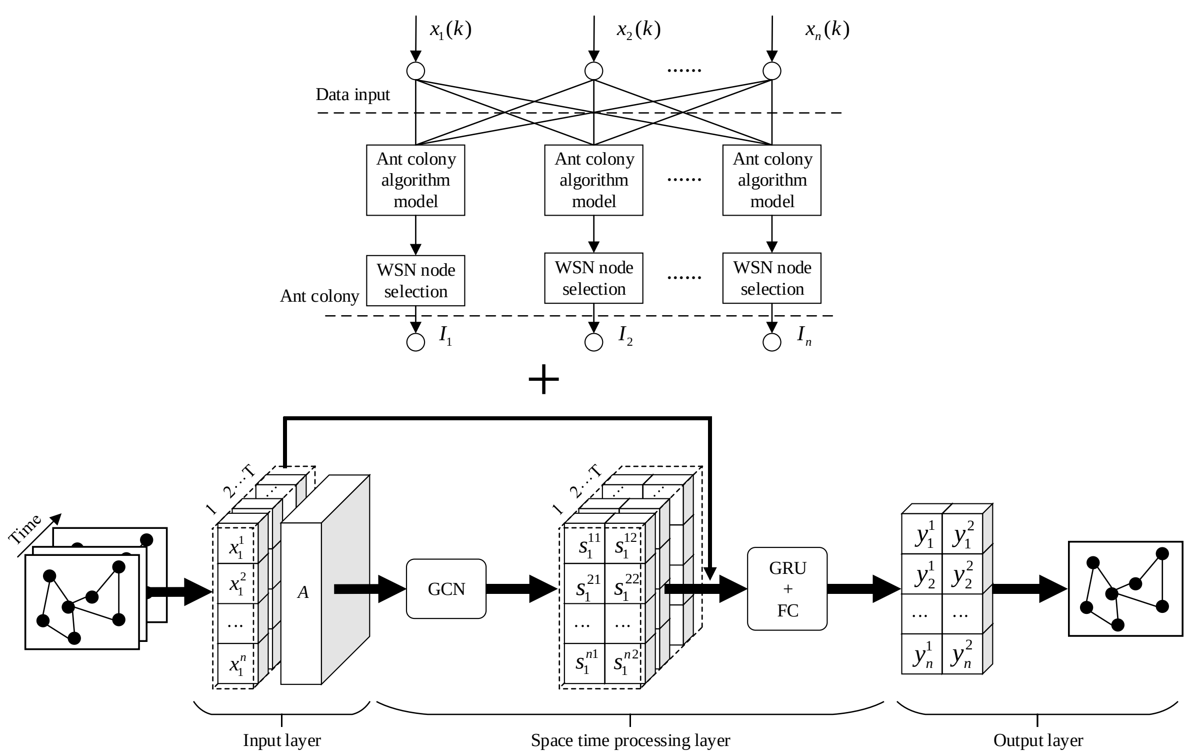

Figure 2. WSN Node Failure Detection Method

4.1 Ant colony algorithm model

ACO algorithm was first proposed by Italian scholar Liu *et al*., which is a global optimization algorithm (Liu *et al*., 2022). According to the theory of the algorithm, if we set the number of WSN nodes to N and the number of unknown nodes to M, the function expression is

$$F\left(x_1, y_1, \cdots, x_i, y_i, \cdots, \cdots, x_j, y_j, \cdots, x_M, y_M\right) = \sum_{i=1}^{M} \sum_{j\in N_i} \left(\varphi_{ij}^{'} - \varphi_{ij}\right)^2. \tag{3}$$

In Equation (3), (x_i, y_i) represents the estimated coordinates of the unknown node$i$, N_i defines the set of neighbor nodes of node i, $j\left(x_j, y_j\right)$ is the neighbor node of node i, and $j$is the unknown node or anchor node. Assuming$j$an unknown node $\left(x_j, y_j\right)$ can be calculated as an estimated coordinate. If$j$is an anchor node, then $\left(x_j, y_j\right)$ is its true coordinate. Then the Equation of $d_{ij}^{'}$ is

$$d_{ij}' = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}\ , \tag{4}$$

where d_{ij}' is the estimated distance calculated from the estimated coordinates of nodes i and j, and d_{ij} represents the measured distance between nodes i and j. If the estimated coordinates of an unknown node i are $\left(x_i(t), y_i(t)\right)$ at the time $t(t = 0,1, \cdots, t)$, the coordinates of M unknown nodes can be expressed as a matrix $D(t)$.

