Journal of Computer Science & Technology

**Indexed in:** SCIE, EI, INSPEC, CBST, DBLP, etc.

**Sponsored by:** ICT, CAS & CCF

# JCST Papers


**Only for academic and non-commercial use**

Thanks for reading!



JCST WeChat Subscription Account

JCST URL: https://jcst.ict.ac.cn

SPRINGER URL: https://www.springer.com/journal/11390

E-mail: jcst@ict.ac.cn

Online Submission: https://mc03.manuscriptcentral.com/jcst

Twitter: JCST_Journal

LinkedIn: Journal of Computer Science and Technology

Yang J, Sheng YQ, Wang JL *et al.* CAGCN: Centrality-aware graph convolution network for anomaly detection in industrial control systems. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 39(4): 967-983 July 2024. DOI: 10.1007/s11390-022-2149-y

# CAGCN: Centrality-Aware Graph Convolution Network for Anomaly Detection in Industrial Control Systems

Jun Yang (杨　骏), Yi-Qiang Sheng* (盛益强), Jin-Lin Wang (王劲林), and Hong Ni (倪　宏)

*National Network New Media Engineering Research Center, Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China*

*School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China*

E-mail: yangjun@dsp.ac.cn; shengyq@dsp.ac.cn; wangjl@dsp.ac.cn; nihong@mail.ioa.ac.cn

Received January 7, 2022; accepted September 8, 2022.

**Abstract**　In industrial control systems, the utilization of deep learning based methods achieves improvements for anomaly detection. However, most current methods ignore the association of inner components in industrial control systems. In industrial control systems, an anomaly component may affect the neighboring components; therefore, the connective relationship can help us to detect anomalies effectively. In this paper, we propose a centrality-aware graph convolution network (CAGCN) for anomaly detection in industrial control systems. Unlike the traditional graph convolution network (GCN) model, we utilize the concept of centrality to enhance the ability of graph convolution networks to deal with the inner relationship in industrial control systems. Our experiments show that compared with GCN, our CAGCN has a better ability to utilize this relationship between components in industrial control systems. The performances of the model are evaluated on the Secure Water Treatment (SWaT) dataset and the Water Distribution (WADI) dataset, the two most common industrial control systems datasets in the field of industrial anomaly detection. The experimental results show that our CAGCN achieves better results on precision, recall, and $F1$ score than the state-of-the-art methods.

**Keywords**　graph convolution network (GCN), data mining, network centrality, anomaly detection, industrial control system

## 1　Introduction

Industrial control systems (ICS) are the general designation of several different control systems and associated instrumentations to automate industrial processes. A typical industrial control system architecture (as shown in Fig.1) normally includes supervisory control and data acquisition (SCADA) systems, distributed control systems (DCSs), and other control system configurations such as programmable logic controllers (PLCs)[1]. SCADA is a control system that processes dispersed plants by the communications network and high-level process supervisory management. A DCS is a computerized control system to process a plant with many distributed control systems. A PLC is an industrial digital computer to control dispersed assets by ruggedized and adapted manufacturing processes, such as robotic devices and assembly lines.

To monitor and control the machining process remotely, engineering stations are granted permission to manage servers in ICS. This leads to potential security risks that the whole system are in danger if these engineering stations are hacked[2]. However, current ICS communication protocol focuses on real-time and stable control but ignores the potential security risks[3], which leads to the ICS being hacked if the hacker finds a way to invade the corporate network.

Regular Paper

This work was supported by the Chinese Academy of Sciences through the Strategic Priority Research Program under Grant No. XDC02020400.

*Corresponding Author

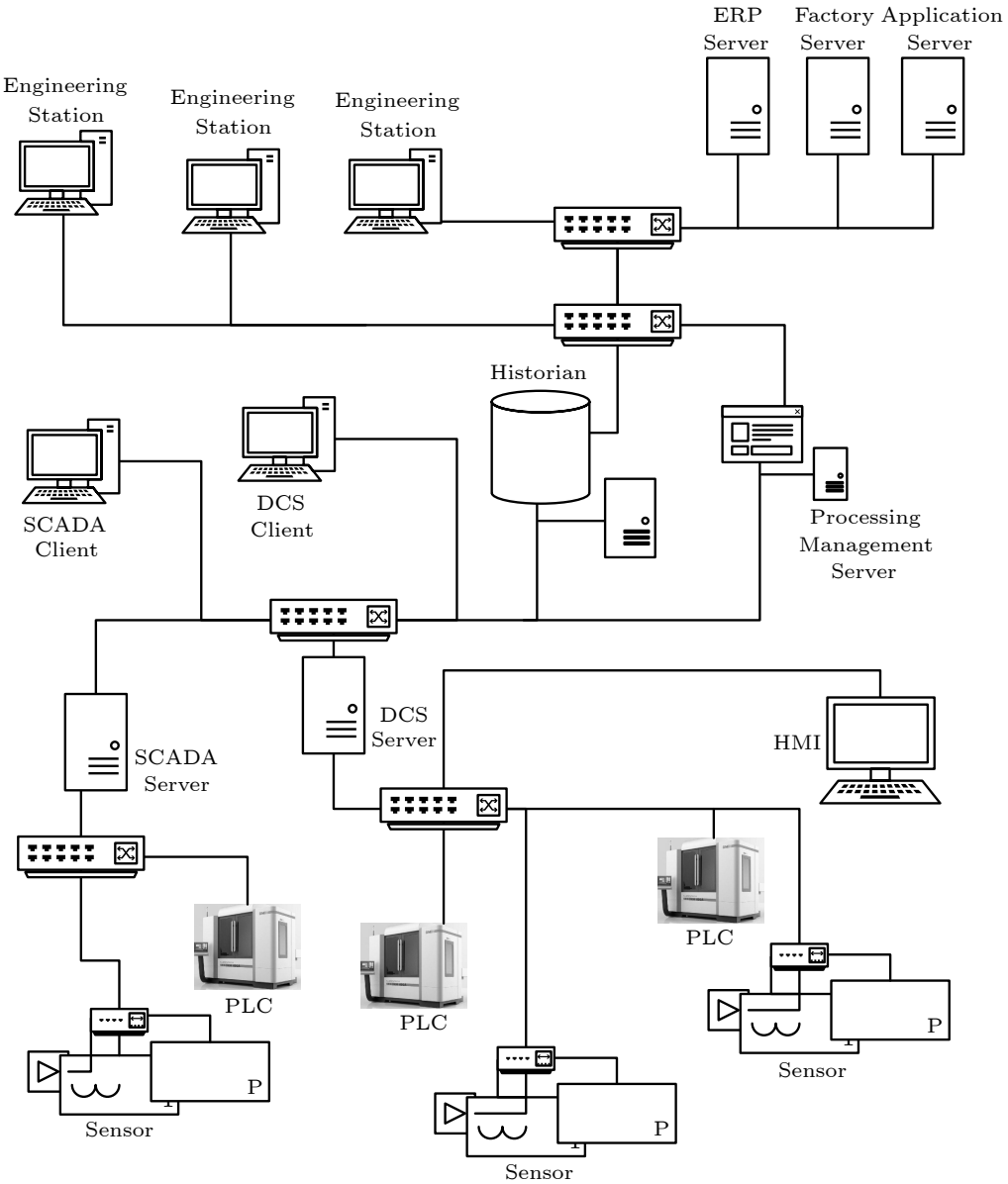



Fig.1. Typical ICS architecture.

Nowadays, ICS are widely used in automated production, energy, transportation, and water conservancy project and wastewater treatment. More and more frequently cyber security incidents in ICS lead to increasing attention to network security[4]. In 2007, a SCADA system in Canada was intruded by hackers, and the water control system was paralyzed[5]. In 2008, the train system in Poland was hacked, which led to four vehicles being derailed[6]. In 2010, Iran's nuclear plant was invaded by the Stuxnet virus, and the centrifuge was broken due to the abnormal action in the uranium enrichment process[7]. In 2011, the control system of China's oil refinery plant was infected by the Conficker virus, causing the communication between the control system server and the controller to be interrupted[8]. In 2012, the Flame virus attacked energy ICS in multiple Middle Eastern countries, collecting sensitive data on the energy industry[9]. In 2015, a power failure happened in 700 000 Ukrainian families, because the electrical power industrial system was hacked[10]. The WannaCry virus that broke out in 2017 threatened a large number of corporate office networks and industrial facilities[11]. These cases prove that the potential cybersecurity hazard in ICS has become a huge threat to social stability and national security.

Anomaly detection is a way to identify rare suspicious events or observations that deviate from nor-

mal behaviors. It is widely used in many fields and has recently been used in the field of ICS. The application of anomaly detection technology can find suspicious events, help managers or engineers find potential threats, and ensure safe and steady operation in ICS. There are a lot of studies about anomaly detection in ICS. These years, the development of machine learning (ML) leads to a new boost in research. Das *et al.*[12] used a Boolean function based supervised classification method to detect anomalies in ICS. Liu *et al.*[13] used the method of growing random trees for unsupervised anomaly detection in ICS. Hao *et al.*[14] proposed a recurrent neural network based anomaly detection method. Perales *et al.*[15] used feature engineering and LSTM (long short-term memory) in ICS anomaly detection. Mantere *et al.*[16] studied network traffic features for anomaly detection in ICS. Feng *et al.*[17] proposed a multi-level anomaly detection method in ICS by using an LSTM network. Kiass *et al.*[18] proposed a data clustering based anomaly detection method for ICS. Inoue *et al.*[19] used SVM (support vector machine) and DNN (deep neural network) to research unsupervised anomaly detection methods for ICS. Kim *et al.*[20] used sequence-to-sequence neural networks for anomaly detection in ICS.

