Article

Identification of Influential Nodes in Industrial Networks Based on Structure Analysis

Tianyu Wang ^{1,2,3,*}, Peng Zeng ^{1,2,3,*}, Jianming Zhao ^{1,2,3,*}, Xianda Liu ^{1,2,3} and Bowen Zhang ^{1,2,3}

- ¹ State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China; wangtianyu@sia.cn (T.W.); liuxianda@sia.cn (X.L.); zhangbowen@sia.cn (B.Z.)
² Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China
³ Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
* Correspondence: zp@sia.cn (P.Z.); zhaojianming@sia.cn (J.Z.)

Abstract: Industrial network systems are facing various new challenges, such as increasing functional failure factors, the accelerating penetration of information threats, and complex and diverse attack methods. Industrial networks are often vulnerable to natural or intentional disasters; therefore, it is highly invaluable to research to identify the influential nodes. Most of the state-of-the-art evaluates the importance of the nodes according to one or more network metrics. Moreover, there are no metrics reflecting all the properties of the network. In this paper, a novel method (Structure-based Identification Method, SIM) to identify the influential nodes in industrial networks is proposed based on the network structure, which goes beyond the use of network metrics. The SIM method extracts the weakly connected components, which are more likely to survive after the important nodes are attacked in the network. Evaluation results show that the SIM method obtains better results than the state-of-the-art methods to identify influential nodes in real-world industrial networks and has a good prospect to be applied in industrial application.

Keywords: industrial networks; influential nodes; structure-based identification method

Citation: Wang, T.; Zeng, P.; Zhao, J.; Liu, X.; Zhang, B. Identification of Influential Nodes in Industrial Networks Based on Structure Analysis. *Symmetry* **2022**, *14*, 211. <https://doi.org/10.3390/sym14020211>

Academic Editor: Christos Volos

Received: 6 January 2022

Accepted: 21 January 2022

Published: 22 January 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of digital, networked, and intelligent control equipment, the industrial system has converted from a closed-island operation mode to an open interconnected operation mode, which has greatly improved the operation efficiency and effectiveness of the industrial system [1–3]. At the same time, intelligent control equipment is also facing new challenges, such as increasing functional failure factors, the accelerating penetration of information threats, and complex and diverse attack sources [4,5]. With the deep integration of 5G technology and industry, the industrial system is gradually evolving to a heterogeneous complex structure, which increases security constraints and gradually blurs security boundaries. For example, On 23 December 2015, the power systems in more than three regions of Ukraine were attacked by highly destructive malware, resulting in large-scale power outages [6]. At least 1.4 million people in the Ivano-Frankivsk region suffered power outages, which also affected the normal operation of many other real-world network systems. Only when the nodes that can seriously affect the structure and function of the network are identified and the robustness of the network is evaluated more accurately can this large-scale network failure be prevented fundamentally [7]. In real-world systems, the influential nodes are usually more vulnerable to disturbances. One example is that large relay stations with traffic congestion are more fragile to attacks, which will greatly affect the function of the whole network [8]. Therefore, identifying the influential nodes in the real-world network accurately can promote the effective evaluation of the robustness of the network system, which guarantees people's daily life.

To analyze the importance of the nodes, the most popular idea is to rank nodes in the network according to a certain network metric [9–11]. The advantage of the idea is that

the ranking process has very low computation complexity, while identifying the optimal influential nodes is an NP-hard problem. According to the background of influential node identification, the computation process on k influential nodes from a network of n nodes should include $C_n^k = \frac{A_n^k}{k!} = \frac{n!}{k!(n-k)!}$. Thus, to obtain the optimally influential nodes of all numbers less than n , the number of node sets should be $\sum_1^n C_n^k$, which is NP-hard to compute. The optimal influential nodes in this paper means that the failure of these nodes will damage the connectivity function of the network the most critically, which is measured by the component with the maximal number of connected nodes, called the giant component. For example, in power grid networks, the giant component means the maximum power supply area. After a careful review of relevant research, we found that most researchers chose to rank the nodes according to a certain metric while identifying influential nodes. However, no single network metric can guarantee the optimality of the node's importance.

In this paper, a novel method—the Structure-based Identification Method, SIM—for identifying influential nodes in complex networks is proposed. To identify the influential nodes, we extract the components, which have less influence to the network structure, and search for the influential nodes based on the components. We summarize the main contributions of our work as follows:

1. We address the problem of identifying influential nodes of a fixed number. It is shown that when the number varies, the optimal influential nodes change significantly. The influential nodes are identified not only from a certain network metric but are derived from the structure of the network.
2. We proposed a series of well-tuned heuristics for the method in order to improve the computation complexity of SIM, which allows us to apply the method to real-world industrial systems.
3. We evaluate the proposed method on real-world industrial networks, and the results show that our method can identify significantly better influential nodes than existing ranking-based methods.