$$D(t) = \begin{pmatrix} x_1(t) & x_2(t) & \cdots & x_M(t) \\ y_1(t) & y_2(t) & \cdots & y_M(t) \end{pmatrix}. \tag{5}$$

Each set of column vectors in $D(t)$ corresponds to an estimated coordinate of an unknown node. When locating WSN nodes, the matrix $D(0)$ is transformed from $D(1)$ to $D(t)$ based on the state.

$$[x_i(t+1) y_i(t+1)]^T = [x_i(t) y_i(t)]^T + \lambda(t) \cdot v_s \tag{6}$$

$$\lambda(t+1) = \xi\lambda(t) \tag{7}$$

$$v_s = \begin{cases} [0,0]^T, \quad s = 0 \\ \left[cos\left(\frac{2s\pi}{S}\right) \cdot sin\left(\frac{2s\pi}{S}\right)\right]^T, \ s = 1,2, \cdots, S \end{cases}. \tag{8}$$

Equations (6), (7), and (8) give the possible direction of node transfer, respectively where $\left(x_i(t+1), y_i(t+1)\right)^T$ and $\left(x_i(t), y_i(t)\right)^T$ represent the estimated node at $t+1$ and t, respectively, and they belong to the group i column vectors of $D(t+1)$ and $D(t)$. $\lambda(t)$ represents the step that the node moves at the time t. ξ is the step decay factor. The step size decreases over time until it finally approaches zero. v_s represents the choice of the direction in which node i moves, either without moving or along one of the S directions in which the plane S is divided. Then the probability function for ants to choose direction is

$$P_{iv_s} = \frac{\tau_{iv_s}}{\sum_{s=1}^{S} \tau_{iv_s}}. \tag{9}$$

The pheromone update rules for ant selection direction can be defined as follows.

$$\tau_{iv_s}(t+1) = \begin{cases} \rho \cdot \tau_{iv_s}(t) + Q \\ \rho \cdot \tau_{iv_s}(t) \end{cases}. \tag{10}$$

In Equation (10), it demonstrates that if v_s is the direction chosen by the ant element with the smallest objective function value. The rest are other cases, and pheromones are representations that depend on the direction in which an ant element moves. Among the Equation P_{iv_s} represents the probability that node i transfers in the direction of v_s. τ_{iv_s}denotes the pheromone concentration of the node i in the v_s direction. ρ represents the persistence of a pheromone, and Q is a constant that represents the contribution of the ant with the smallest objective function to the moving direction of the node.

At moment $t = 0$, the unknown node estimation coordinate matrix $D(0)$ is initialized in a random manner, and the pheromone constants in all directions are assigned initial values. When m ants traverse all unknown nodes, the estimated coordinates for the next moment are determined by Equation (9). The objective function is calculated from the estimation matrix formed by each ant element. The estimation matrix of the ant element with the smallest objective function is selected as the new estimation matrix, and the pheromone concentration is changed according to Equation (10). Repeat the above steps.

Then, the trilateral measurement method is applied to the location estimation of WSN nodes. As shown in Figure 3, assume that D is the unknown node, A, B, and C are the three anchor nodes within the communication range of D, whose coordinates are (x_1, y_1), (x_2, y_2), (x_3, y_3), respectively. The distances from A, B, C to D are d_1, d_2, d_3.

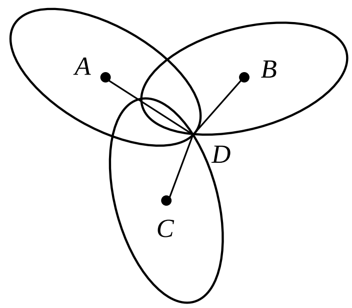

Figure 3. Schematic Diagram of the Trilateral Measurement Method

Set coordinates of D to (x, y), and set the column equations as follows.

$$\begin{cases} \sqrt{(x-x_1)^2+(y-y_1)^2} = d_1 \\ \sqrt{(x-x_2)^2+(y-y_2)^2} = d_2 \\ \sqrt{(x-x_3)^2+(y-y_3)^2} = d_3 \end{cases} . \tag{11}$$

Solve the equations

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2(x_1-x_3) & 2(y_1-y_3) \\ 2(x_2-x_3) & 2(y_2-y_3) \end{bmatrix}^{-1} = \begin{bmatrix} x_1^2 - x_3^2 + y_1^2 - y_3^2 + d_3^2 - d_1^2 \\ x_2^2 - x_3^2 + y_2^2 - y_3^2 + d_3^2 - d_2^2 \end{bmatrix} . \tag{12}$$

The result is the coordinates of the unknown node$D$. Therefore, the location of WSN nodes can be achieved by using the ant colony algorithm model to obtain the location of WSN nodes.

