GraphCCI: Critical Components Identification for Enhancing Security of Cyber-Physical Power Systems

Yigu Liu, Alexandru Ştefanov, *Member*, *IEEE*, Ioannis Semertzis, *Student Member, IEEE*, and Peter Palensky, *Senior Member*, *IEEE*

***Abstract*— Cyber security risks are emerging in Cyber-Physical power Systems (CPS) due to the increasing integration of cyber and physical infrastructures. Critical component identification is a crucial task for the mitigation and prevention of catastrophic blackouts. In this paper, we propose a novel method using graph data mining for critical CPS components identification named GraphCCI. First, it defines two categories of component correlations to reveal the cascading features of CPS. GraphCCI maps cascading failure datasets under time-varying operational states into weighted cascading graphs and constructs a graph database for graph data mining. By adopting graph data mining techniques, frequent subgraphs are identified to construct the Cascading Characteristics Graph (CC-Graph). Finally, the Node Criticality Index (NC-Index) is proposed to quantify the criticality of each CPS component. The experimental results on the IEEE 39-bus system verify the effectiveness of the proposed method and present an in-depth analysis of the CPS cascading features.**

***Index Terms*—Cyber-physical systems, critical component identification, graph theory, smart grids.**

I. Introduction

DUE to the rapid integration of cyber and physical infrastructures, modern power systems are becoming more efficient while also exhibiting increased vulnerabilities. This emerging risk was starkly demonstrated by the three major cyber attacks on the Ukrainian power grid in 2015, 2016, and 2022 [1]-[3], underscoring the critical need for enhanced security measures in this landscape. The evolving communication infrastructures have significantly altered the propagation mechanisms of cascading failures in the Cyber-Physical power Systems (CPS) [4]. These changes present novel challenges in ensuring safe system operation. Consequently, it is imperative to thoroughly investigate the new cascading mechanisms and pinpoint the critical components of CPS, which will enable the implementation of timely mitigation strategies, thereby enhancing the overall security and resilience of CPS.

In recent years, growing attention has been paid to the system resilience of CPS, with an emphasis on extreme events, e.g., severe weather [5][6], and cyber attacks [7][8]. Identifying the critical components plays a crucial role in enhancing the overall system resilience. It offers crucial insights to system operators during decision-making, particularly when defensive resources for the system are limited. The current methods for identifying critical CPS components can be broadly grouped into three categories: (i) topology-based, (ii) model-based, and (iii) machine learning-based methods. In general, the topology-based methods focus more on the nature of CPS as an interdependent network. In this sense, the percolation theory [9] and complex network theory [10] are adopted to prioritize the components based on the network parameters, e.g., node degree and betweenness. On the other hand, the model-based methods consider the system operation models, e.g., power and information flow models, and evaluate the criticality of each component based on the system operational data, e.g., stability analysis [11]-[13], historical cascading failure data [14]-[16]. Lastly, machine learning-based methods tend to train and learn the system features from the historical data, e.g., cascading failure data [17]-[19], where graph neural networks [18], reinforcement learning [20][21], and data mining algorithms [7] are used to extract the system features and identify the vulnerable CPS components.

The current literature has yielded fruitful results in identifying critical CPS components, yet each methodological category has notable limitations. Topology-based methods partially unravel network structural features but overlook the complexity models [22] and heterogeneity [23] inherent in CPS as industrial systems, potentially skewing identification results. Both model-based and machine-learning approaches consider CPS's operational facets, analyzing historical data to discern inter-component correlations. However, these methods typically extract correlations solely from the known data. Although some works consider different operational states, no historical data can cover all possible system conditions and capture all possible correlations between components. In these two categories, commonly employed statistical methods, like machine learning algorithms and graph theory indices, are limited to quantifying correlations presented in the historical data. This process overlooks latent correlations under unrepresented operational states, introducing significant bias in

This work was supported in part by the EU Horizon Europe eFORT Project with Grant Agreement No 101075665, Dutch Research Council's RESCUE Project with Grant Agreement NWO ESI.2019.006, and China Scholarship Council. (*Corresponding author: Yigu Liu.*)

Yigu Liu, Alexandru Ştefanov, Ioannis Semertzis, and Peter Palensky are with the Department of Electrical Sustainable Energy, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mails: {Y.Liu-18, A.I.Stefanov, I.Semertzis, P.Palensky}@tudelft.nl).

identifying critical components. Therefore, this paper aims to introduce a methodology that not only analyzes apparent component correlations but also quantifies latent ones, ensuring more accurate and realistic identification outcomes.