The rest of this paper is organized as follows. In Section 2, we introduce related work on identifying influential nodes and discuss behaviors of ranking-based methods on a small industrial network. Section 3 provides the Structure-based Identification Method, SIM. In Section 4, an evaluation on some real-world industrial networks is shown. Finally, Section 5 gives the conclusions and future research.

2. Related Work

Before proceeding to the description of our algorithm, several commonly used ranking algorithms to attack network are introduced. In this paper, we research the problem of identifying influential nodes of a fixed number. Connectivity is one of the most important characteristics to measure the network structure and function, and the scale of the giant component is the most commonly used metric to evaluate connectivity. Therefore, the effect of a set of influential nodes is measured by the size of the giant component in absence of these nodes in the network.

After a careful review of the related work on the problem of identifying influential nodes in the network, it is found that most of the methods rely on the use of network metrics or extensions of these. Some of the methods and also their results are chosen for comparison with our algorithm in the latter part of this paper. For the reason that we are concerned with undirected networks, the algorithms to be introduced are all limited to undirected networks.

1. Degree-based method. Degree is the most direct measure of node centrality in complex network theory. The node i 's degree (deg_i) represents the number of links connecting node i . deg_i could be defined in terms of the adjacency matrix A , which is symmetric in undirected networks [12]:

$$deg_i = \sum_{j \in N} a_{ij} \quad (1)$$

As a method to identify the influential nodes, the degree-based method ranks the nodes descendingly according to the degree, and nodes with a larger degree are more important [13,14]. The degree-based methods have very low computation complexity, since the complexity to compute the degree of all nodes in the network is $O(|L|)$. This means that the computational time of degree-based methods is quite efficient even in real-world industrial networks with large scales.

2. Betweenness-based method. Betweenness is another popular and understandable network metric. Betweenness of nodes refers to the number of shortest paths through nodes in a network. Betweenness can represent the efficiency of transmitting information or material between nodes in the network. The node i 's betweenness is defined as [15]

$$b_i = \sum_{x,y \in N, x \neq y} \frac{n_{xy}(i)}{n_{xy}} \quad (2)$$

where n_{xy} indicates the number of shortest paths connecting x and y , and $n_{xy}(i)$ is the number of paths that pass through node i and are the shortest path. Similarly to degree-based methods, the nodes with larger betweenness are more important in betweenness-based methods [16]. Betweenness-based methods are more commonly used in real-world applications, because most real-world networks have the need to transmit information or material.

3. Closeness-based method. Closeness represents the reciprocal of the means of the distance between all other nodes in the network and node i . For every node in a network, it is easy to compute the average distance between the node and all the other nodes in the network, which is denoted by d_i and defined as

$$d_i = \frac{1}{N} \sum_{j=1}^{|N|} d_{ij} \quad (3)$$

where d_{ij} is the distance between i and j . Then, the reciprocal of d_i is called closeness, which is defined as [17]

$$c_i = \frac{1}{d_i} = \frac{N}{\sum_{j=1}^{|N|} d_{ij}} \quad (4)$$

4. CI-based method [18]. The Collective Influence (CI) is a ranking-based influential node identification method, which has shown good effectiveness on several real-world networks. The CI of node i is defined as

$$CI_l(i) = (deg_i - 1) \sum_{j \in Ball(i,l)} (deg_j - 1) \quad (5)$$

where l is the shortest path length between node i and j while $Ball(i,l)$ means the node set at a distance of l away from i . The CI-based method evaluates the importance of nodes in the network according to the formula, and a larger CI value means it is more influential. It should be noted that the CI value is equal to the degree value when $l = 0$.

Based on the related methods above, we evaluate the four methods on a small chemical network with only twenty-two nodes and twenty-five links in Figure 1, which is a Carbon-14 treatment technology research platform containing a hydrogen elimination unit, methane catalytic oxidation unit, and absorption unit. The nodes in the network represent the equipment in the research platform, and the links stand for the existence of transported chemicals between equipment. Figure 1 shows the most important four nodes' importance to the connectivity of the network according to each method. The most important four nodes would influence the size of the giant component (GC_{size}) most critically, which is the metric to measure the effectiveness of methods in this paper. We will observe GC_{size} after the four important nodes are removed, while a more effective identification method leads to a lower GC_{size} .

Figure 1. Example network (a), the result of an optimal identification (b), and the effect of different ranking-based identification methods (c–f). Identified nodes are crossed out. For each absence of identified nodes, we denote the size of the giant component in parentheses.