4.2 Graph convolution network model

The GCN module receives the adjacent matrix $H$of the sensor network data X and its graph model and extracts the spatial characteristics through the graph convolution operation. Graph convolution is defined as the product of filter $g_\theta = diag(\theta)$ and graph signal $x \in \Re^{n\times n}$ in the frequency domain, as shown in Equation (13).

$$g_\theta * x = U_s g_\theta U_s^T x . \tag{13}$$

Among the term$*$ represents the convolution of a graph. The computational complexity of Equation (13) is $O(N^2)$, so the amount of operation of Equation (13) increases significantly with the increase in the dimension of data used. Therefore, it is necessary to use methods in the document to fit g_θ through a second-order Chebyshev polynomial to reduce computational complexity (Bianchi *et al.*, 2022), that is:

$$g_\theta * x \approx c_0 x - c_1 D^{-\frac{1}{2}} H D^{-\frac{1}{2}} x = c\left(I_N + D^{-\frac{1}{2}} H D^{-\frac{1}{2}}\right) x , \tag{14}$$

where $c = c_0 = -c_1$, while c_0 and c_1 represent the coefficients of the Chebyshev polynomial. There are two different situations that may occur when Equation (14) is iterated: gradient explosion and gradient. Gradient explosion (gradient disappearance) refers to the phenomenon that the updating result increases (decreases) rapidly with the number of iterations when the parameter gradient is calculated by the inverse propagation algorithm, which results in slow convergence of the neural network and other issues. Thus, $I_N + D^{-\frac{1}{2}}HD^{-\frac{1}{2}}$ is normalized to $\widetilde{D}^{-\frac{1}{2}}\widetilde{H}\widetilde{D}^{-\frac{1}{2}}$ to mitigate the effects of gradient explosion or gradient disappearance, resulting in the definition of a convolution layer as follows

$$h = \widetilde{D}^{-\frac{1}{2}}\widetilde{H}\widetilde{D}^{-\frac{1}{2}}xw^T \ , \tag{15}$$

where $\widetilde{H} = H + I_N$ represents the sum of the adjacent matrix H and the unit matrix I_N, $\widetilde{D}$ represents the degree matrix of H, and w^T represents the weight vector.

According to the definition of Equation (15), the forward propagation equation in the GCN module is

$$S_t = f_{GCN}\left(x_t, \widetilde{H}_s\right) = \widetilde{H}_s\left(\widetilde{H}_s x_t w_0^T\right)W_1 \ , \tag{16}$$

where $\widetilde{H}_s = \widetilde{D}^{-\frac{1}{2}}\widetilde{H}\widetilde{D}^{-\frac{1}{2}}$ means normalizing $\widetilde{H}$. w_0^T and W_1 represent the weight vectors of the first convolution layer and the weight matrices of the second convolution layer, respectively. $S_t \in \Re^{n\times 2}$ represents the spatial characteristics of the graph signal at t time.

Equation (16) enables each node in the graph to take advantage of its second-order neighbors by using a two-layer convolution layer. This makes full use of the spatial correlation of wireless sensor networks. As shown in Figure 4 (c). The shadow nodes in Figure 4 represent the center nodes, and the black nodes represent the k -order neighbors of the center nodes.

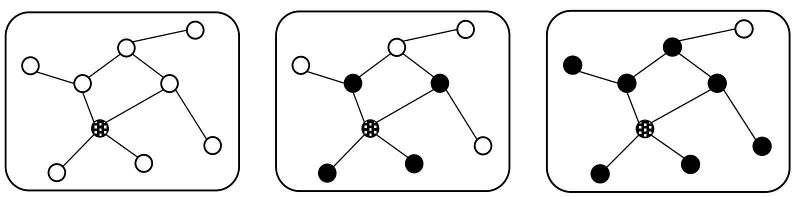
(a)k=0 (b)k=1 (c)k=2

Figure 4. GCN Aggregation Neighbor Node Diagram

4.3 Flowchart of fault detection fusion algorithm

In order to verify the reliability of the ACO-GCN algorithm, each node in WSN regularly sends routing information and data collection to the root node, and the root node connects to the upper computer through the USB serial port to transmit the received routing information and data to the upper computer. The fault analysis module saves the routing information and data transmitted by the root node to the database in real time. At the same time, the algorithm can draw the topology structure of the whole network according to the received routing information and calculate the success rate of data transmission according to the received serial number. In addition, the key steps of the proposed ACO-GCN fusion algorithm are performed, as shown in Figure 5.