To address the issues mentioned above, we propose a critical components identification model named GraphCCI. It comprehensively models the cascading failure features under various operational states into cascading graphs and forms a weighted cascading graph database. Furthermore, a graph data mining algorithm, namely TKG [24], is adopted to mine the frequent subgraphs from the constructed database. Finally, we proposed the Node Criticality Index (NC-Index), considering both manifest and latent correlations of components to identify the critical components. The major contributions of this paper are summarized as follows:

(1) We define two correlations, i.e., manifest and latent correlations, to better reveal the cascading mechanism of CPS and comprehensively investigate the apparent and potential correlations between CPS components.

(2) We propose a set of definitions to map the historical cascading failures datasets into weighted cascading graphs, and then construct the weighted cascading graph database for graph data mining to thoroughly capture the cascading features of CPS.

(3) By jointly considering the manifest and latent correlations and the graph data mining results, we construct the Cascading Characteristics Graph (CC-Graph) and propose the NC-Index to quantify the criticality of each node in the CC-Graph.

The remainder of this paper is organized as follows: Section II describes the cyber-physical cascading model and constructs the weighted cascading graph database considering time-varying operational states. Section III presents the implementation of the graph data mining process and defines the NC-Index. Section IV gives the critical component identification results and analysis. Conclusions are drawn in Section V.

II. Cyber-Physical Cascading Model Considering Time-Varying Operational States

In this section, we introduce the cyber-physical cascading failure model used in this paper. Then, we define the cascading graph of a single cyber-physical operational state. Furthermore, we propose to construct a cascading graph database to capture the cascading characteristics over different operational states, which is the solid basis for further critical component evaluation.

A. Cyber-Physical Cascading Failure Model

In power systems, cascading failures can be described as a rapid, uncontrolled sequence of power equipment disconnections from the power grid, which may result in a blackout. In general, the fundamental idea of generating cascading failure datasets in power systems is based on simulations with existing cascading models [25]-[27]. In cyber-physical power systems, the cascading process described above is further influenced by the cyber-physical interactions. The cyber-physical interplay can amplify the cascading effects. For instance, cyber attacks, e.g., false data injection and distributed denial of service, can misguide the decision-making in the control center and pose a significant threat to power system operation. Furthermore, a power system outage can disrupt communication networks affecting the power grid monitoring and control capabilities, which can further destabilize the CPS. In this paper, we adopt the cyber-physical cascade model developed in our previous work [27]. Note that to use the methodology proposed in this paper for analyzing the cyber-physical cascading mechanism, the cascade data can also be generated based on other cyber-physical system models in the literature. The cyber-physical cascading failures chain C_{CF} can be represented as in (1).

$$C_{CF} = \langle \boldsymbol{C}_1 \rangle \to \langle \boldsymbol{C}_2 \rangle \to \langle \boldsymbol{C}_3 \rangle \to \ldots \langle \boldsymbol{C}_i \rangle \to \ldots \langle \boldsymbol{C}_n \rangle \tag{1}$$

where $\boldsymbol{C}_i = \{C_{i1}, C_{i2}, ..., C_{ik}, ...C_{im}\}$ represents a set of components in CPS and the element C_{ik} can be either a cyber or physical component. The transmission lines represent the physical components, while the Supervisory Control and Data Acquisition (SCADA) system in the control center, communication network components, and the protection, automation and control systems in substations are abstracted into cyber components. $\{C_{i1}, C_{i2}, ..., C_{ik}, ...C_{im}\}$ indicates that after the removal of prefixed components $\langle \boldsymbol{C}_{i-1} \rangle$, multiple components can be disabled simultaneously. In general, a cascading failure chain as (1) contains information about components correlation and transitivity. (i) Components correlation: in (1), the relationship between $\langle \boldsymbol{C}_1 \rangle$ and $\langle \boldsymbol{C}_2 \rangle$ can be considered as the causality correlation, which indicates that the failure of the components in $\langle \boldsymbol{C}_2 \rangle$ is caused by the removal of all components in $\langle \boldsymbol{C}_1 \rangle$. (ii) Transitivity: in [8], the transitivity of a cascading failure chain is defined as: if there exist $\langle \{C_{11}\} \rangle \to \langle \{C_{21}, C_{22}\} \rangle \to \langle \{C_{31}\} \rangle$, the components C_{11} and C_{31} are correlated even if the failure of C_{31} is not directly cause by C_{11}. Note that if the correlations $\langle \{C_{11}\} \rangle \to \langle \{C_{21}, C_{22}\} \rangle$ and $\langle \{C_{21}, C_{22}\} \rangle \to \langle \{C_{31}\} \rangle$ originate from two different cascading failure chains, the transitivity property cannot be used directly. We will further discuss this issue in Section II, Part B.