As we can see, the ranking-based methods above network metrics often do not lead to optimal solutions, which could be attributed to two reasons. The first reason is that no single metric can completely cover all aspects of node importance for the connectivity. While it was shown that the combination of network metrics can gradually improve results, the problem persists at its core. Another reason is that metrics can only obtain a node ranking, and important nodes of number l are created simply by extracting the l -prefix of the ranking, which means a larger number of important nodes must contain the smaller ones. However, in real-world systems, nodes that are important for a short length might turn out to be less relevant for longer length. This insight would be ignored by concentrating on metric rankings. For instance, Figure 2 presents the optimal solutions on a small example network with only 10 nodes of different numbers. Figure 2a is the initial example network, and Figure 2b–d show the most influential nodes of different numbers of 1 to 3. It can be seen that the optimal important nodes of one more does not simply have a previous solution as a prefix. In other words, ranking-based methods can hardly obtain effective solutions, when the number of to-be-identified nodes varies.

Although most research is based on ranking metrics, there are several methods not induced by metric rankings. Some research considers robustness of networks with localized attacks, in which a group of neighboring nodes are identified [19,20]. In another study, the robustness of the route network is analyzed and an inverted adaptive method is proposed to anticipate network breakdown [21]. In the inverted adaptive strategy, nodes whose activation maximizes network efficiency are activated first. The network efficiency is defined as

$$E = \frac{1}{|N|(|N| - 1)} \sum_{i \neq j} \frac{1}{d_{ij}} \quad (6)$$

where $\frac{1}{d_{ij}}$ is the length of the shortest path between node i and j . Inspired by these methods, a network not based on ranking metrics is focused on.

Figure 2. Example network (a); the best identification strategies of different length of from 1 to 3 (b–d). Identified nodes are crossed out. For each absence of identified nodes, we denote the size of the giant component in parentheses.

3. Proposed Methods

Given the limitation of existing ranking-based methods to identify optimal important nodes, we proposed to identify influential nodes without prioritizing the importance of nodes beforehand. Since the robustness of network is measured by the size of the giant component in this paper, we design a method to identify interesting node subsets, which cut a network into components. Generally, our method can be seen as an extension of cut-nodes, which identifies single nodes that separate a network into at least two components [22]. However, since in real-world networks cut nodes are rather uncommon, we are interested in cut-node-sets in this paper. The cut-nodes method could be seen as a special case of our method, where the candidate influential nodes only focus on those containing one node. Although a ranking-free method is expected, the ranking concept is also referred to for node-sets' construction and selection processes. The limitations shown above reveal that ranking-based methods could hardly obtain optimal influential nodes but network metrics are indeed a practical and efficient way to measure the importance of nodes and to exploit its structure, where nodes with larger metric values are more possible in the optimal influential nodes.

We identify interesting subcomponents by performing a variant of Breath-First Search (BFS) from each node in the network. The reason we start from each node in the search process is that the probability of each node to be in the weak components should be symmetrical, which means that each node should be the origin of the search. At each step of BFS, we pick the neighbor with the smallest degree as the next node to be visited. Before discussing the method in detail, we visualize the idea in Figure 3. We start with a random node: in the case of Figure 3a, we start with node *j*. The neighbors of *j* are *g*, *h*, *k*, and *l*. This means that if it is allowed to remove at least four nodes, we could extract a component of size one (the component consisting only of node *j*) from the network. We keep track of all these components and their cut nodes during the execution of our method. Starting from the component containing node *j*, we select the component's neighbor with the lowest degree (here, node *k*; when at least two nodes have the lowest degree, we choose the first one according to the order of nodes stored in the network) and add it to the component (Figure 3b). Since nodes with lower metric values have more possibility to not be in the optimal identified nodes and degree is the metric with the lowest computational complexity, the node with the lowest degree is selected here. If it is allowed to remove three nodes, we could cut out the component of size two (the component consisting of nodes *a* and *b*). We proceed with our component-discovering strategy until we have visited all nodes (Figure 3p). During this process, as shown in Figure 3f, we identify that, among other interesting node sets, removing node *b*, *g*, and *h* from the network causes a significant

reduction in the size of the giant component, if only three important nodes are expected. We repeat the same procedure with each node as a start node, and thus, discover a huge number of interesting node-sets.

Figure 3. Steps I–XVI (a–p) visualize the process of finding the components of node j . Nodes filled with blue color belong to the current component; nodes in gray color are neighbors of the nodes in the current component; gray-colored nodes with a thick black line are those neighbors with the smallest degree among all neighbors, which means that they should be put into the component's next step.

We have identified interesting candidate influential nodes, following the procedure outlined above. However, in the process, we create many candidate nodes for each $i \leq k$ and need to check all possible combinations to create a complete length of important nodes. Given the large number of candidates, the processing is time-consuming. Therefore, we proposed a filtering process to limit the number of candidates. Once we have obtained the filtered, interesting candidates, these candidates will be combined into a complete solution to important nodes of length k . To show our algorithm's details clearly, we demonstrate the process as a pseudocode in Algorithm 1. The SIM method consists of three building blocks: (1) Create candidate influential nodes; (2) Filter candidate influential nodes; (3) Select final influential nodes.