However, most of the researches failed to consider the inner link between different components. The previous researches based on ML only used the input data collected by sensors as features and ignored the importance of relationships between different devices or nodes. In ICS, the device nodes are connected by the communication links, and the managers use the terminals to remotely control the devices or collect information. These individual devices (including sensors, controllers, servers, and terminals) connected by the network constitute the industrial control system. According to the industrial control production, processing, manufacturing process, and connection between devices, there is a topology relationship between different devices. Nowadays, the majority of researches about industrial control abnormal detection focus on the abnormal action in a single node and do not use the topology relationship in ICS. In order to measure the security of ICS more comprehensively and detect the abnormal more accurately, the topological information in ICS cannot be ignored. Lots of researches in the field of industrial control showed that the key node in an industrial control network can play an important role in the field of ICS security. Wang *et al.*[21] showed that the critical nodes are easier to be influenced under attack. Ur-Rehman *et al.*[22] pointed out that the critical nodes are more vulnerable than the normal nodes. Based on this research, we believe that the different importance of nodes may help us detect anomalies. In this paper, we use a GCN (graph convolution network) to extract the information of connection between different devices, and use the centrality to measure the importance of nodes to improve the ability of anomaly detection.

The main contributions of this paper are as follows. To utilize the topological connection relationship of nodes in ICS, this paper uses a GCN to extract the topological information for detecting anomalies in a more holistic perspective. The centrality is used to grant the weight of nodes to distinguish the importance and influence of different nodes in ICS. We evaluate our method with different centralities and compare it with state-of-the-art methods. Experimental results on two datasets show that the performance of the proposed method is much better than that of the state-of-the-art methods.

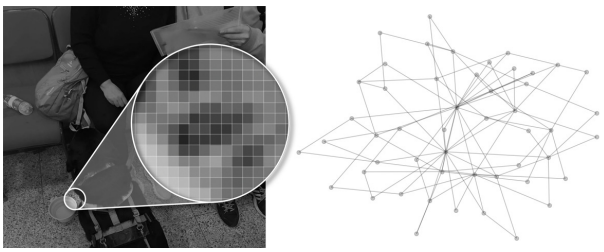
The remainder of this paper is organized as follows. In Section 2, we review the related work about anomaly detection and the application of deep learning in ICS. Section 3 presents the method we use in this paper. In Section 4, we introduce the comparative experiments and discuss the results of experiments on the proposed method and other state-of-the-art methods. Section 5 concludes this paper.

## 2 Related Work

Recently, researchers use many ML-based methods in anomaly detection in ICS. In addition to the methods mentioned above, there are many new methods based on ML that have achieved great results. Lin *et al.*[23] proposed a method named TABOR, which is a graphical model based method to detect anomalies in ICS. Li *et al.*[24] proposed a generative adversarial network to do anomaly detection in ICS. Zhang *et al.*[25] proposed a fuzzy probability Bayesian network approach for ICS security. Yoon and Ciocarlie[26] proposed a method to analyze the content of ICS traffic. Kravchik and Shabtai[27] used autoencoder to research anomaly detection in ICS. Elnour *et al.*[28] proposed a dual-isolation-forests-based model to detect attacks in ICS. In previous studies, researchers use the convolution neural network (CNN) to utilize the information of adjacent nodes and obtain some progress. CNN is one of the most successful models

that use the discrete convolution to aggregate the neighbor information of the pixels and achieve good performance in image processing[29–31] and audio analysis[32–33]. There are tons of relative studies about the application of CNN in the field of anomaly detection in ICS. Kravchik and Shabtai[34] used a one-dimensional CNN to detect cyber-attacks. Liu *et al.*[35] proposed a novel algorithm using CNN and process state transition to detect intrusion in ICS. Hu *et al.*[36] also used CNN in network data analysis and anomaly detection. Abdelaty *et al.*[37] proposed a method named DAICS, which is a deep learning solution for anomaly detection in ICS, using the deep convolution branch. Kusakina *et al.*[38] used VGG16 (one famous CNN model) to detect anomalies. However, they only use the CNN as a method to extract features and ignore the inner non-Euclidean relationship of adjacent nodes in ICS. According to the research on vulnerability mentioned earlier, when a key node is under attack, it may lead greater threats to the network[22]. However, current methods only consider the input data and ignore the influence of network structure. In this case, the key node is hard to be detected. Therefore, a new method which considers the importance of key nodes is very necessary.

Some studies have proved that the usage of centrality has huge potential in the study of ICS security. Salama *et al.* used centrality to calculate the importance of nodes and evaluated the robustness of power grid resilience enhancement under cyberattack[39]. Milanović and Zhu used a complex network theory modeling interconnected critical infrastructure systems and used the centrality to analysis the vulnerability of network[40]. These researches show that the information of adjacent nodes in ICS can help us detect the potential attacks. Although CNN shows a strong ability to deal with Euclidean structure data, it still has a theoretical drawback on non-Euclidean structure data, because non-Euclidean structure data does not meet the translational equivariance[41]. As shown in Fig.2(a), in the Euclidean graph, each pixel has eight neighboring pixels, therefore a CNN can use a square convolution kernel to extract spatial information. However, as shown in Fig.2(b), the topographic graph belongs to the non-Euclidean structure and the number of neighboring nodes is unfixed, therefore the traditional operation of convolution and pooling in CNN cannot be used to process directly[42]. The inner connection in ICS is a typical non-Euclidean topographical graph. In real



(a) (b)

Fig.2. Example of (a) Euclidean structure data and (b) non-Euclidean structure data.

world, the topographic graph also widely exists in computer networks, biology, and chemistry.

Due to the increasing demand for non-Euclidean structure data processing, in 2005, Gori *et al.*[43] put forward the conception of graph neural network (GNN) and Scarelli *et al.*[44] offered a detailed definition of GNN. However, GNN focuses on the recurrent graph neural network which has high computational complexity. In 2013, based on the theory of the spectral domain, Bruna *et al.*[41] defined graph convolution based on the Fourier transform. In 2017, Kipf and Welling[45] officially put forward GCN and used GCN to do a semi-supervised classification. The core purpose of GCN is to extract spatial information in non-Euclidean structure data. In order to achieve this purpose, there are two main methods: vertex domain and spectral domain. GCN uses the theory of the spectral domain. By defining a graph Fourier transform as well as graph convolution, Bruna *et al.*[41] proved the theory of GCN. In 2017, Kipf and Welling[45] formally put forward and created GCN structure. Compared with the traditional CNN, GCN has the ability to extract the spatial feature of the topology relationship. After the theory of GCN came out, a series of studies (such as localized spectral filtering[46], graph long short-term memory[47], graph attention network[48], temporal graph convolutional network[49], and spatial-temporal graph convolutional networks[50]) have been carried out and facilitate a wide range of GCN applications. GCN can use the topological relationship in ICS to aggregate the information between neighboring nodes and achieve better performance in ICS' abnormal detection. The invention of GCN enhances the development of anomaly detection in ICS greatly. However, the traditional GCN does not consider the importance of different nodes. In ICS, several key nodes have a greater influence on connectivity or stability[51]. To distinguish the importance of different nodes, centrality is a very

popular measure. We believe the application of centrality can improve the performance of GCN in ICS.

## 3 Methodology

Based on the research of GCN and centrality, we propose a centrality-aware graph convolution network (CAGCN), which is aware of centrality of nodes by using centrality to weight the key nodes, and use convolution to aggregate information. In this way, our method can utilize both topological connection information and node centrality information in ICS.

### 3.1 Graph Convolution Network

Compared with the traditional CNN, GCN can better explain the potential influence of the network structure. In the theory of GCN, the nodes in GCN will be influenced by the contiguous nodes. The closer a node is to other nodes, the greater its influence. If a node is under attack, it will influence the contiguous nodes until the whole system reaches a balance. In other words, the essence of GCN is the message passing of features in the graph network, and the effect of Laplace transformation is to aggregate the influence of features in the graph space.

#### *3.1.1 Laplacian Operator*

Laplacian is a matrix representation of a graph. Actually, the Laplacian operator can be used to evaluate the total gain under minor disturbance. The Laplacian operator is usually denoted by the symbol $\nabla$. We assume we have a graph $G = (V,\ E)$, $V$ is the node set of graph $G$ and $E$ is the edge set of $G$. In the discrete space, we have

$$\nabla f = \sum_{j \in N_i} (f_i - f_j),$$

where $f_i$ is the value of node $i$ and $N_i$ is the set of adjacent nodes of node $i$. Assuming the value of $E_{ij}$ is $w_{ij}$, we have

$$\nabla f = \sum_{j \in N_i} w_{ij} (f_i - f_j). \tag{1}$$

Because node $i$ and node $j$ are not connected, i.e., $w_{ij} = 0$, we can rewrite (1) as

$$\nabla f = \sum_{j \in N} w_{ij} (f_i - f_j).$$

Let $d_i$ be the degree of node $i$, $d_i = \sum_{j \in N} w_{ij}$, then we have

$$\begin{aligned} \nabla f &= \sum_{j \in N} w_{ij} f_i - \sum_{j \in N} w_{ij} f_j \\ &= d_i f_i - \boldsymbol{W}_i \boldsymbol{f} \\ &= \mathrm{diag}(d_i) \boldsymbol{f} - \boldsymbol{A} \boldsymbol{f} \\ &= (\boldsymbol{D} - \boldsymbol{A}) \boldsymbol{f} \\ &= \boldsymbol{L} \boldsymbol{f}. \end{aligned}$$

Then we get a discrete Laplace matrix $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. Here $\boldsymbol{D}$ is the degree matrix and $\boldsymbol{A}$ is the adjacency matrix. We can call it combinatorial Laplacian.