In this paper, we further investigate the correlations among CPS components. Based on the cascading failure data, in the following content, we construct the cascading graph database and mine the frequent subgraph to further reveal the cascading mechanism of CPS.

B. Weighted Cascading Graph Generation for a Single Operational State

By utilizing the cascading model in [27], we generate N cascading chains at a given operational state as in (1) and construct a weighted cascading graph. The definitions and detailed generation process are as follows.

Definition 1 (Manifest Correlation): For two given CPS components $C_{ik} \in \boldsymbol{C}_i$ and $C_{jk} \in \boldsymbol{C}_j$, if $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ are in the same cascading failure chain, then we define the correlation between C_{ik} and C_{jk} as manifest correlation, and it is denoted as $C_{ik} \to C_{jk}$.

Definition 2 (Latent Correlation): For three given CPS components $C_{ik} \in \boldsymbol{C}_i$ $C_{jk} \in \boldsymbol{C}_j$ and $C_{lk} \in \boldsymbol{C}_l$, if it satisfies $C_{ik} \to C_{jk}$, $C_{jk} \to C_{lk}$, and $\boldsymbol{C}_i$ is not in the same cascading failure chain with $\boldsymbol{C}_l$, then we define the correlation between C_{ik} and C_{lk} as latent correlation, and it is denoted as $C_{ik} \Rightarrow C_{lk}$.

Example 1. Let two cyber-physical cascading failure chains both with the length of 3 be $C^{(1)}_{CF} = \langle\{C_{11}\}\rangle \to \langle\{C_{21}, C_{22}\}\rangle \to \langle\{C_{31}\}\rangle$ and $C^{(2)}_{CF} = \langle\{C_{21}, C_{22}\}\rangle \to \langle\{C_{31}\}\rangle \to \langle\{C_{41}\}\rangle$, where $C^{(1)}_{CF}$ and $C^{(2)}_{CF}$ are generated under the same system condition. In this example, C_{11} and C_{31} have the manifest correlation. C_{11} and C_{41} have the latent correlation.

Definition 3 (Transitivity of Cascading Correlation): We define the symbol $\rhd$ to indicate the cascading correlation between any two components C_{ik} and C_{jk} ,and it is denoted as:

$$R\left(C_{ik}, C_{jk}\right) = C_{ik} \rhd C_{jk} \tag{2}$$

Note that $C_{ik} \rhd C_{jk}$ indicates that C_{ik} and C_{jk} either satisfy $C_{ik} \to C_{jk}$ or $C_{ik} \Rightarrow C_{lk}$. Then, the transitivity of cascading correlation is defined as if $\exists$ $C_{1k}, C_{2k}, C_{3k}, ..., C_{ik}, ...C_{nk}$ satisfy $C_{1k} \rhd C_{2k}$, $C_{2k} \rhd C_{3k}$, …, $C_{ik} \rhd C_{(i+1)k}$, …, $C_{(n-1)k} \rhd C_{nk}$, then

$$\begin{aligned} &R\left(C_{1k}, C_{2k}, C_{3k}, ..., C_{ik}, ...C_{nk}\right) = \\ &C_{1k} \rhd C_{2k} \rhd C_{3k} ... \rhd C_{ik} ... \rhd C_{nk} \end{aligned} \tag{3}$$

Note that once (3) is satisfied, there is a transitivity property between any two components in (3).