Algorithm 1 Structure-based Identification Method, SIM

Input: Graph G ; number of nodes to be identified k
Output: List of identified influential nodes
 // Create candidate influential nodes

- 1: Let *candidateInfluentialNodes* be a map from N to sets of nodes
- 2: Let *subsets* be a map from sets of nodes (current component) to sets of nodes (candidate influential nodes)
- 3: **for all** $n \in \text{nodes}(G)$ **do** *Create candidate influential nodes from each node as the start.*\
 - 4: $\text{subset}[n] = \text{neighbors}(n)$, where $\text{neighbors}(n)$ are sorted with increasing degree
 - 5: **if** $|\text{neighbors}(n)| \leq k$ **then**
 - 6: Add $\text{neighbors}(n)$ into *candidateInfluentialNodes*[$\text{subset}[n]$]
 - 7: **end if**
- 8: **end for**
- 9: **while** $|\text{subsets}| > 0$ **do**
 - 10: Let *subsets** be a map from sets of nodes (current component) to sets of nodes (candidate influential nodes)
 - 11: **for all** $x \in \text{subsets}$ **do**
 - 12: **if** $|\text{subsets}[x]| > 0$ **then**
 - 13: Let $\text{comp} = \text{subsets}[x] \cup \{\text{subsets}[x]\}$
 - 14: Let *newneighbors* be a list of all nodes in $\text{neighbors}(\text{comp})$, sorted according to their increasing degree
 - 15: Let $\text{subsets}^*[\text{comp}] = \text{newneighbors}$
 - 16: **if** $|\text{newneighbors}| \leq k$ **then**
 - 17: Add *newneighbors* into *candidateInfluentialNodes*[newneighbors]
 - 18: **end if**
 - 19: **end if**
 - 20: **end for**
 - 21: Let *subsets* = *subsets**
 - 22: **end while**
 - 23: // Filter candidate influential nodes
 - 24: Let *filteredInfluentialNodes* be a map from N to sets of nodes
 - 25: **for all** $s \in \text{domain}(\text{candidateInfluentialNodes})$ **do** *Each element of *candidateInfluentialNodes* is the candidate influential node set with a fixed number.*\
 - 26: Let *a_deg* be the node-set in *candidateInfluentialNodes*(s) with the highest average degree
 - 27: Let *a_betw* be the node-set in *candidateInfluentialNodes*(s) with the highest average betweenness
 - 28: Let *a_GC* be the node-set in *candidateInfluentialNodes*(s) with the highest effectiveness (in terms of $\text{GC} - \text{size}$)
 - 29: Set *filteredInfluentialNodes*(s) = *a_deg, a_betw, a_GC*
 - 30: **end for**
 - 31: // Select final influential nodes
 - 32: Let *bestNodeSet* = $\{\}$
 - 33: Let *bestNodeSetGCSize* = $|\text{nodes}(G)|$
 - 34: **for all** $s_1 \in [0, 1, 2, \dots, k/2]$ **do** *Only consider dividing k into two parts in SIM method.*\
 - 35: Let $s_2 = k - s_1$
 - 36: **for all** $a_1 \in \text{filteredInfluentialNodes}(s_1)$ **do**
 - 37: **for all** $a_2 \in \text{filteredInfluentialNodes}(s_2)$ **do**
 - 38: Let *GCSize* be the size of the giant component after removing $\{a_1\} \cup \{a_2\}$ in G
 - 39: **if** $\text{GCSize} < \text{bestNodeSetGCSize}$ **then**
 - 40: $\text{bestNodeSetGCSize} = \text{GCSize}$ *Select the best node set leading to the smallest *GCSize*.*\
 - 41: $\text{bestNodeSet} = \{a_1\} \cup \{a_2\}$
 - 42: **end if**
 - 43: **end for**
 - 44: **end for**
 - 45: **for all** $i < k$ **do**
 - 46: Do the same steps as above. Compared with *bestNodeSetGCSize*. Record the better one.
 - 47: **end for**
 - 48: **if** $|\text{bestNodeSet}| = k$ **then** *If there are the same influential nodes in both the two combined subinfluential node sets, add the nodes of the fixed number with the best effectiveness.*\
 - 49: $\text{bestNodeSet} = \text{bestNodeSet} \cup \text{filteredInfluentialNodes}(k - |\text{bestNodeSet}|)$
 - 50: **end if**
 - 51: Return *bestNodeSet*

Block 1: Create candidate influential nodes: Construct a set of candidate influential nodes, from which we will choose components' neighbors to combine the final solution. We iterate over all nodes in the network. In other words, block 1 should be operated on each node of the network. For each node, we begin with the node and put it into a set (current component). Then, we create another set to record the neighbors of the the set (current component). If the number of the neighbors is less than or equal to k (the number of important nodes we plan to identify), we put the neighbors into the candidate set as a

candidate influential node-set. Once we remove the neighbors, we can at least obtain the component above. Then, we choose a neighbor with the lowest degree from the neighbors. The rationale for selecting by degree is we conjecture that this kind of neighbor has more possibility to be in one of the final optimal components, since, though degree could not lead to optimal solution, it is indeed an efficient measurement of node importance. We add the chosen node into the component and proceed. We redo these steps above until the component set covers the whole network. This procedure is repeated for all nodes. Through this searching process, the computational complexity of generating candidate influential nodes is reduced to $O(|N|^2)$, while the complexity of all possible candidates is $O(2^{|N|})$.