Because $\boldsymbol{L}$ is a positive semi-definite matrix, we can get a normalized graph Laplacian:

$$\boldsymbol{L} = \boldsymbol{I_N} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}} = \boldsymbol{U} \boldsymbol{\Lambda} \boldsymbol{U}^{\mathrm{T}},$$

where $\boldsymbol{U}$ is the eigenvalue of $\boldsymbol{L}$. we can call it symmetric normalized Laplacian. Because the Laplace matrix is symmetric, it can achieve spectral decomposition which is the core of the spectral domain of GCN.

#### *3.1.2 Graph Fourier Transform*

Mathematically, the graph Fourier transform is a mathematical transform which eigendecomposes the Laplacian matrix of a graph into eigenvalues and eigenvectors[52]. Analogously to classical Fourier transform, the eigenvalue of Laplacian matrix after graph Fourier transform represents frequencies and an eigenvectors form what is known as a graph Fourier basis.

The definition of traditional Fourier transform is:

$$F(\omega) = \mathcal{F}[f(t)] = \int f(t)\, \mathrm{e}^{-\mathrm{i}\omega t} \mathrm{d}t.$$

Here $f$ is the signal function. Note that the Laplacian operator is $\nabla$, we have

$$\nabla \mathrm{e}^{-\mathrm{i}\omega t} = \frac{\partial^2}{\partial t^2} \mathrm{e}^{-\mathrm{i}\omega t} = -\omega^2 \mathrm{e}^{-\mathrm{i}\omega t}.$$

Here $\mathrm{e}^{-\mathrm{i}\omega t}$ is the eigenfunction of $\nabla$.

We assume $\boldsymbol{L}$ is a Laplace matrix, $V$ is the set of nodes ($N$ being the number of the nodes), and $E$ is the set of edges. A graph signal $f : V \to R$ is a function defined on the vertices of the graph $G$. The signal $f$ maps every vertex $\{v_i\}_{i=1,\ \ldots,\ N}$ to a real number $f(i)$. Any graph signal can be projected on the eigenvectors of the Laplacian matrix, respectively. Let $\lambda_l$ and $u_l$ be the $l$-th eigenvalue and eigenvector of the Laplacian matrix, respectively, the graph Fourier transform of $\hat{f}$ is the graph signal $f$ on the vertices of $G$, and $G$ is the expansion of the term of the eigen-

function of $\boldsymbol{L}$. It is defined as:

$$F(\lambda_l) = \hat{f}(\lambda_l) = \sum_{i=0}^{N} f(i)\, \boldsymbol{u}_l^*(i),$$

where $\boldsymbol{u}_l^* = \boldsymbol{u}_l^{\mathrm{T}}$.

Since $\boldsymbol{L}$ is a real symmetric matrix, its eigenvectors form is an orthogonal basis. Hence an inverse graph Fourier transform (IGFT) exists, and it is written as:

$$IF\left[\hat{f}\right](i) = f(i) = \sum_{i=0}^{N} \hat{f}(\lambda_l)\, u_l(i).$$

Analogously to the classical Fourier transform, the graph Fourier transform provides a way to represent a signal in two different domains: the vertex domain and the graph spectral domain. The eigenvectors of the normalized Laplacian matrix are also a possible base to define the forward and inverse graph Fourier transform.

Just as we mention in the traditional Fourier transform, $\mathrm{e}^{-\mathrm{i}\omega t}$ is the eigenfunction of Laplace operator, $\boldsymbol{U}$ is the eigenvalue matrix of $\boldsymbol{L}$, we can rewrite the GFT and IGFT as:

$$GFT: \hat{\boldsymbol{f}} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{f},$$

$$IGFT: \boldsymbol{f} = \boldsymbol{U}^{\mathrm{T}}\hat{\boldsymbol{f}}.$$

Now we can define the graph convolution as:

$$\boldsymbol{f} * \boldsymbol{h} = \boldsymbol{U}\left(\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{h}\right) \odot \left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{f}\right)\right).$$

We can rewrite $\boldsymbol{h}$ as:

$$\begin{pmatrix} \hat{h}(\lambda_1) & & \\ & \ddots & \\ & & \hat{h}(\lambda_n) \end{pmatrix},$$

$$\hat{h}(\lambda_l) = \sum_{i=1}^{N} h(i)\, \boldsymbol{u}_l^*(i).$$

Using $\mathrm{diag}(\theta)$ replace $\mathrm{diag}(\hat{h}(\lambda_l))$, we can get

$$y = \sigma\left(\boldsymbol{U} g_\theta \boldsymbol{\Lambda} \boldsymbol{U}^{\mathrm{T}} x\right).$$

Here, $g_\theta(\boldsymbol{\Lambda})$ is the convolution kernel,

$$g_\theta(\boldsymbol{\Lambda}) = \begin{pmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_n \end{pmatrix}.$$

$\sigma$ is the active function and $\Theta = (\theta_1, \theta_2, ..., \theta_n)$ is the learnable variable.

However, we need compute the product of $\boldsymbol{U}$, $\mathrm{diag}(\theta)$, and $\boldsymbol{U}^{\mathrm{T}}$ with $n$ factors; therefore it leads to the computational complexity of $O(n^2)$.

In order to deal with this problem, Hammond *et al.*[53] proved that $g_\theta(\boldsymbol{\Lambda})$ can be approached by $K$-order Chebyshev polynomials $T_K(x)$, in other words,

$$g_\theta(\boldsymbol{\Lambda}) = \sum_{k=0}^{K} \left(\theta_k T_k\left(\hat{\boldsymbol{\Lambda}}\right)\right),$$

$$\hat{\boldsymbol{\Lambda}} = \frac{2}{\lambda_{\max}} - \boldsymbol{I},$$

where $\lambda_{\max}$ is the max eigenvalue of the Laplace matrix $\boldsymbol{L}$.

Using Chebyshev polynomials:

$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x),$$

$$T_0(x) = 1,$$

$$T_1(x) = x.$$

We have

$$y \approx \sum_{k=0}^{K} \left(\theta_k T_k\left(\hat{\boldsymbol{L}}\right)\right),$$

$$\hat{\boldsymbol{L}} = \frac{2}{\lambda_{\max}} - \boldsymbol{I}.$$

If we let $\lambda_{\max} = 2$, then

$$y = \theta\left(\boldsymbol{I} + \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{1/2}\right)x.$$

Because the range of $\boldsymbol{I} + \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{1/2}$ is between the 0 and 2, we can renormalization it as $\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{1/2}$, where

$$\tilde{\boldsymbol{D}} = \mathrm{diag}\left(\sum_J \tilde{A}_{ij}\right),$$

$$\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}.$$

Assuming $\boldsymbol{G}^i$ is the $i$-th layer of a GCN network structure, and $\boldsymbol{W}^i$ is the weight matrix of the $i$-th layer network, then we can get the final GCN form:

$$\boldsymbol{G}^{i+1} = \sigma\left(\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{1/2}\boldsymbol{G}^i\boldsymbol{W}^i\right).$$

### 3.2 Interpretability of GCN

Based on the importance of nodes, the process of GCN aggregating information can be explained by the message passing theory.

In an industrial control network, a more important node, compared with a less important node, contains more information. In the theory of message passing, the adjacency matrix $\boldsymbol{A}$ is used to describe the connection relationship between different nodes, and the essence of the adjacency matrix is to pass messages between adjacent nodes. We assume the influence of a node to itself as 1, and we can get $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$.

For the graph in Fig.3, the adjacency matrix $\boldsymbol{A}$ is

$$\boldsymbol{A} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2}$$

In (2), $A_{ij} = 1$ means that node $i$ is connected to node $j$. Therefore, it can be used to explain the connection relationship of nodes in the graph.

After adding self-connection, $\tilde{\boldsymbol{A}}$ of the graph in Fig.3 is

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}. \tag{3}$$

In (3), the connection of a node to itself is also taken into account.

On the other hand, degree matrix $\tilde{\boldsymbol{D}}$ is used to describe the influence of a node to its adjacent nodes:

$$\tilde{\boldsymbol{D}} = \mathrm{diag}(\sum_{J} \tilde{A}_{ij}).$$

For the graph in Fig.3, $\tilde{\boldsymbol{D}}$ is

$$\tilde{\boldsymbol{D}} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}. \tag{4}$$

In (4), the larger $\tilde{D}_{ij}$ is, the greater the influence of the messages it passes to other nodes is.

Considering the final GCN form ($\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{1/2}$) we mentioned before, if we use the message passing theory to explain the adjacency matrix and the degree matrix, the operation of spatial domain in GCN is the process of message passing.

### 3.3 Network Centrality

Centrality is an essential index in network analysis, which is used to estimate the importance of a node in a network. Centrality is widely used in different network analysis[54], including social network[55–58], Citation Network[59–62], biological network[63–65], transfer network[66–68], etc. As mentioned before, the centrality shows a huge potential in the study of ICS[39–40].

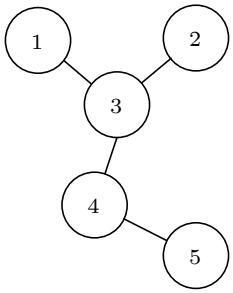



Fig.3. Simple graph for interpretability of GCN.

Different centralities can reflect the importance of nodes in the network from different angles, and the most common centralities are degree centrality[69], closeness centrality[70], and betweenness centrality[71].

In degree centrality[69], the node with more neighboring nodes has more importance and influence. For a node, its degree centrality is:

$$DC\,(i) = \frac{k_i}{n-1}.$$

Here $k_i = \sum_{i=1}^{n} a_{ij}$, $k_i$ is the amount of neighboring nodes of $v_i$ which can be called degree, $a_{ij}$ is the element in the $i$-th row and the $j$-th column of the adjacency matrix $\boldsymbol{A}$, and $n$ is the number of nodes.