Definition 4 (Mapping a Cascading Chain into a Graph): we define a mapping operator $F: R\left(C^{(i)}_{CF}\right) \mapsto \boldsymbol{G}^{(i)}_{CF}$, and $\boldsymbol{G}^{(i)}_{CF} = F\left(R\left(C^{(i)}_{CF}\right)\right) = \left\langle \boldsymbol{V}^{(i)}_{CF}, \boldsymbol{E}^{(i)}_{CF}, \boldsymbol{w}^{(i)}, \boldsymbol{\phi}^{(i)}_{w} \right\rangle$ is a directed graph, where $\boldsymbol{V}^{(i)}_{CF}$ is the set of vertices in $\boldsymbol{G}^{(i)}_{CF}$ and is mapped from all the components in $C^{(i)}_{CF}$, $\boldsymbol{E}^{(i)}_{CF}$ is the set of edges in $\boldsymbol{G}^{(i)}_{CF}$ and is mapped from all the cascading correlations in $C^{(i)}_{CF}$, $\boldsymbol{w}^{(i)}$ is the weight set of all edges mapped by the mapping relationship $\boldsymbol{\phi}^{(i)}_{w}$.

Based on definitions 1-4, one can map a cascading failure chain $C^{(i)}_{CF}$ into a directed and weighted graph. Note that in definition 4, the weights of all edges are set to 1 by default because for one cascading failure chain, each component can only be removed once, and the weights of edges represent the frequency of the corresponding correlation in the cascading data. To thoroughly evaluate the importance of each component in the system, one can construct N cyber-physical cascading failure chains, i.e., $C^{(1)}_{CF}$, $C^{(2)}_{CF}$, …, $C^{(N)}_{CF}$. Then, based on definitions 1-4, we can construct N directed graphs, i.e., $\boldsymbol{G}^{(1)}_{CF}$, $\boldsymbol{G}^{(2)}_{CF}$, …, $\boldsymbol{G}^{(N)}_{CF}$. Furthermore, these graphs can be combined to generate a weighted cascading graph $\boldsymbol{G}_{CF}\left(t_x\right)$ for a single operational state t_x as follows:

$$\boldsymbol{G}_{CF}\left(t_x\right) = \left\langle \boldsymbol{V}^{(t_x)}_{CF}, \boldsymbol{E}^{(t_x)}_{CF}, \boldsymbol{w}^{(t_x)}, \boldsymbol{\phi}^{(t_x)}_{w} \right\rangle \tag{4}$$

$$\boldsymbol{V}^{(t_x)}_{CF} = \bigcup_{i=1}^{N} \boldsymbol{V}^{(i)}_{CF} \tag{5}$$

$$\boldsymbol{E}^{(t_x)}_{CF} = \bigcup_{i=1}^{N} \boldsymbol{E}^{(i)}_{CF} \tag{6}$$

$$\boldsymbol{w}^{(t_x)} = \left\{ w_{E^{(t_x)}_{CF}} \middle| w_{E^{(t_x)}_{CF}} = f(E^{(t_x)}_{CF}) \right\} \tag{7}$$

Where $f(E^{(t_x)}_{CF})$ is the frequency of edge $E^{(t_x)}_{CF}$ among $\boldsymbol{G}^{(1)}_{CF}$, $\boldsymbol{G}^{(2)}_{CF}$, …, $\boldsymbol{G}^{(N)}_{CF}$. By following definitions 1-4 and equations (2)-(7), the cascading correlations are captured and emerged into the weighted cascading graph. The transitivity of cascading correlations is also converted into the connectivity of components. If there exists a path between two vertices in the weighted cascading graph, it indicates that there is a manifest or latent correlation between the two components. In Algorithm 1, we present the detailed generation process of the weighted cascading graph.

```
Algorithm 1: Generation of weighted cascading graph
Input:
    $C^{(1)}_{CF}$, $C^{(2)}_{CF}$, …, $C^{(N)}_{CF}$ at $t_x$
Output:
    Optimal candidate edge set: $\boldsymbol{G}_{CF}\left(t_x\right)$
Step 1   $\boldsymbol{V}^{(t_x)}_{CF} \leftarrow \varnothing$, $\boldsymbol{E}^{(t_x)}_{CF} \leftarrow \varnothing$
Step 2   For each $C^{(i)}_{CF}$ do
Step 3       Covert $C^{(i)}_{CF}$ into $\boldsymbol{G}^{(i)}_{CF}$ based on definition 1-4
Step 4   End For
Step 5   For each $\boldsymbol{G}^{(i)}_{CF}$ do
Step 6       $\boldsymbol{V}^{(t_x)}_{CF} \leftarrow \boldsymbol{V}^{(i)}_{CF} \cup \boldsymbol{V}^{(t_x)}_{CF}$
Step 7       $\boldsymbol{E}^{(t_x)}_{CF} \leftarrow \boldsymbol{E}^{(i)}_{CF} \cup \boldsymbol{E}^{(t_x)}_{CF}$
Step 8   End For
Step 9   Employ equation (7) to calculate $\boldsymbol{w}^{(t_x)}$
Step 10  Return $\boldsymbol{G}_{CF}\left(t_x\right)$
```