(1) Initialize the unknown node coordinate matrix, set the initial values of each parameter, assume T is the number of iterations and set the initial values of each parameter.
(2) Set the concentration of pheromone $\tau_{iv_s} = C$for all elements in the matrix (τ_{iv_s}) $M^*(S+1)$where$C$is a constant representing the initial value of the pheromone concentration.
(3) Traverse the ant element through all unknown nodes. The direction in which each unknown node moves next is chosen by Equation (9).
(4) Calculate the objective function value obtained by each ant, and select the smallest ant's estimation matrix as the unknown node's estimation matrix.
(5) Update the pheromone concentration according to Equation (10), and then update λ according to Equation (7).
(6) Determine whether the end condition is met (i.e., whether T is greater than the maximum number of iterations or if the solution found by the ants is the same). End iteration if satisfied; otherwise, jump to Step 3.
(7) The result of the output solution.

With the above calculations, as the first experiment. Then, using the above methods, the BP network model is used individually to experiment, and the error data is obtained as the second experiment. The ACO fusion algorithm was used alone as the third experiment. Below are three different forms of error data representations represented by a graph.

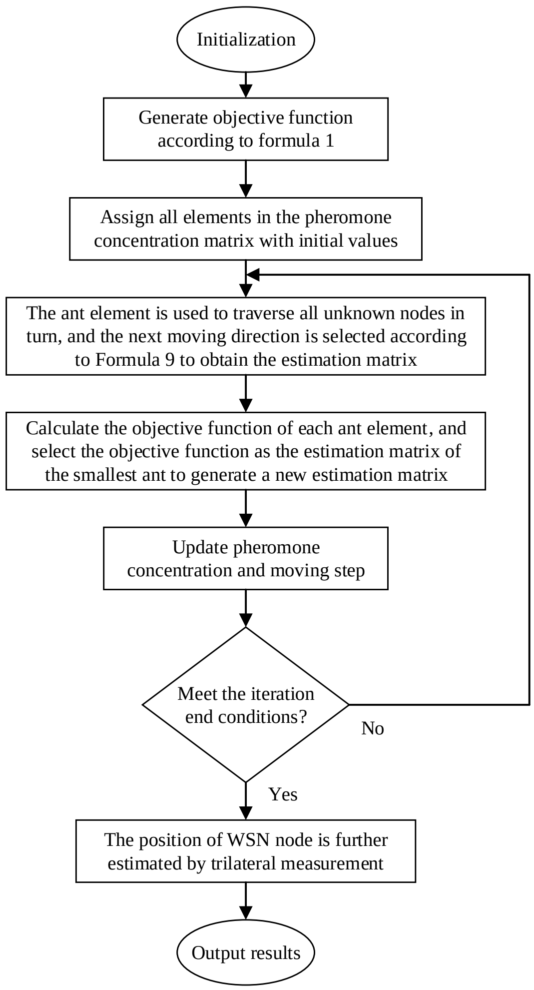

Figure 5. Flow Diagram of the ACO-GCN Fusion Algorithm

5. RESULTS AND DISCUSSIONS

5.1 Experimental datasets

The software of this experiment uses Windows 10 and a 64-bit operating system. In addition, Python is used as a programming tool. The interior 58250U processor is selected as the hardware part coupled with 8G memory. The experimental data come from the Berkeley Laboratory Data Set (IBRL), and 60% of the data are used for model training and 40% for model performance testing. In order to simulate the actual failure situation, set the time of the failure sensor in the training set to 10% of the total number and the time of the failure sensor to 5% of the total number. When a fault occurs, the number of fault sensors is 5% of the total number of sensors. Whether a fault occurs at the current moment or not, it needs to be detected using the first three consecutive moments of anomaly-free data. If the network operates for the same period of data transmission, nodes consume the same energy to forward data. The data for this experiment come from the wireless sensor network deployed by InterBrkeleey Lab (Suthaharan *et al.*, 2010). The number of MICA2 sensor nodes is 54. Sampling tests for ambient data are performed every 30 seconds, and the time interval for node data collection is from February 28, 2004, to April 5, 2004. Data collection through nodes mainly covers four attributes: temperature, light intensity, node voltage, and temperature. In the experiment, the nodes of the sensor are set to 6, 13, 25, 37, and 49. By analyzing the time stamps in the samples, it is found that the locations where the anomalies occur are in the sample data of different periods that are randomly inserted. The collected datasets are divided into two parts, one is a test set, and the other is a training set. The dataset is divided into a test set and a training set. In subsequent experiments, the number of four different fault data will be increased from 0 to 5% to achieve the comparison of different fault detection results. The experimental data are shown in Table 1.

Table 1. Experimental Data Set

Data set	Number of training samples	Number of exceptions	Number of test samples	Number of attributes
IBRL-6	7000	500	7000	1
IBRL-13	7000	500	7000	1
IBRL-25	7000	500	7000	1
IBRL-37	7000	500	7000	1
IBRL-49	7000	500	7000	1

5.2 Parameter settings