The metric of degree centrality has a low computational complexity.

However, degree centrality only focuses on the local information but ignores the global information; therefore in many cases, it cannot measure the real importance of the node.

Closeness centrality[70] uses the average distance to measure the centrality of a node in the network. The node with the shortest average distance with other nodes has the largest closeness centrality. The definition of distance for node $v_i$ is:

$$d_i = \frac{1}{n-1}\sum_{i \neq j} d_{ij}.$$

Because the closeness centrality uses the average distance to justify the importance of nodes, we can define closeness centrality as the reciprocal of $d_i$:

$$CC\,(i) = \frac{1}{d_i} = \frac{n-1}{\sum\limits_{i \neq j} d_{ij}}.$$

Closeness centrality is widely used, but its time complexity is very high because it needs to traverse all the nodes to count the shortest distance of every node.

Betweenness centrality[71] counts the shortest path of all the nodes and quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. The definition of betweenness centrality is:

$$BC\,(i) = \frac{2}{(n-1)\,(n-2)} \sum \frac{g_{st}^{i}}{g_{st}},$$

where $g_{st}$ is the total number of all the shortest paths from node $v_s$ to node $v_t$, $g_{st}^{i}$ is the number of times of node $v_i$ as a bridge along the shortest path between $v_s$ and $v_t$, and $n$ is the total number of nodes.

Betweenness centrality is very important in the computer networking field because a computer will always select the shortest path in the Internet communication. Similar to closeness centrality, the time complexity of betweenness centrality is quite high; therefore there are many studies about the optimization of fast calculation of betweenness centrality[72].

Besides these common centralities, there are other centralities that attract the attention of researchers, such as eccentricity[73], semi-local centrality[74], eigenvector centrality[75], and information indices[76]. These centralities can measure the importance of nodes at different angles.

### 3.4 Proposed Centrality-Aware Graph Convolution Network

We propose a centrality-aware graph convolution network (CAGCN) based on the previous researches on GCN and centrality.

In the traditional GCN:

$$\boldsymbol{G}^{i+1} = \sigma\left(\tilde{\boldsymbol{D}}^{-1/2}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{D}}^{1/2}\boldsymbol{G}^{i}\boldsymbol{W}^{i}\right),$$

$$\tilde{\boldsymbol{D}} = \mathrm{diag}(\sum_{J}\tilde{\boldsymbol{A}}_{ij}),$$

$$\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I},$$

where $\boldsymbol{A}$ is the adjacency matrix and $\boldsymbol{D}$ is the degree matrix. Normally, $\boldsymbol{A}$ is used to describe the value of edge, and $\boldsymbol{D}$ is used to describe the value of mode.

However, just as we mentioned before, the importance of a different node in ICS is uncertain and a more important key node may have more influence than a less important node.

In order to solve this problem, we use a centrality-aware adjacency matrix $\boldsymbol{A}'$ to replace the traditional adjacency $\boldsymbol{A}$. Our adjacency matrix $\boldsymbol{A}'$ includes two pieces of different information: the adjacent node information from original adjacency matrix $\boldsymbol{A}$ and the node centrality information from centrality matrix $\boldsymbol{C}$. In GCN, an adjacency matrix $\boldsymbol{A}$ is a 0-1 matrix:

$$A_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ are connected,} \\ 0, & \text{otherwise.} \end{cases}$$

We define $\boldsymbol{C}$ as the centrality matrix, then we can get a new centrality-aware adjacency matrix $\boldsymbol{A}'$:

$$A'_{ij} = C_{ii}A_{ij} = \begin{cases} C_{ii}, & \text{if } A_{ij} = 1, \\ 0, & \text{if } A_{ij} = 0. \end{cases}$$

Just like the normal GCN, we get the new degree matrix $\boldsymbol{D}'$. Now we have the new GCN structure:

$$\boldsymbol{G}^{i+1} = \sigma\left(\tilde{\boldsymbol{D}'}^{-1/2}\tilde{\boldsymbol{A}}'\tilde{\boldsymbol{D}'}^{1/2}\boldsymbol{G}^{i}\boldsymbol{W}^{i}\right),$$

$$\tilde{\boldsymbol{D}}' = \mathrm{diag}(\sum_{J}\tilde{A}'_{ij}),$$

$$\tilde{\boldsymbol{A}}' = \boldsymbol{A}' + \boldsymbol{I},$$

where $\sigma$ is the active function, $\boldsymbol{W}^{i}$ is the weight matrix of the $i$-th layer of the network, and $\boldsymbol{G}^{i}$ is the $i$-th layer of CAGCN network structure.

Because the proposed CAGCN is also a variant of GCN, the interpretability mentioned earlier still applies.

Just as we mentioned before, the definition of network centrality is not unique. For exploring the suitable centrality in ICS, we choose several different centralities and create a mixed centrality. This mixed centralities can be defined as follows:

$$MC(i) = K_{\mathrm{DC}}DC(i) + K_{\mathrm{CC}}CC(i) + K_{\mathrm{BC}}BC(i),$$

where $K_{\mathrm{DC}}$, $K_{\mathrm{CC}}$, and $K_{\mathrm{BC}}$ are the coefficients used to adjust the ratio of degree centrality, closeness centrality, and betweenness centrality, respectively. This ratio can be set manually or learned by our CAGCN network during training.

After we choose the centrality to use, we can define a new centrality matrix $\boldsymbol{C}$. For $i = j$, $C_{ij}$ is the value of the chosen centrality. By introducing a mixed centrality, our CAGCN can have better versatility in different network environments.

## 4 Evaluation

In this section, we conduct experiments by answering the following research questions.

- $RQ1$. Compared with the state-of-the-art methods, whether the proposed method can have better precision when detecting anomalies in ICS?
- $RQ2$. Whether the proportion of underreporting of some attacks in the process of anomaly detection by the proposed model is acceptable?
- $RQ3$. How well does the proposed model perform comprehensively in terms of the trade-off between precision and sensitivity?
- $RQ4$. How to understand changes in features during centrality-based message passing and use this to understand the proposed model?

### 4.1 Datasets

To evaluate the performance of the proposed CAGCN, we test our model in the Secure Water Treatment (SWaT) dataset[77] and the Water Distribution (WADI) dataset[78]. The SWaT dataset and the WADI dataset are the most common ICS anomaly detection datasets. The abbreviations for sensors and actuators appearing in the SWaT dataset and the WADI dataset are summarized in Table 1. The details of the datasets are as follows.

**Table 1.** Abbreviations of Sensors and Actuators in the Datasets

| Abbreviation | Definition |
|---|---|
| AIT | Analyzer indicator transmitter |
| DPIT | Inferential pressure indicator transmitter |
| FIT | Flow indicator transmitter |
| FS | Flow switch |
| LIT | Level indicator transmitter |
| LT | Level transmitter |
| MCV | Motorized consumer valve |
| MV | Motorized valve |
| P | Pump |
| PIT | Pressure indicator transmitter |

#### 4.1.1 *The SWaT Dataset*

The SWaT dataset is an industrial control systems anomaly detection dataset provided by the Singapore University of Technology and Design[77]. SWaT is a water treatment site launched in 2015 for cybersecurity research. The data collection process was implemented on a six-stage SWaT testbed, as shown in Fig.4. Nowadays, the SWaT dataset is one of the most popular real-world public cybersecurity datasets. Lots of researchers use the SWaT dataset to test their models' performances in complex real-world environments[30].

Fig.4. Secure Water Treatment (SWaT) testbed.

As shown in Fig.5, SWaT has six main processes according to the physical and control components of the water treatment facility. The details of the sensors and actuators in every processing stage are summarized in Table 1A in the supplementary file①.

This system can produce five gallons of clean water per minute. It has a layered communication network, programmable logic controllers (PLCs), human machine interfaces (HMIs), a supervisory control and data acquisition (SCADA) workstation, and a historian server used to record data. Authors of the SWaT dataset collected network traffic and all the physical properties obtained from all the sensors and actuators in their experiment environment. The details of attacks to the system are shown in Table 2A in the supplementary file①.

The SWaT project team has built a scaled down version of a real-world industrial water treatment system and collected data for 11 days. The system operated 24 hours per day during the entire 11-day period. The system ran without any attacks during the first seven days, then the system was under attack in the remaining four days.

In the SWaT dataset, 496 800 samples were collected in the first seven days and 449 919 samples were collected when attacks were inserted to the system in the last four days. Among the samples collect-

①https://github.com/yangjun1994/CAGCN/blob/main/Supplementary%20File.pdf, Jul. 2024.