Block 2: Filter candidate influential nodes: In the first block, we create many candidate influential node-sets for each $i \leq k$ and check all possible combinations to create a complete influential node. Given that the sizes of real-world networks often have numerous nodes, which will result in a large number of candidates, the process of creating all combinations of candidates of influential nodes is time-consuming. Therefore, we propose to limit the number of candidates for each $i \leq k$. We only choose three selected candidate influential nodes: The one with the largest average degree; the one with the largest average betweenness; and the one with the highest effectiveness (the network becomes one with the smallest size of the giant component in the absence of the candidates). It is worthy to note that network metrics are exploited here for filtration of effective candidate influential nodes. As discussed before, although network metrics do not often lead to optimal influential nodes, they can indeed be a way to measure the importance of nodes. In this step, we believe the nodes with larger metric values are more likely to occur in the optimal solutions. With this point, we sacrifice possibly good candidate solutions for the sake of a significantly reduced running time. During this block, the computation complexity of obtaining the degree list and effectiveness list is $O(|N|)$, while that of computing the betweenness list is $O(|N||L|)$ ($|L|$ is the number of links) [23].

To illustrate the effectiveness of the filtration process, we still take an example run on Figure 1a. Table 1 presents the candidate influential nodes after “Create candidate influential nodes” for $k = 3$, when the optimal selection of important nodes is b, g, h . For each line of the candidate influential nodes, the red tuples are preserved after the “Filter candidate influential nodes”. As Table 1 shows, we can also obtain the optimal important influential nodes (b, g, h) of this network for $k = 3$ through combining (b) and (g, h) . At the same time, the final combination number of the important nodes can be decreased from 546 to 12.

Table 1. Effectiveness of filtration process.

Number of Nodes	Set of Nodes
1	$(a), (b), (h), (j), (l), (m), (n), (q), (r), (s), (t), (u), (v)$
2	$(s, v), (c, g), (b, j), (m, o), (q, s), (h, q), (b, h), (r, u), (q, t), (q, v),$ $(b, g), (h, r), (h, p), (j, n), (r, t), (c, f), (r, v), (b, c), (h, s), (d, g),$ $(b, e), (p, v), (j, o), (l, n), (s, u), (g, h), (h, u), (c, e), (t, v), (b, f),$ $(h, v), (h, t), (h, i), (e, g), (q, u), (d, f), (j, m), (b, d), (p, r)$
3	$(e, i, j), (g, h, u), (c, i, k), (b, g, m), (b, h, m), (a, h, i), (g, h, k), (c, h, k),$ $(b, h, o), (b, h, p), (g, h, l), (b, h, t), (b, g, j), (b, h, s), (g, h, n), (b, h, j),$ $(f, h, j), (b, g, h), (b, h, u), (c, h, j), (d, i, j), (b, g, o), (g, h, t), (g, h, m),$ $(g, h, q), (e, h, j), (g, h, r), (g, h, o), (g, h, p), (b, g, n), (g, h, s), (b, h, n),$ $(g, h, j), (c, i, j), (b, h, q), (f, i, j), (b, h, r), (b, h, v), (d, h, j)$

Block 3: Select final influential nodes: For the design of complete influential nodes of length of k , we combine several candidates with lengths smaller than k . For each decomposition of k , there are several addends to consider $k = k_{d1} + k_{d2} + \dots + k_{dr}$, $r \in \{1, 2, \dots, k\}$. For each of these k -partition elements, we select a candidate influential node with the same length as the addend number. During this process, we keep track of the best solution seen

so far (i.e., yielding the smallest giant component). However, if we consider all partitionings of k into addends, the number of these partitions grows quite fast with k , which makes it hard to be used in real-world industrial networks. Thus, in order to avoid a long running time for large k , we have to further restrict the number of partitioning. In this work, we only consider partitionings of size 2, i.e., we only split up k into two addends. Of course, longer partitionings lead to more effective solutions, but in order to make it practical for large-scale applications, the most efficient case (two partitionings) is considered.

Since the SIM method searches for candidate influential nodes of some fixed lengths $(1, 2, \dots, k)$, the computational time is influenced by the network structure. For networks with different structures, the maximum numbers of neighbors obtained after “Create candidate influential nodes” are different, denoted by a_{max} . For instance, a_{max} in star networks should be $|N| - 1$, and a_{max} in line networks are always 2. Based on the three blocks shown above, the computational complexity of SIM is approximately $O(k|N||L|)$ when k is smaller than a constant (a_{max}) of a network because most of the running time is spent in the filtering process, where the complexity to calculate the betweenness is much higher than that to calculate the size of giant component and degree. When k is larger than a_{max} , the time complexity of SIM is nearly $O(a_{max}|N||L|)$.