Python has been used as the programming language in this experiment, assuming that the gross number of nodes is 100, and distributing them in a square area of $10L * 10L$, where L is the edge length of the area, $L = 5m$ in this experiment. Anchor nodes and unknown nodes are randomly distributed in a square area. In WSN, the communication radius of nodes is 15m, and the network connectivity is 10.2. Then set the relevant parameters, assuming the number of ants $N = 20$, the direction of movement $(S + 1) = 9$, and the initial step length $\lambda = 1$, step decay factor$\zeta = 0.96$, pheromone persistence$\rho = 0.9$, and$Q = 5$stands for the smallest ant contribution to the pheromone of the objective function, and 150 stands for the largest iteration number. As is shown in Table 2.

Table 2. Parameters Setting

Symbol	Description	Value
N	Total number of nodes	100
S	Size	50*50
R	Communication radius	15m
η	Network connectivity	10.2
N_c	Number of ant colonies	20
λ	Initial step size	1
ξ	Step size attenuation coefficient	0.96
ρ	Pheromone persistence	0.9
Q	Pheromone contribution	5
N_{max}	Maximum number of iterations	150
Ψ	Optimizer	RMSprop
$L(y, \hat{y})$	Loss function	Cross entropy loss function
N_{tri}	Number of trainings	200
τ	Rate of learning	0.01

In this experiment, Python is chosen as the programming language, Pytorch 1.6 is used to build the network model, and RMSprop optimizer is used to optimize the model parameters. RMSprop is a gradient-based loss function optimization algorithm for neural networks which has obvious advantages in reducing update fluctuation range and accelerating convergence speed and is an excellent optimizer in many cases. The cross-entropy loss function has strong applicability in classification, so it is used as the loss function in the training process, as defined in Equation (17).

$$L(y, \hat{y}) = -\frac{1}{N}\sum_{n=1}^{N} y_n \, log_2^{\hat{y}_n} \; + (1 - y_n)\, log_2^{1-\hat{y}_n} \quad , \tag{17}$$

where y is a one-hot coded label vector and y^* is a distribution probability. An appropriate learning rate allows the model to converge as quickly as possible without falling into a locally optimal solution. A better learning rate is selected by cross-validation.

The detection rate P_{DR}, failure detection rate P_{FDR}, and false alarm rate P_{FAR} are commonly used as indicators for evaluating sensor network failure detection results. Detection rate, failure detection rate, and false alarm rate can be expressed by real case N_{TP}, false positive N_{FP}, true and negative case N_{TN}, and false negative case N_{FN}. The fault conditions are regarded as positive examples, and the normal situations are regarded as negative examples. Then N_{TP} represents the number of positive examples with correct model classification. N_{FP} represents the number of positive examples of model classification errors. N_{TN} represents the number of negative examples with correct model classification. N_{FN} represents the number of negative instances of model classification errors. Thus, the definitions of P_{DR}, P_{FDR}, and P_{FAR} can be derived.

$$\begin{cases} P_{DR} = \frac{N_{TP}+N_{TN}}{N_{TP}+N_{FN}+N_{FP}+N_{TN}} \\ P_{FDR} = \frac{N_{TP}}{N_{TP}+N_{FN}} \\ P_{FAR} = \frac{N_{FN}+N_{FP}}{N_{TP}+N_{FN}+N_{FP}+N_{TN}} \end{cases} . \quad (18)$$

As can be seen from the definition, the detection rate refers to the proportion of the number of samples that the model correctly classifies into corresponding categories. Failure detection rate refers to the proportion of samples correctly classified as correct by the model to all positive samples. The false alarm rate represents the proportion of the number of samples with errors in the total number of negative samples when the model predicts positive samples.