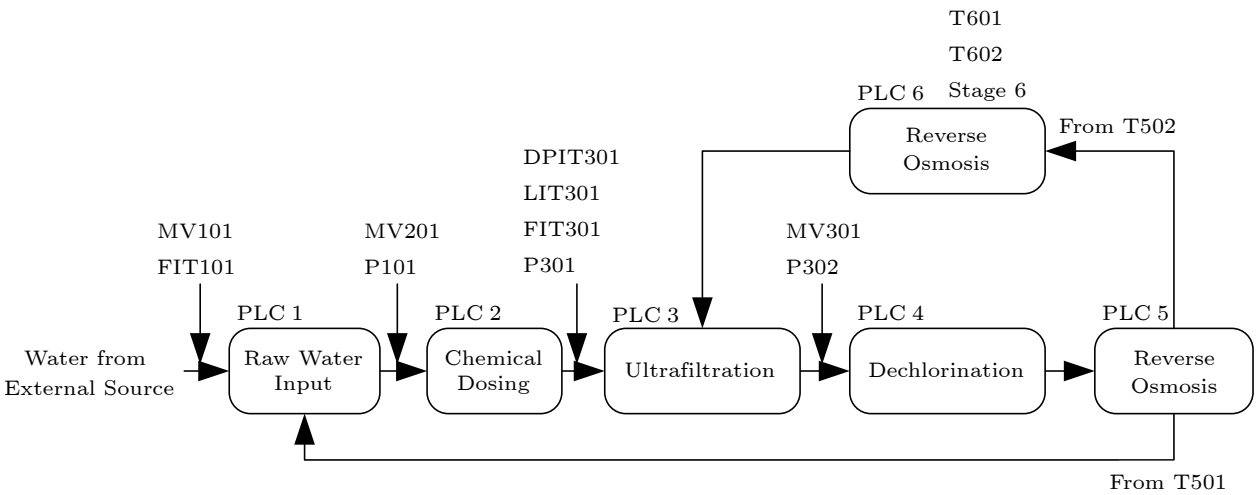



Fig.5. Processes of the SWaT testbed.

ed in the last four days, 396 019 of them are normal and the other 53 900 samples are attack samples. All features in the network traffic were obtained from 51 sensors and actuators. In this dataset, data was labeled as normal or anomaly by the authors of the dataset.

The SWaT dataset includes the 51 features which can be used to build the feature matrix. The SWaT dataset also provides the connection relationship between different devices, which can be used to build the adjacency matrix in GCN. The feature matrix and adjacency matrix are used as the input in our model.

### 4.1.2 *The WADI Dataset*

The WADI dataset[78] is another high-quality real-world ICS anomaly detection dataset provided by the authors of the SWaT dataset. WADI is a natural extension of SWaT. The authors of the WADI dataset also collected normal and attack data on their testbed, as shown in Fig.6.

Similar to SWaT, WADI also has a layered communication network, PLCs, HMIs, an SCADA workstation, and a historian server used to record data. The data was collected from all the sensors and actuators in their experimental environment. As shown in Fig.7, WADI has four main processes, and the details of the sensors and actuators in every processing stage are summarized in Table 3A, and attack details of the WADI dataset are shown in Table 4A in the supplementary file[②].

Fig.6. The water distribution (WADI) testbed.

Compared with the SWaT dataset, the WADI dataset includes more features and samples. In the WADI dataset, 789 371 samples with 123 features were collected in the first fourteen days without attacks. In the last two days, 172 801 samples with cyberattacks were collected. We can also build the feature matrix and adjacency matrix from the WADI dataset.

## 4.2 Metrics

We use precision, recall, and $F1$ score to evaluate the model performance[28].

To explain precision, recall, and $F1$ score, we use $TP$, $FP$, and $FN$ in the following ways:

- $TP$ (true positive): anomaly in actual is classified as anomaly in prediction.
- $FP$ (false positive): normal in actual is classified as an anomaly in prediction.

[②]https://github.com/yangjun1994/CAGCN/blob/main/Supplementary%20File.pdf, Jul. 2024.

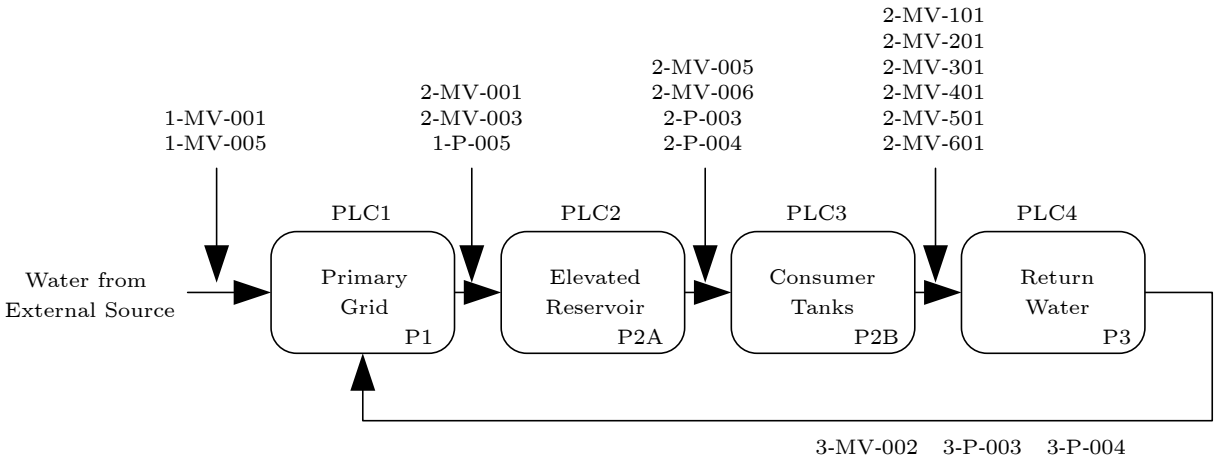



Fig.7. Processes of the WADI testbed.

• $FN$ (false negative): anomaly in actual is classified as normal in prediction.

Precision is the ratio of the correctly predicted positive anomaly data to the total predicted positive anomaly data.

$$precision = \frac{TP}{TP + FP}.$$

Recall is the ratio of the correctly predicted positive anomaly data to all the data in actual anomaly class.

$$recall = \frac{TP}{TP + FN}.$$

$F1$ score can be thought as a weighted average of the model precision and recall, with a maximum of 1 and a minimum of 0.

$$F1\ score = \frac{2 \times recall \times precision}{recall + precision}.$$

### 4.3 Experimental Setup

Our experiment was conducted on a workstation with Intel Core i9-7900X CPU @3.30 GHz, 128 G RAM, 1 TB SSD, 4 TB HDD, NVIDIA Titan XP GPU. The details are shown in Table 2.

In order to analyze the value of the centralities in different network structures, we used five kinds of centralities in our experiment: degree centrality, closeness centrality, betweenness centrality, mixed centrality (the ratio of degree centrality, closeness centrality, and betweenness centrality is 1:1:1), and the self-learning centrality (the ratio of degree centrality, closeness centrality, and betweenness centrality is the self-learning variable in our network which could be updated in the training process). We used a randomly 3:1:1 split of training set, validation set and test set on the both datasets. We also used cross validations to test the robustness of our result. This 3:1:1 partitaioning ratio of the both datasets is also used in the comparison methods below. We uploaded the code to a GitHub code repository[③].

**Table 2.** Experimental Environment

| Type | Value |
|---|---|
| CPU | Intel Core i9-7900X @3.30 GHz |
| Memory | 128 GB |
| Storage | 1 TB SSD and 4 TB HDD |
| GPU | NVIDIA Titan XP |
| GPU memory | 12 GB |
| Operating | System Ubuntu 18.04 LTS |
| Python version | Python 3.6 |
| CUDA version | CUDA 10 |
| cuDNN version | cuDNN 7.4 |
| TensorFlow version | TensorFlow 1.13.1 |

### 4.4 Comparison Methods

We compared our method with several state-of-the-art methods using the SWaT dataset and the WADI dataset.

In the selection of comparative methods, we covered as many types of models as possible: traditional neural network models, generative adversarial models, auto-encoders-based models, tree-based models, etc.

We compared our proposed CAGCN method with nine existing methods: DIF[28], MAD-GAN[24], AE[27], 1D-CNN[27], SVM[19], DNN[19], PCA-Reconstruction[27], Windowed-PCA[27], and DAICS[37].

DIF is a dual-isolation-forests-based attack detec-

[③]https://github.com/yangjun1994/CAGCN, Jul. 2024.

tion framework. MAD-GAN is a multivariate anomaly detection method based on a generative adversarial network. AE is a method based on auto-encoders. 1D-CNN is a method using one-dimensional CNN to detect anomalies. SVM is a method based on support vector machine. DNN is a method using a deep neural network. PCA-Reconstruction and Windowed-PCA are two methods based on principal component analysis (PCA). DAICS is a deep learning solution for anomaly detection in ICS using deep convolutional branch.

### 4.5 RQ1: Precision of Proposed Method

The results of the experiment based on the SWaT dataset and the WADI dataset are shown in Fig.8.

As for the precision results on the SWaT dataset, CAGCN with self-learning centrality (CAGAN-SLC) achieves the highest precision result (0.991). CAGCN with degree centrality (CAGCN-DC), CAGCN with closeness centrality (CAGCN-CC), CAGCN with betweenness centrality (CAGCN-BC), and CAGCN with 1:1:1 mixed centrality (CAGCN-MC) achieve at least 0.989 in precision. As for the compared methods, MAD-GAN achieves 0.990 and DNN gets 0.983 in our experiment. The other methods show a significantly low predictive ability. AE gets the lowest score with 0.726.

As for the precision results on the WADI dataset, CAGCN-SLC gets the highest result in precision (0.942). CAGCN-DC, CAGCN-CC, CAGCN-BC, and CAGCN-MC achieve at least 0.916 in precision. As for the compared methods, DAICS achieves 0.908, DNN gets 0.880, MAD-GAN gets 0.869, and the other compared methods get low results between 0.510 and 0.869.

Based on the results mentioned above, we believe that our model can have better precision when detecting anomalies in ICS.

### 4.6 RQ2: Sensitivity of Proposed Method

We used the evaluation metric recall, introduced earlier, to evaluate the sensitivity of the proposed method and the compared methods, because this evaluation metric directly calculates the proportion of true positive attacks in all attacks.

The recall results on the SWaT dataset and the WADI dataset are shown in Fig.9.