C. Constructing Weighted Cascading Graph Database Considering Time-Varying Operational States

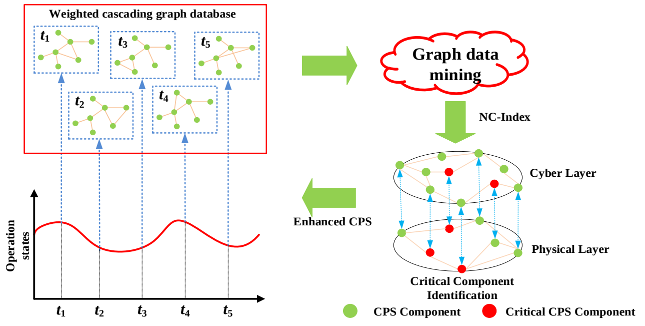


Fig. 1 The framework of GraphCCI

The cascading characteristics captured in $\boldsymbol{G}_{CF}\left(t_x\right)$ contain only the system information under one specific operational state, which fail to capture the overall cascading features of CPS under different operational states [7][27]. For example, the critical components identified under a specific operational state may not apply to other operational states. Therefore, to capture the overall cascading characteristics, we define a weighted cascading graph database that contains the cascading characteristics under different time-varying operational states. As in Fig. 1, for a certain time interval $\left[t_0, t_u\right]$, the weighted cascading graph database $\boldsymbol{G}_D$ can be represented as:

$$\boldsymbol{G}_D = \left\{ \boldsymbol{G}_{CF}\left(t_x\right) \middle| t_0 \le t_x \le t_u \right\} \tag{8}$$

In this paper, we propose a model for critical components identification, i.e., GraphCCI. As represented in Fig. 1, we first collect the cascading failure data under different operational state. Then, by adopting the methods proposed in Section II, we map the cascading information into a weighted cascading graph database. Note that to increase the accuracy of the critical component evaluation results, one should simulate different operational states as much as possible so that $\boldsymbol{G}_D$ can comprehensively cover the cascading failures information. The next step is to utilize graph data mining algorithms to identify the critical subgraphs. In this paper, we focus on the frequency aspect of subgraphs and adopt the TKG algorithm [24] to identify the top-K frequent subgraphs from $\boldsymbol{G}_D$. Then, by using the proposed NC-Index, we identify the critical CPS components and enhance the security level of CPS.

III. Graph Mining-Based Critical Component Evaluation

A. Mining Frequent Subgraphs Based on the Weighted Cascading Graph Database

To better reveal the cascading characteristics of CPS, we adopt graph data mining algorithms to mine the frequent subgraphs in the weighted cascading graph database constructed in Section II. The definitions of graph data mining are given as follows:

Definition 5 (Cascading Subgraphs) For a given cascading graph $\boldsymbol{G}_{CF}\left(t_x\right) = \left\langle \boldsymbol{V}^{(t_x)}{}_{CF}, \boldsymbol{E}^{(t_x)}{}_{CF}, \boldsymbol{w}^{(t_x)}, \boldsymbol{\phi}_w^{(t_x)} \right\rangle$, if there exists a graph $\boldsymbol{g}^{(i)}{}_{CF} = \left\langle \boldsymbol{v}^{(i)}{}_{CF}, \boldsymbol{e}^{(i)}{}_{CF}, \boldsymbol{w}_g{}^{(i)}, \boldsymbol{\phi}_g{}^{(i)}{}_w \right\rangle$ that satisfies $\boldsymbol{v}^{(i)}{}_{CF} \subseteq \boldsymbol{V}^{(t_x)}{}_{CF}$, $\boldsymbol{e}^{(i)}{}_{CF} \subseteq \boldsymbol{E}^{(t_x)}{}_{CF}$, $\boldsymbol{w}_g{}^{(i)} \subseteq \boldsymbol{w}^{(t_x)}$, $\boldsymbol{\phi}_g{}^{(i)}{}_w \subseteq \boldsymbol{\phi}^{(t_x)}{}_w$, then $\boldsymbol{g}^{(i)}{}_{CF}$ is a subgraph of $\boldsymbol{G}_{CF}\left(t_x\right)$, which is denoted as $\boldsymbol{g}^{(i)}{}_{CF} \subseteq \boldsymbol{G}_{CF}\left(t_x\right)$.