Following these three blocks, we indeed identify the optimal influential nodes for the example in Figure 1a. However, it should be noted that our method cannot guarantee optimality in a general case, since we only look at a restricted number of components, as induced by the smallest degree.

4. Experiments and Results

In this section, the effectiveness of SIM method is evaluated on real-world industrial networks compared with popular ranking-based methods. To show the performance of SIM on industrial networks, the size of the giant component and the robustness with SIM method and other ranking-based methods are discussed. The definition of the robustness is [24]

$$R = \frac{1}{|N|} \sum_{|I|=1}^{|N|} GC_{size}(I) \quad (7)$$

The metric of Robustness considers the influence of all possible absence of important nodes and records the size of the giant component, which keeps track on the connectivity function of networks even before collapsing. The normalization factor $\frac{1}{|N|}$ ensures that the measure can be compared among networks with different sizes.

4.1. Datasets

In order to evaluate the effectiveness of our method, real-world industrial networks (power bus networks) are used in this paper. Power bus network, which is the only bus technology supporting high-power load and high-speed communication in the industry, is a typical kind of real-world industrial network. In power bus networks, the nodes represent high-voltage or low-voltage equipment, while there is a link between two nodes only if electric current or information transmission exists. Table 2 shows the basic statistics for the power bus networks used in our paper [25], where the community structure is detected by Louvain method [26] and Figure 4 visualizes the three networks, where the size of a node is proportional to its degree. As shown in the figure, there is no obvious difference in power bus network structure, while the node and link numbers are quite different. The hub nodes are clearly visible in the network structure, where a network with a larger scale commonly has more hub nodes. It is notable that although the power bus 3 network has the largest network density—defined as $\frac{|L|}{|N|^2}$, reflecting the connection strength between nodes in the network—it has the largest shortest path length among the three power bus networks.

Table 2. Basic statistics for three real-world industrial networks.

Metrics	Power Bus 1	Power Bus 2	Power Bus 3
Node Number	494	662	685
Link Number	585	905	1281
Ave. Deg	2.37	2.73	3.74
Ave. Bet	0.019	0.014	0.017
Ave. Clo	0.095	0.098	0.080
Ave. SPL	10.57	10.25	12.44
Diameter	26	25	26
CC	0.042	0.046	0.17
Community Number	18	19	15

Power bus 1: Power bus network with 494 nodes **Power bus 2:** Power bus network with 662 nodes **Power bus 3:** Power bus network with 685 nodes**Figure 4.** Real-world industrial networks used in this study. The size of nodes is proportional to its degree in the network. The graph layout was created with the force-directed algorithm Fruchterman–Reingold.

4.2. Experimental Results

To evaluate the effectiveness of our method on the real-world industrial networks, we use six methods (five competing methods and SIM method) to identify important nodes in the networks and to see the influence these important nodes have on network robustness. Among the six methods, “DEG”, “BETW”, “CLOS”, and “CI2” (Figure 5) show the effectiveness of the six methods on the real-world industrial networks. “RND10” means that the important nodes are identified randomly ten times, and the best important nodes are selected in the experiments. In the experiments, the six methods (SIM, “DEG”, “BETW”, “CLOS”, “CI2”, “RND10”) are applied to identify influential nodes at a number from 0 to the whole number of nodes in the networks, and the sizes of giant component are recorded when different numbers of influential nodes are invalid. As different values of l in the CI-based method can lead to different results and the optimal value cannot be obtained beforehand, we set l to be 2 in the evaluation part, which is the same as that in [18]. In Figure 5, the ordinate represents the ratio of the scale of the giant component in the network when influential nodes are invalid to the node number of the initial network (i.e., the identification effectiveness of the method), and the abscissa represents the ratio of the number of invalid influential nodes to the node number of the initial network.

Figure 5. Relative size of giant components in the three real-world industrial networks calculated as k increases by six methods.

As we can see, in the three industrial networks, our method could effectively identify the most influential nodes among the six methods, which reduces the size of the giant component at the fastest speed. Through the statistics, it can be found that in the three power bus networks, our method can decrease the size of the giant component at the fastest speed, while the random method usually obtains the worst identification of the influential nodes. Since the three networks have various structures and characteristics, the behaviors of these six methods' effectiveness on the networks are quite different. Through observing the curve trajectory of each method in different networks, the "power bus 1" network would be completely destroyed by four of the six methods, where only nearly 10% of important nodes are removed; in the "power bus 2" network, the size of the giant component could be decreased to nearly 0% in the absence of 20% of important nodes under most of the six methods; the "power bus 3" network is totally destroyed when 40% of important nodes are removed by most of the methods. In the majority of the whole range from 0% to 100% of the whole network, SIM is better than the others. Especially in the first 20% of the range, SIM can destroy the network to a very low size of giant component. It is notable that the closeness-based method shows worse behaviors than the random method when the number of important nodes is larger than 40% in all of the three power bus networks. Among the six methods, our method can always obtain the most effective results, and other methods could be sorted as CI-based method, degree-based method, betweenness-based method, closeness-based method, and random method, in descending order of the effectiveness of importance identification. It can be concluded that no matter the network with the largest robustness such as "Power Bus 3" or the network with low robustness such as "Power Bus 1", SIM has obvious superiority over the others.