5.3 Results and Analysis

Based on the experimental datasets and parameter settings, this work is accomplished according to the flowchart of the fault detection fusion algorithm, as shown in Figure 5. In Figure 6, the horizontal coordinate$X$represents time$t$, and the vertical coordinate$Y$ represents the calculation error data. The diagram shows three different ways of experimenting in t-time, where ▲ represents the error data obtained by using the ACO algorithm, ● represents the error data obtained by using the BP network model, ■ represents the error data obtained by using the ACO-GCN fusion algorithm. The calculation error data accordingly decreases with the increasing operation time$t$, in which the benchmark ACO algorithm decreases significantly due to the inherent slower convergence.

According to the comparison of the above curves, it is shown that the ACO-GCN algorithm of this research has a higher accuracy in obtaining error data, demonstrating that the algorithm mentioned in this research has less error, greatly improving the accuracy in determining fault nodes in WSN. The reason the ACO-GCN algorithm contributes to the lower calculation error data is that the fusion model consists of the input layer, space-time processing layer and output layer. More specially, the input layer receives sensor network data and graph models built by the wireless sensor network and firstly transmits them to the space-time processing layer. Then, in the space-time processing layer, the graph convolution network is used to extract the spatial distribution and fault characteristics of the wireless sensor network in high-dimensional space and construct the characteristics as time series as input of gated loop unit. Then the time and space evolution characteristics of sensor network data are extracted and fused by a gated loop unit. Finally, fault detection results are obtained at the output layer.

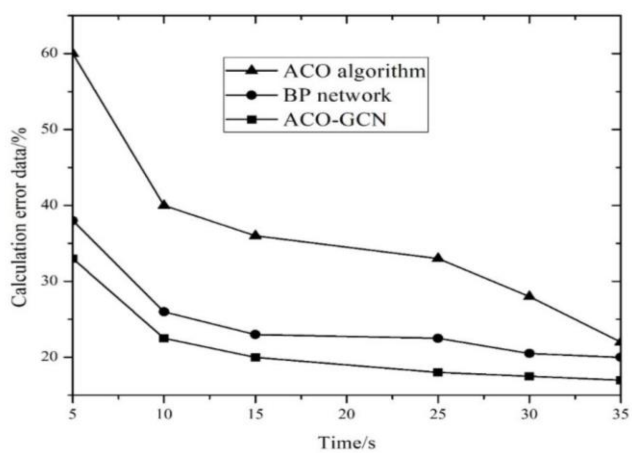

Figure 6. Comparison of Error Data Using Different Algorithms

In addition, the ACO algorithm [3], GCN [4] algorithm, and ACO-BP [5] algorithm are selected as benchmark algorithms, which can be used to evaluate sensor network fault detection results, and the simulation results are shown in Table 3. As can be seen from Table 3, compared with ACO, GCN and ACO-BP algorithms, the proposed ACO-GCN model has a higher detection rateP_{DR}, a higher failure detection rate P_{FDR}, and a lower false alarm rateP_{FAR}. Its overall detection performance has obvious advantages over the other three algorithms. It is owing to that the ACO-GCN model aggregates the second-order neighbor information of each sensor node through two convolution layers, making full use of the spatial correlation of wireless sensor networks. Therefore, compared with the GRU model that only uses time characteristics, the ACO-GCN model has obvious advantages in performance. Compared with DFD and NDFD algorithms, the ACO-GCN model extracts the time characteristics of sensor network data through GRU, which performs better in feature extraction of time sample data. Therefore, this model has a better performance when detecting whether there is a fault in the sensor. The ACO-GCN model has better fault detection performance than the comparison algorithm, which shows that it has better performance in extracting and utilizing space-time characteristics and can effectively handle the problem of fault sensor detection in the WSN.