As for the recall results on the SWaT dataset, CAGCN-SLC also achieves the highest score (0.872). The following is CAGCN-DC (0.861). CAGCN-CC, CAGCN-BC, and CAGCN-MC get a similar level at 0.85. On the other hand, DAICS gets a good recall score at 0.861. 1D-CNN, Windowed-PCA, and DIF get the recall score of 0.854, 0.841, and 0.835, respec-

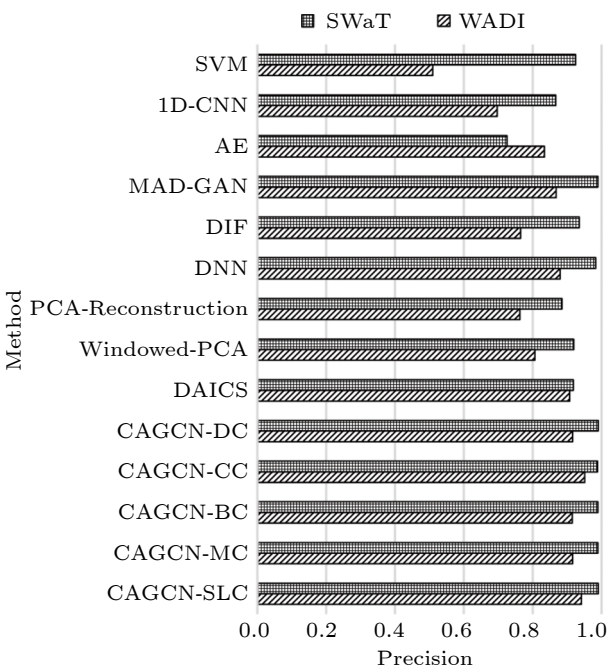



Fig.8. Precision of the proposed method and the compared methods on datasets SWaT and WADI.

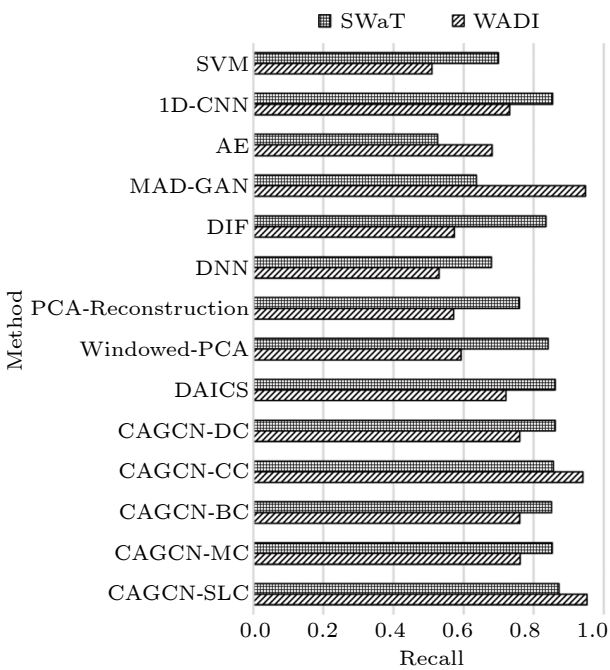



Fig.9. Recall of the proposed method and the compared methods on datasets SWaT and WADI.

tively. The recall scores of the other compared methods are below 0.8.

As for the recall results on the WADI dataset, CAGCN-SLC also achieves the highest score (0.952). The following is CAGCN-CC (0.940). CAGCN-DC, CAGCN-BC, and CAGCN-MC obtain similar results of 0.76. The recall score of MAD-GAN is 0.948. This score is very close to the best CAGCN method. The recall scores of 1D-CNN and DAICS are about 0.721 and 0.731, respectively. The other compared methods only get recall scores between 0.510 and 0.681.

Based on the results above, the proportion of underreporting attacks in the process of anomaly detection of the proposed model is acceptable and better than that of most of the compared methods.

### 4.7 RQ3: *F*1 Score of Proposed Method

*F*1 score can find the best trade off between precision and recall of the proposed method.

The *F*1 score results on the SWaT dataset and the WADI dataset are shown in Fig.10.

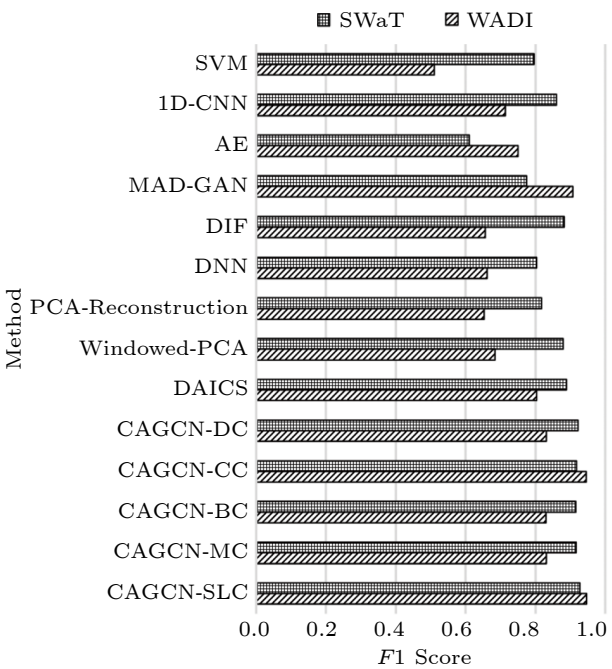



Fig.10. *F*1 score of the proposed method and the compared methods on datasets SWaT and WADI.

As for the *F*1 score results on the SWaT dataset, all the *F*1 scores of our CAGCN method are over 0.915 and the highest one is CAGCN-SLC (0.928). DIF achieves the best performance among all the compared methods (0.882). According to our results, the *F*1 scores of our CAGCN method are much better than those of the compared methods.

As for *F*1 score results on the WADI dataset, CAGCN-SLC also gets the best score (0.947). CAGCN-CC gets a similarly good result of 0.945. CAGCN-DC, CAGCN-BC, and CAGCN-MC gain similar results around 0.83. MAD-GAN has a *F*1 score of 0.907. The other compared methods get *F*1 scores between 0.510 (SVM) and 0.804 (DAICS).

Based on the results above, in terms of the trade-off between precision and sensitivity, the overall performance of the proposed model is better than the overall performance of the compared methods.

### 4.8 RQ4: Interpretability of Proposed Method

According to our result, LIT401 is a key node with high centrality. We compared the value of LIT401 and the adjacent node before and after attack 26. As shown in Table 3, when the system was under attack, the original feature of LIT401 decreases dramatically (1.720 to 0.700) whereas the original features of LIT301 and LIT501 do not change at all. As for the features after message passing, because the LIT401 passes the message to its adjacent nodes, the features of LIT301 and LIT501 both decrease (2.302 to 2.251, 1.80 to 1.785). These results prove that the change of key node could lead to a wide change in the whole network, therefore the model can identify this fluctuation easily. Then our model is able to detect the anomaly of LIT401.

Just as mentioned before, the application of GCN can be deemed as the process of message passing. GCN can catch the process of message passing in attacks and pay more attention to the key nodes in industrial control networks. On the other hand, the introduction of centrality also improves the ability to detect attacks in key nodes. Due to the application of GCN and centrality, the performance of our method is better than that of the compared methods.

**Table 3**. Changes to Values Before and After Message Passing in an Attack Case

| Value | LIT301 | LIT401 | LIT501 |
|---|---|---|---|
| Original values (before attack) | 2.216 | 1.720 | 1.733 |
| Original values (after attack) | 2.216 | 0.700 | 1.733 |
| Values after message passing (before attack) | 2.302 | 1.782 | 1.800 |
| Values after message passing (after attack) | 2.251 | 0.762 | 1.785 |

## 5 Conclusions

In this paper, we created a centrality-aware graph convolution network to detect anomalies in industrial control systems (ICS). Our method combines the merits of GCN and centrality by using the centrality of nodes to improve the detective ability of GCN. The topological information is extracted by a graph convolution operation and a node's importance is granted by its centrality. According to the results, it is clear that compared with state-of-the-art methods, our CAGCN method, especially CAGCN-SLC, achieves better performance on two ICS datasets. Our results imply that the topological relationship in ICS should not be ignored in anomaly detection.

We used three common centralities (degree centrality, closeness centrality, and betweenness centrality) and the combination of these centralities to enhance the performance of GCN. Our results proved that the application of centrality can improve the performance of anomaly detection in ICS. Our CAGCN method with self-learning combination centrality achieved the best performance among all the compared methods.

In future, we will try more different centralities to test the learning ability of our model as well as design a more effective centrality for mining the inner information in ICS. We also plan to test our method in more kinds of datasets to verify the robustness of our method. We have updated our code in GitHub and provided some cases about how to use our model to improve the performance of other GCN models.

**Conflict of Interest** The authors declare that they have no conflict of interest.

## References


[1] Stouffer K, Falco J, Scarfone K. Guide to industrial control systems (ICS) security. National Institute of Standards and Technology, 2011. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-82.pdf, July 2024.

[2] Drias Z, Serhrouchni A, Vogel O. Analysis of cyber security for industrial control systems. In *Proc. the 2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications*, Aug. 2015. DOI: 10.1109/SSIC.2015.7245330.

[3] Galloway B, Hancke G P. Introduction to industrial control networks. *IEEE Communications Surveys & Tutorials*, 2013, 15(2): 860–880. DOI: 10.1109/SURV.2012.071812.00124.

[4] Ogie R I. Cyber security incidents on critical infrastructure and industrial networks. In *Proc. the 9th International Conference on Computer and Automation Engineering*, Feb. 2017, pp.254–258. DOI: 10.1145/3057039.3057076.