To display the effectiveness of the methods for different influences on the industrial networks, Figure 6 shows the number of invalid nodes with the remaining size of giant components at 80%, 60%, 40%, and 20%. It can be concluded from the figures that our method behaves the best in 11 of the 12 measuring points, except for that in "power bus 2" network at 20% of the size of the giant component. Similar to the discussion for Figure 5, the closeness-based method obtains worse results as the size of the giant components becomes smaller. The behaviors of our method and the CI-based method are quite stable compared with other methods, which are obviously better than other methods. It is interesting that in Figure 6 of the "power bus 3" network, the degree-based methods behave worse than the random method, which inflects the significance to study identification methods of influential nodes.

Figure 6. Number of invalid nodes with the remaining size of giant components at 80%, 60%, 40%, and 20% in the three real-world industrial networks.

Table 3 presents the robustness values of the three industrial networks under six influential node identification methods. Compared with ranking-based methods, SIM has a great advantage in measuring the robustness of power bus networks, showing that our method can identify the influential nodes, which have a more serious impact on the industrial networks, and then help the relevant departments to judge the robustness of the real-world networks more accurately. In the power bus networks, the CI-based method behaves better than the other four methods. Among other ranking-based methods, the betweenness-based method can obtain better solutions. This phenomenon is related to the construction structure of the power bus networks. It is worthy to note that among the ranking-based methods, the best of them are often not the same for different networks, which addresses the limitation of ranking-based methods. As a result, our method is more effective in identifying influential nodes in industrial networks than popular ranking-based methods.

Table 3. Robustness values of real-world industrial networks under six methods.

Methods	Power Bus 1	Power Bus 2	Power Bus 3
RND10	0.17	0.23	0.22
Deg	0.060	0.091	0.21
Betw	0.057	0.13	0.16
Clos	0.14	0.24	0.24
CI2	0.048	0.080	0.14
SIM	0.039	0.063	0.09

5. Conclusions

In this paper, we designed a network influential node identification method, based on efficient extraction of weak components. Compared to related work, our method (SIM) does not rely on ranking nodes according to their importance in the network. We evaluate our method on three industrial networks with different parameters. Our method shows effective performance on all of the networks in terms of giant components and identifying the influential nodes. Our method can provide more accurate guidance for the design and operation of industrial networks, such as which influential nodes need to be allocated more supporting resources to ensure more-stable operation of the network. SIM provides a trade-off between fast identification process and effectiveness design, and thus, makes an important contribution to future industrial network robustness estimation.

In future research, our method can be improved in the following ways:

1. Identify weak components more intelligently. In the current method, the first part is to construct the list of components and their neighbors but there are many repeated

components when counting; therefore, if some components or some sequences are identified to be in the final network with low probability, they should be discarded early.

2. Divide k into partitions adaptively. In SIM method, the number of influential nodes is divided into two parts. However, as the size of the network becomes larger, how to decide the number of parts into which the k should be split is considerable. Such a decision should be made by taking into account local and global network properties.

Author Contributions: Conceptualization, T.W.; methodology, T.W.; software, T.W., J.Z. and X.L.; validation, T.W., P.Z. and B.Z.; data curation, X.L. and B.Z.; writing—original draft preparation, T.W.; writing—review and editing, T.W., P.Z. and J.Z.; visualization, T.W.; supervision, P.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Program of China “Key technology and application of safety and credibility for industrial control system” (2020YFB2009500), Independent Subject of State Key Laboratory of Robotics “Research on security industry network construction technology for 5G communication” (No. 2022-Z13).

Data Availability Statement: All data is available with an open source license at <https://networkrepository.com/power.php> (assessed on 2 January 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