Table 3. Evaluation Indicators of Each Algorithm Under Different Data Sets

Data set	Evaluating indicator	Algorithm			
		ACO/%	GCN/%	ACO-BP/%	ACO-GCN/%
Temperature data set	Detection rate P_{DR}	99.89	99.94	99.90	99.98
	Failure detection rate P_{FDR}	77.55	89.80	87.76	97.96
	False alarm rate P_{FAR}	0.11	0.60	0.10	0.02
Sea level pressure data set	Detection rate P_{DR}	99.85	99.92	99.84	99.97
	Failure detection rate P_{FDR}	78.15	85.14	63.91	95.03
	False alarm rate P_{FAR}	0.15	0.08	0.16	0.03

6. CONCLUSION

The ACO is applied to the location of wireless sensor nodes, and a WSN fault node location algorithm based on ACO is proposed, which can realize the optimal search of WSN fault nodes. Through the iterative operation, the ant elements on the search path can be uniformly gathered, and the path of the ant elements can be adjusted in a standard and dynamic way to realize the estimation of the selected path, update the pheromone, and quickly obtain the local optimal solution of the path. The spatial characteristics of the sensor network are extracted by GCN. Then the extracted spatial features are combined with the input signal as the input of GRU, and the space-time features are extracted by GRU. Finally, the space-time characteristics are fed into the full connection layer and fitted into the fault detection results. The simulation results show that the ACO-GCN model proposed by the author is superior to the ACO algorithm, GCN algorithm, and ACO-BP algorithm, which significantly improves the accuracy and reliability of WSN fault node detection and helps users to further study and analyze the WSN fault node events.

ACKNOWLEDGEMENT

This work was supported by the 2021 Higher Education Teaching Reform Research and Practice Project of Henan Province: Research on Construction and Practice of "Golden Lesson" of Electronic Design Automation Course Based on Online and Offline Hybrid Teaching 2021SJGLX540; First-class Undergraduate Courses in Henan Province [2022] 38156.

REFERENCES

Agarwal, V., Tapaswi, S., and Chanak, P. (2022). Intelligent Fault-Tolerance Data Routing Scheme for IoT-Enabled WSNS. *IEEE Internet of Things Journal*, 9(17): 16332-16342.

Aslam, N., Xia, K., and Hadi, M.U. (2019). Optimal Wireless Charging Inclusive of Intellectual Routing Based on SARSA Learning in Renewable Wireless Sensor Networks. *IEEE Sensors Journal*, 19(18): 8340-8351.

Bianchi, F.M., Grattarola, D., Livi, L., and Alippi, C. (2022). Graph Neural Networks with Convolutional ARMA Filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7): 3496-3507.

Chen, Z., Tian, H., and Shen, H. (2022). Improve The Quality of Charging Services for Rechargeable Wireless Sensor Networks by Deploying A Mobile Vehicle with Multiple Removable Chargers. *Wireless Networks*, 28(7): 2805-2819.

Feng, Y., Liu, H., Yang, J., and Fu, X.D. (2019). A Localized Inter-Actuator Network Topology Repair Scheme for Wireless

Sensor and Actuator Networks. *Communications China*, 16(2): 215-232.

Hoyingcharoen, P., and Teerapabkajorndet, W. (2019). Expected Probabilistic Detection and Sink Connectivity in Wireless Sensor Networks. *IEEE Sensors Journal*, 19(12): 4480-4493.
Hu, Q., Li, C., Lu, Y., and Li, S. (2020). A Novel Construction and Inference Methodology of Belief Rule Base. *IEEE Access*, 8: 209738-209749.

Huang, H., Ding, S., Zhao, L., Huang, H., Chen, L., Gao, H., and Ahmed, S.H. (2020). Real-Time Fault Detection for IoT Facilities Using GBRBM-Based DNN. *IEEE Internet of Things Journal*, 7(7): 5713-5712.

Jabbari, A., and Mohasefi, J.B. (2021). User-Sensor Mutual Authenticated Key Establishment Scheme for Critical Applications in Wireless Sensor Networks. *Wireless Networks*, 27(1): 227-248.

Jaiswal, K., and Anand, V. (2022). FAGWO-H: A Hybrid Method Towards Fault-Tolerant Cluster-Based Routing in Wireless Sensor Network for IoT Applications. *The Journal of Supercomputing*, 78(8): 11195-11227.