[5] Zhou S X, Han J H, Li C, Wu D C. Research on trusted measurement of industrial control network with Markov reward model. *Telecommunications Science*, 2015, 31(2): 113–117, 139. DOI: 10.11959/j.issn.1000-0801.2015013.

[6] Wei Q Z. Industrial network control system security and management. *Measurement & Control Technology*, 2013, 32(2): 87–92. DOI: 10.19708/j.ckjs.2013.02.023.

[7] Kim S, Heo G, Zio E, Shin J, Song J G. Cyber attack taxonomy for digital environment in nuclear power plants. *Nuclear Engineering and Technology*, 2020, 52(5): 995–1001. DOI: 10.1016/j.net.2019.11.001.

[8] Lu G M. The analysis of present situation and future threats for the industrial control security in China. *Cyberspace Security*, 2018, 9(3): 1–7. DOI: 10.3969/j.issn.1674-9456.2018.03.001.

[9] Munro K. Deconstructing flame: The limitations of traditional defences. *Computer Fraud & Security*, 2012, 2012(10): 8–11. DOI: 10.1016/S1361-3723(12)70102-1.

[10] Zhang X M, Wang L H, He Y Y, He S P. Analysis of potential vulnerabilities and security testing in industrial control system. *Chinese Journal on Internet of Things*, 2017, 1(1): 34–39. DOI: 10.11959/j.issn.2096-3750.2017.00005.

[11] Kshetri N, Voas J. Hacking power grids: A current problem. *Computer*, 2017, 50(12): 91–95. DOI: 10.1109/MC.2017.4451203.

[12] Das T K, Adepu S, Zhou J Y. Anomaly detection in industrial control systems using logical analysis of data. *Computers & Security*, 2020, 96: 101935. DOI: 10.1016/j.cose.2020.101935.

[13] Liu L W, Hu M D, Kang C Q, Li X Y. Unsupervised anomaly detection for network data streams in industrial control systems. *Information*, 2020, 11(2): 105. DOI: 10.3390/info11020105.

[14] Hao Y R, Sheng Y Q, Wang J L, Li C P. Network security event prediction based on recurrent neural network. *Journal of Network New Media*, 2017, 6(5): 54–58. DOI: 10.3969/j.issn.2095-347X.2017.05.010. (in Chinese)

[15] Perales Gómez Á L, Fernández Maimó L, Celdrán A H, García Clemente F J. MADICS: A methodology for anomaly detection in industrial control systems. *Symmetry*, 2020, 12(10): 1583. DOI: 10.3390/sym12101583.

[16] Mantere M, Sailio M, Noponen S. Network traffic features for anomaly detection in specific industrial control system network. *Future Internet*, 2013, 5(4): 460–473. DOI: 10.3390/fi5040460.

[17] Feng C, Li T T, Chana D. Multi-level anomaly detection in industrial control systems via package signatures and LSTM networks. In *Proc. the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Jun. 2017, pp.261–272. DOI: 10.1109/DSN.2017.34.

[18] Kiss I, Genge B, Haller P, Sebestyén G. Data clustering-based anomaly detection in industrial control systems. In *Proc. the 10th IEEE International Conference on Intelligent Computer Communication and Processing*, Sept. 2014, pp.275–281. DOI: 10.1109/ICCP.2014.6937009.

[19] Inoue J, Yamagata Y, Chen Y Q, Poskitt C M, Sun J. Anomaly detection for a water treatment system using unsupervised machine learning. In *Proc. the IEEE International Conference on Data Mining Workshops*, Nov. 2017, pp.1058–1065. DOI: 10.1109/ICDMW.2017.149.

[20] Kim J, Yun J H, Kim H C. Anomaly detection for industrial control systems using sequence-to-sequence neural networks. In *Proc. the 2019 International Workshops*, Sept. 2019, pp.3–18. DOI: 10.1007/978-3-030-42048-2_1.

[21] Wang T Y, Zeng P, Zhao J M, Liu X D, Zhang B W. Identification of influential nodes in industrial networks based on structure analysis. *Symmetry*, 2022, 14(2): 211. DOI: 10.3390/sym14020211.

[22] Ur-Rehman A, Gondal I, Kamruzzaman J, Jolfaei A. Vulnerability modelling for hybrid industrial control system networks. *Journal of Grid Computing*, 2020, 18(4): 863–878. DOI: 10.1007/s10723-020-09528-w.

[23] Lin Q, Adepu S, Verwer S, Mathur A. TABOR: A graphical model-based approach for anomaly detection in industrial control systems. In *Proc. the 2018 on Asia Conference on Computer and Communications Security*, May 2018, pp.525–536. DOI: 10.1145/3196494.3196546.

[24] Li D, Chen D C, Jin B H, Shi L, Goh J, Ng S K. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *Proc. the 28th Int. Con. Artificial Neural Networks*, Sept. 2019, pp.703–716. DOI: 10.1007/978-3-030-30490-4_56.

[25] Zhang Q, Zhou C J, Tian Y C, Xiong N X, Qin Y Q, Hu B W. A fuzzy probability Bayesian network approach for dynamic cybersecurity risk assessment in industrial control systems. *IEEE Trans. Industrial Informatics*, 2018, 14(6): 2497–2506. DOI: 10.1109/TII.2017.2768998.

[26] Yoon M K, Ciocarlie G F. Communication pattern monitoring: Improving the utility of anomaly detection for industrial control systems. In *Proc. the 2014 NDSS Workshop on Security of Emerging Networking Technologies*, Feb. 2014. DOI: 10.14722/sent.2014.23012.

[27] Kravchik M, Shabtai A. Efficient cyber attack detection in industrial control systems using lightweight neural networks and PCA. *IEEE Trans. Dependable and Secure Computing*, 2022, 19(4): 2179–2197. DOI: 10.1109/TDSC.2021.3050101.

[28] Elnour M, Meskin N, Khan K, Jain R. A dual-isolation-forests-based attack detection framework for industrial control systems. *IEEE Access*, 2020, 8: 36639–36651. DOI: 10.1109/ACCESS.2020.2975066.

[29] Lee H, Kwon H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Processing*, 2017, 26(10): 4843–4855. DOI: 10.1109/TIP.2017.2725580.

[30] Zheng H L, Fu J L, Mei T, Luo J B. Learning multi-attention convolutional neural network for fine-grained image recognition. In *Proc. the 2017 IEEE International Conference on Computer Vision*, Oct. 2017, pp.5219–5227. DOI: 10.1109/ICCV.2017.557.

[31] Xie X Z, Niu J W, Liu X F, Li Q F, Wang Y, Han J, Tang S J. DG-CNN: Introducing margin information into convolutional neural networks for breast cancer diagnosis in ultrasound images. *Journal of Computer Science and Technology*, 2022, 37(2): 277–294. DOI: 10.1007/s11390-020-0192-0.

[32] Yin Y F, Shah R R, Zimmermann R. Learning and fusing multimodal deep features for acoustic scene categorization. In *Proc. the 26th ACM International Conference on Multimedia*, Oct. 2018, pp.1892–1900. DOI: 10.1145/3240508.3240631.

[33] Abdoli S, Cardinal P, Lameiras Koerich A. End-to-end environmental sound classification using a 1D convolutional neural network. *Expert Systems with Applications*, 2019, 136: 252–263. DOI: 10.1016/j.eswa.2019.06.040.

[34] Kravchik M, Shabtai A. Detecting cyber attacks in industrial control systems using convolutional neural networks. In *Proc. the 2018 Workshop on Cyber-Physical Systems Security and PrivaCy*, Jan. 2018, pp.72–83. DOI: 10.1145/3264888.3264896.

[35] Liu J J, Yin L B, Hu Y, Lv S C, Sun L M. A novel intrusion detection algorithm for industrial control systems based on CNN and process state transition. In *Proc. the 37th IEEE International Performance Computing and Communications Conference*, Nov. 2018. DOI: 10.1109/PCCC.2018.8710993.

[36] Hu Y B, Zhang D H, Cao G Y, Pan Q. Network data analysis and anomaly detection using CNN technique for industrial control systems security. In *Proc. the 2019 IEEE International Conference on Systems, Man and Cybernetics*, Oct. 2019, pp.593–597. DOI: 10.1109/SMC.2019.8913895.

[37] Abdelaty M, Doriguzzi-Corin R, Siracusa D. DAICS: A deep learning solution for anomaly detection in industrial control systems. *IEEE Trans. Emerging Topics in Computing*, 2022, 10(2): 1117–1129. DOI: 10.1109/TETC.2021.3073017.

[38] Kusakina N M, Orlov S P, Kravets O J. Convolutional neural network for detecting anomalies in the control system of a machine-building enterprise. *IOP Conference Series: Materials Science and Engineering*, 2020, 862: 052020. DOI: 10.1088/1757-899X/862/5/052020.

[39] Salama M, El-Dakhakhni W, Tait M. Mixed strategy for power grid resilience enhancement under cyberattack. *Sustainable and Resilient Infrastructure*, 2022, 7(5): 568–588. DOI: 10.1080/23789689.2021.1974675.

[40] Milanović J V, Zhu W T. Modeling of interconnected critical infrastructure systems using complex network theory. *IEEE Trans. Smart Grid*, 2018, 9(5): 4637–4648. DOI: 10.1109/TSG.2017.2665646.

[41] Bruna J, Zaremba W, Szlam A, LeCun Y. Spectral networks and locally connected networks on graphs. In *Proc. the 2nd International Conference on Learning Representations*, Apr. 2014.

[42] Zhang Z W, Cui P, Zhu W W. Deep learning on graphs: A survey. *IEEE Trans. Knowledge and Data Engineering*, 2022, 34(1): 249–270. DOI: 10.1109/TKDE.2020.2981333.