- Fitó, J.; Ramousse, J.; Hodencq, S.; Wurtz, F. Energy, exergy, economic and exergoeconomic (4E) multicriteria analysis of an industrial waste heat valorization system through district heating. *Sustain. Energy Technol. Assess.* **2020**, *42*, 100894. [\[CrossRef\]](#)
- Lu, C.; Saifullah, A.; Li, B.; Sha, M.; González, H.; Gunatilaka, D.; Wu, C.; Nie, L.; Chen, Y. Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems. *Proc. IEEE* **2016**, *104*, 1013–1024. [\[CrossRef\]](#)
- Robles-Durazno, A.; Moradpoor, N.; McWhinnie, J.; Russell, G.; Porcel-Bustamante, J. Implementation and Evaluation of Physical, Hybrid, and Virtual Testbeds for Cybersecurity Analysis of Industrial Control Systems. *Symmetry* **2021**, *13*, 519. [\[CrossRef\]](#)
- Lee, J.; Kang, J.; Jun, M.-S.; Han, J. Design of a Symmetry Protocol for the Efficient Operation of IP Cameras in the IoT Environment. *Symmetry* **2019**, *11*, 361. [\[CrossRef\]](#)
- Zhang, H.; Peng, C.; Ling, S.; Chen, J. Optimal DoS Attack Scheduling in Wireless Networked Control System. *IEEE Trans. Control Syst. Technol.* **2015**, *24*, 1. [\[CrossRef\]](#)
- Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z. The 2015 Ukraine Blackout: Implications for False Data Injection Attacks. *IEEE Trans. Power Syst.* **2017**, *32*, 3317–3318. [\[CrossRef\]](#)
- Yue, Z.; Peng, Y.; Long, K. Improving robustness against the coordinated attack by removing crashed hub nodes in complex network. In Proceedings of the Network Architectures, Management, & Applications VII, Shanghai, China, 2–6 November 2009.
- Netter, B. Port and transportation development. *J. Coast. Res.* **2005**, 159–178.
- Zhu, J.; Wang, L. Identifying Influential Nodes in Complex Networks Based on Node Itself and Neighbor Layer Information. *Symmetry* **2021**, *13*, 1570. [\[CrossRef\]](#)
- Zhong, L.; Liu, J.; Shang, M. Iterative resource allocation based on propagation feature of node for identifying the influential nodes. *Phys. Lett. A* **2015**, *23*, 2272–2276. [\[CrossRef\]](#)
- Zareie, A.; Sheikahmadi, A.; Fatemi, A. Influential nodes ranking in complex networks: An entropy-based approach. *Chaos Solitons Fractals* **2017**, *104*, 485–494. [\[CrossRef\]](#)
- Diestel, R. *Graph Theory*, 3rd ed.; Springer: Berlin, Germany; New York, NY, USA, 2000.
- Sun, S.; Wu, Y.; Ma, Y.; Wang, L.; Gao, Z.; Xia, C. Impact of Degree Heterogeneity on Attack Vulnerability of Interdependent Networks. *Sci. Rep.* **2016**, *6*, 32983. [\[CrossRef\]](#) [\[PubMed\]](#)
- Wang, J.-W.; Rong, L.-L.; Zhang, L. Effect of attack on scale-free networks due to cascading failure. *Mod. Phys. Lett. B* **2009**, *23*, 1577–1587. [\[CrossRef\]](#)
- Freeman, L.C. A Set of Measures of Centrality Based on Betweenness. *Sociometry* **1977**, *40*, 35–41. [\[CrossRef\]](#)
- Puzis, R.; Yagil, D.; Elovici, Y.; Braha, D. Collaborative attack on Internet users’ anonymity. *Internet Res.* **2009**, *19*, 60–77. [\[CrossRef\]](#)
- Alex, B. Communication Patterns in Task-Oriented Groups. *J. Acoust. Soc. Am.* **1950**, *22*, 725–730.
- Morone, F.; Makse, H.A. Influence maximization in complex networks through optimal percolation. *Nature* **2015**, *524*, 65–68. [\[CrossRef\]](#)
- Berezin, Y.; Bashan, A.; Danziger, M.M.; Li, D.; Havlin, S. Localized attacks on spatially embedded networks with dependencies. *Sci. Rep.* **2015**, *5*, 8934. [\[CrossRef\]](#)
- Shao, S.; Huang, X.; Stanley, H.E.; Havlin, S. Percolation of localized attack on complex networks. *New J. Phys.* **2015**, *17*, 023049. [\[CrossRef\]](#)
- Lordan, O.; Sallan, J.M.; Simo, P.; David, G. Robustness of airline alliance route networks. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *22*, 587–595. [\[CrossRef\]](#)

-
22. Albertson, M.O.; Berman, D.M. The number of cut-vertices in a graph of given minimum degree. *Discret. Math.* **1991**, *89*, 97–100. [[CrossRef](#)]
 23. Brandes, U. A faster algorithm for betweenness centrality. *J. Math. Sociol.* **2001**, *25*, 163–177. [[CrossRef](#)]
 24. Schneider, C.M.; Moreira, A.A., Jr.; Havlin, S.; Herrmann, H. Mitigation of malicious attacks on networks. *Proc. Natl. Acad. Sci. USA* **2011**, *108*, 3838–3841. [[CrossRef](#)] [[PubMed](#)]
 25. Rossi, R.A.; Ahmed, N.K. The Network Data Repository with Interactive Graph Analytics and Visualization. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015.
 26. Blondel, V.D.; Guillaume, J.L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *2008*, P10008. [[CrossRef](#)]