Jurado, F., Nirmalathas, A., and Clarke, K. (2021). Energy-Aware Routing for Software-Defined Multihop Wireless Sensor Networks. *IEEE Sensors Journal*, 21(8): 10174-10182.

Kandris, D., Nakas, C., Vomvas, D., and Koulouras, G. (2020). Applications of Wireless Sensor Networks: An Up-To-Date Survey. *Applied System Innovation*, 3(1): 14.

Kane, M.B., Peckens, C., and Lynch, J.P. (2022). Introduction to Wireless Sensor Networks for Monitoring Applications. *Sensor Technologies for Civil Infrastructures (Second Edition)*. 335-368.

Lee, H.C., and Ke, K.H. (2018). Monitoring of Large-Area Iot Sensors Using A Lora Wireless Mesh Network System: Design and Evaluation. *IEEE Transactions On Instrumentation and Measurement*, 67(9): 2177-2187.

Li, C., Zhao, G., Meng, J., Zheng, Z., and Yu, S. (2021). Multi-Objective Optimization Strategy Based On Entropy Weight, Grey Correlation Theory, and Response Surface Method in Turning. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 28(5): 490-507.

Lin, J., Wu, H-C., and Chan, S-C. (2021). A New Regularized Recursive Dynamic Factor Analysis with Variable Forgetting Factor and Subspace Dimension for Wireless Sensor Networks with Missing Data. *IEEE Transactions On Instrumentation and Measurement*, 70: 1-13.

Liu, J., Wang, Y., Sun, G., and Pang, T. (2022). Multisurrogate-Assisted Ant Colony Optimization for Expensive Optimization Problems with Continuous and Categorical Variables. *IEEE Transactions on Cybernetics*, 52(11): 11348-11361.

Lo, C., Lynch, J.P., and Liu, M. (2013). Distributed Reference-Free Fault Detection Method for Autonomous Wireless Sensor Networks. *IEEE Sensors Journal*, 13(5): 2009-2019.

Michaelides, C., and Pavlidou, F.N. (2020). Mutual Aid Among Sensors: An Emergency Function for Sensor Networks. *IEEE Sensors Letters*, 4(9): 1-4.

Osman, M.F.S. (2022). Centralized Vs. Decentralized Planning of Dynamic Flows on Capacitated Transport Networks: A Comparative Computational Study. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 29(4): 453-463.

Peng, Y., Qiao, W., and Qu, L. (2022). Compressive Sensing-Based Missing-Data-Tolerant Fault Detection for Remote Condition Monitoring of Wind Turbines. *IEEE Transactions on Industrial Electronics*, 69(2): 1937-1947.

Priyadarshi, R., Gupta, B., and Anurag, A. (2020). Deployment Techniques in Wireless Sensor Networks: A Survey, Classification, Challenges, and Future Research Issues. *The Journal of Supercomputing*, 76(9): 7333-7373.

Suthaharan, S., Alzahrani, M., Rajasegarar, S., Leckie, C., and Palaniswami, M. (2010). Labelled Data Collection for Anomaly Detection in Wireless Sensor Networks. *2010 6th International Conference on Intelligent Sensors*, Brisbane, 269.

Wei, Y., Tong, D., Chen, Q., Sun, Y.Q., and Zhou, W.N. (2022). Simultaneous Actuator and Sensor Fault Estimation for Neutral-Type Systems Via Intermediate Observer. *Transactions of The Institute of Measurement and Control*, 44(7): 1505-1517.

Xiao, R., Liu, S., Li, Y., Ni, Y.C., and Du, X. (2020). Toward An Effective Locality-Sensitive Hashing Search for Wmsns Based on The Neighborhood Rough Set Approach. *IEEE Internet of Things Journal*, 7(11): 10985-10995.

Zhang, B., Niu, Z., and Feng, L. (2020). Evaluation System for Lean Knowledge Management Ability Based on Improved Gray Correlation Analysis. *International Journal of Industrial Engineering: Theory, Applications and Practice*, 27(5): 712-730.

Zhang, H., Liu, J., and Kato, N. (2018). Threshold Tuning-Based Wearable Sensor Fault Detection for Reliable Medical Monitoring Using Bayesian Network Model. *IEEE Systems Journal*, 12(2): 1886-1896.

Zidi, S., Moulahi, T., and Alaya, B. (2018). Fault Detection in Wireless Sensor Networks Through SVM Classifier. *IEEE Sensors Journal*, 18(1): 340-347.