[43] Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains. In *Proc. the 2005 IEEE International Joint Conference on Neural Networks*, Jul. 31-Aug. 4 2005, pp.729–734. DOI: 10.1109/IJCNN.2005.1555942.

[44] Scarselli F, Gori M, Tsoi A C, Hagenbuchner M, Monfardini G. The graph neural network model. *IEEE Trans.*

*Neural Networks*, 2009, 20(1): 61–80. DOI: 10.1109/TNN.2008.2005605.

[45] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. In *Proc. the 5th International Conference on Learning Representations*, Apr. 2017.

[46] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proc. the 30th Conference on Neural Information Processing Systems*, Dec. 2016, pp.3844–3852.

[47] Liang X D, Shen X H, Feng J S, Lin L, Yan S C. Semantic object parsing with graph LSTM. In *Proc. the 14th European Conference on Computer Vision*, Oct. 2016, pp.125–143. DOI: 10.1007/978-3-319-46448-0_8.

[48] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In *Proc. the 6th Int. Conf. Learning Representations*, May 2018.

[49] Zhao L, Song Y J, Zhang C, Liu Y, Wang P, Lin T, Deng M, Li H F. T-GCN: A temporal graph convolutional network for traffic prediction. *IEEE Trans. Intelligent Transportation Systems*, 2020, 21(9): 3848–3858. DOI: 10.1109/TITS.2019.2935152.

[50] Yan S J, Xiong Y J, Lin D H. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proc. the 32nd AAAI Conference on Artificial Intelligence*, Feb. 2018. pp.7444–7452.

[51] Stergiopoulos G, Theocharidou M, Kotzanikolaou P, Gritzalis D. Using centrality measures in dependency risk graphs for efficient risk mitigation. In *Proc. the 9th IFIP 11. 10 International Conference on Critical Infrastructure Protection*, Mar. 2015, pp.299–314. DOI: 10.1007/978-3-319-26567-4_18.

[52] Ricaud B, Borgnat P, Tremblay N, Gonçalves P, Vandergheynst P. Fourier could be a data scientist: From graph Fourier transform to signal processing on graphs. *Comptes Rendus Physique*, 2019, 20(5): 474–488. DOI: 10.1016/j.crhy.2019.08.003.

[53] Hammond D K, Vandergheynst P, Gribonval R. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 2011, 30(2): 129–150. DOI: 10.1016/j.acha.2010.04.005.

[54] Saxena A, Iyengar S. Centrality measures in complex networks: A survey. arXiv: 2011.07190, 2020. https://arxiv.org/abs/2011.07190, Jul. 2024.

[55] Das K, Samanta S, Pal M. Study on centrality measures in social networks: A survey. *Social Network Analysis and Mining*, 2018, 8(1): 13. DOI: 10.1007/s13278-018-0493-2.

[56] Landherr A, Friedl B, Heidemann J. A critical review of centrality measures in social networks. *Business & Information Systems Engineering*, 2010, 2(6): 371–385. DOI: 10.1007/s12599-010-0127-3.

[57] Tuğal İ, Karcı A. Comparisons of Karcı and Shannon entropies and their effects on centrality of social networks. *Physica A: Statistical Mechanics and its Applications*, 2019, 523: 352–363. DOI: 10.1016/j.physa.2019.02.026.

[58] Morelli S A, Ong D C, Makati R, Jackson M O, Zaki J. Empathy and well-being correlate with centrality in different social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 2017, 114(37): 9843–9847. DOI: 10.1073/pnas.1702155114.

[59] Leydesdorff L, Wagner C S, Bornmann L. Betweenness and diversity in journal citation networks as measures of interdisciplinarity—A tribute to Eugene Garfield. *Scientometrics*, 2018, 114(2): 567–592. DOI: 10.1007/s11192-017-2528-2.

[60] Ding Y, Yan E J, Frazho A, Caverlee J. PageRank for ranking authors in co-citation networks. *Journal of the American Society for Information Science and Technology*, 2009, 60(11): 2229–2243. DOI: 10.1002/asi.v60:11.

[61] Ji P S, Jin J S. Coauthorship and citation networks for statisticians. *The Annals of Applied Statistics*, 2016, 10(4): 1779–1812. DOI: 10.1214/15-AOAS896.

[62] Samad A, Arshad Islam M, Azhar Iqbal M, Aleem M. Centrality-based paper citation recommender system. *EAI Endorsed Trans. Industrial Networks and Intelligent Systems*, 2019, 6(19): e2. DOI: 10.4108/eai.13-6-2019.159121.

[63] Cickovski T, Peake E, Aguiar-Pulido V, Narasimhan G. ATria: A novel centrality algorithm applied to biological networks. *BMC Bioinformatics*, 2017, 18(Suppl 8): 239. DOI: 10.1186/s12859-017-1659-z.

[64] Koschützki D, Schreiber F. Centrality analysis methods for biological networks and their application to gene regulatory networks. *Gene Regulation and Systems Biology*, 2008, 2: 193–201. DOI: 10.4137/grsb.s702.

[65] Ashtiani M, Salehzadeh-Yazdi A, Razaghi-Moghadam Z, Hennig H, Wolkenhauer O, Mirzaie M, Jafari M. A systematic survey of centrality measures for protein-protein interaction networks. *BMC Systems Biology*, 2018, 12(1): 80. DOI: 10.1186/s12918-018-0598-2.

[66] Jayasinghe A, Sano K, Rattanaporn K. Application for developing countries: Estimating trip attraction in urban zones based on centrality. *Journal of Traffic and Transportation Engineering (English Edition)*, 2017, 4(5): 464–476. DOI: 10.1016/j.jtte.2017.05.011.

[67] Gao S, Wang Y L, Gao Y, Liu Y. Understanding urban traffic-flow characteristics: A rethinking of betweenness centrality. *Environment and Planning B: Urban Analytics and City Science*, 2013, 40(1): 135–153. DOI: 10.1068/b38141.

[68] Parmar A, Gnanadhas J, Mini T T, Abhilash G, Biswal A C. Multi-agent approach for anomaly detection in automation networks. In *Proc. the 2014 International Conference on Circuits, Communication, Control and Computing*, Nov. 2014, pp.225–230. DOI: 10.1109/CIMCA.2014.7057795.

[69] Opsahl T, Agneessens F, Skvoretz J. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social Networks*, 2010, 32(3): 245–251. DOI: 10.1016/j.socnet.2010.03.006.

[70] Bavelas A. Communication patterns in task-oriented groups. *The Journal of the Acoustical Society of America*, 1950, 22(6): 725–730. DOI: 10.1121/1.1906679.

[71] Freeman L C. A set of measures of centrality based on betweenness. *Sociometry*, 1977, 40(1): 35–41. DOI: 10.2307/3033543.

[72] Brandes U. A faster algorithm for betweenness centrality. *The Journal of Mathematical Sociology*, 2001, 25(2): 163–177. DOI: 10.1080/0022250X.2001.9990249.

[73] Hage P, Harary F. Eccentricity and centrality in net-

works. *Social Networks*, 1995, 17(1): 57–63. DOI: 10.1016/0378-8733(94)00248-9.

[74] Chen D B, Lü L Y, Shang M S, Zhang Y C, Zhou T. Identifying influential nodes in complex networks. *Physica A: Statistical Mechanics and its Applications*, 2012, 391(4): 1777–1787. DOI: 10.1016/j.physa.2011.09.017.

[75] Bonacich P. Factoring and weighting approaches to status scores and clique identification. *The Journal of Mathematical Sociology*, 1972, 2(1): 113–120. DOI: 10.1080/0022250X.1972.9989806.

[76] Stephenson K, Zelen M. Rethinking centrality: Methods and examples. *Social Networks*, 1989, 11(1): 1–37. DOI: 10.1016/0378-8733(89)90016-6.

[77] Goh J, Adepu S, Junejo K N, Mathur A. A dataset to support research in the design of secure water treatment systems. In *Proc. the 11th International Conference on Critical Information Infrastructures Security*, Oct. 2016, pp.88–99. DOI: 10.1007/978-3-319-71368-7_8.

[78] Ahmed C M, Palleti V R, Mathur A P. WADI: A water distribution testbed for research in the design of secure cyber physical systems. In *Proc. the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, Apr. 2017, pp.25–28. DOI: 10.1145/3055366.3055375.

**Jun Yang** received his B.S. degree in electronics and information engineering from Beijing University of Technology, Beijing, in 2016. He is currently pursuing his Ph.D. degree in signal and information processing with the National Network New Media Engineering Research Center, Chinese Academy of Sciences, Beijing. His research interests include industrial anomaly detection, network security, and machine learning.

**Yi-Qiang Sheng** received his Ph.D. degree in information and communications engineering from the Tokyo Institute of Technology, Tokyo, in 2014. He is currently with the National Network New Media Engineering Research Center, Chinese Academy of Sciences, Beijing, as an academic researcher and an associate professor. His research interests include smart systems, optimization algorithms, machine learning, big data, and network theory with applications.

**Jin-Lin Wang** received his M.S. degree in signal and information processing from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, in 1989. He is currently the director of the National Network New Media Engineering Research Center, Chinese Academy of Sciences, Beijing. His research interests include network media, digital signal processing, source coding and channel coding in IPTV, technologies of media streaming-based applications in networks, and wireless communications.

**Hong Ni** received his M.S. degree in signal and information processing from the Institute of Acoustics, Chinese Academy of Sciences, Beijing, in 1989. He is currently with the National Network New Media Engineering Research Center, Chinese Academy of Sciences, Beijing, as an academic researcher and a doctoral tutor. His research interests include broadband multimedia communication, network new media technology and application, embedded systems, and middleware technology.