Definition 6 (Frequent Cascading Subgraphs) For a given weighted cascading graph database $\boldsymbol{G}_D$ and a subgraph $\boldsymbol{g}^{(i)}{}_{CF} \subseteq \boldsymbol{G}_{CF}\left(t_x\right)$, the support (occurrence frequency) of $\boldsymbol{g}^{(i)}{}_{CF}$ is calculated by (9)

$$sup(\boldsymbol{g}^{(i)}{}_{CF}) = \left| \left\{ \boldsymbol{G}_{CF}\left(t_x\right) \middle| \boldsymbol{G}_{CF}\left(t_x\right) \in \boldsymbol{G}_D \cap \boldsymbol{g}^{(i)}{}_{CF} \subseteq \boldsymbol{G}_{CF}\left(t_x\right) \right\} \right| \tag{9}$$

If $sup(\boldsymbol{g}^{(i)}{}_{CF})$ is greater than a user-defined minimum threshold $minsup$, then $\boldsymbol{g}^{(i)}{}_{CF}$ is considered a frequent cascading subgraphs, and is denoted as $\boldsymbol{g}^{(i)}{}_f$.

In general, graph data mining algorithms require a user-defined $minsup$ to determine whether a subgraph is frequent or not. However, how to set an appropriate $minsup$ is challenging. If the $minsup$ is too high, few or even no subgraphs can be discovered. If the $minsup$ is too low, plenty of useless subgraphs will be included in the results and thus decrease the accuracy of identifying critical components for CPS. Therefore, to address the mentioned issue, we adopted a Top-K structure [28]. For a user-defined $K \ge 1$ and a graph database $\boldsymbol{G}_D$, the Top-K graph mining problem is to find a set $\boldsymbol{F}_g = \left\{ \boldsymbol{g}^{(i)}{}_f \middle| \boldsymbol{g}^{(i)}{}_f = \left\langle \boldsymbol{v}^{(i)}{}_f, \boldsymbol{e}^{(i)}{}_f, \boldsymbol{w}_f{}^{(i)}, \boldsymbol{\phi}_f{}^{(i)}{}_w \right\rangle \right\}$ consists of K subgraphs that their support is greater or equal to that of any other subgraphs not in $\boldsymbol{F}_g$. There is a fundamental distinction between the *minsup* and Top-K approaches. Compared with the Top-K method, the *minsup* approach does not prioritize the results according to the frequency of subgraphs. As a result, modifying the *minsup* parameter might result in the omission of important information. However, in Top-K method, adjusting the K value ensures the consistent retrieval of the top K most frequent subgraphs, irrespective of the adjustments. That is, the most critical components are always prioritized. Note that K is a parameter defined by the user, which should be set with consideration to the defensive capabilities of the CPS operator. This means that the CPS operator must select K by considering the number of critical components that can be simultaneously defended or enhanced. In Section IV, a thorough analysis of how to determine an appropriate value for K are presented.

In this paper, we adopt the TKG algorithm [24] to mine the Top-K frequent cascading subgraphs from the constructed database $\boldsymbol{G}_D$. The critical questions that need to be answered during the graph data mining process are how to effectively traverse all the possible subgraphs and how to efficiently calculate the support of each subgraph. To do so, we utilize the rightmost path extension strategy [25] to traverse the target graphs without missing any nodes and edges. Then, the canonical Depth-First Search (DFS) code [24] is used to represent the graphs in a unified format so that it can significantly facilitate the mining process. The reason we employ DFS rather than Breadth-First Search (BFS) is that BFS is less efficient than DFS when traversing the graph data and generating subgraph candidates [24]. In [29], the authors thoroughly compared the DFS and BFS strategies, focusing on two classic algorithms: FSG (which uses a BFS strategy) [30] and *gSpan* (which uses a DFS strategy) [29]. The test dataset comprises 340 different graphs, each containing an average of 27 nodes and 28 edges, with the largest graph containing 214 nodes and 214 edges. The experimental results indicate that gSpan using DFS consumes significantly less computational memory and achieves a better performance, i.e., 15 to 100 times, than FSG using BFS. Therefore, we choose DFS over BFS in our method. Also, this is the reason why we choose the rightmost path extension strategy because it can avoid using BFS and it allows to explore the search space while avoiding generating extra candidates.

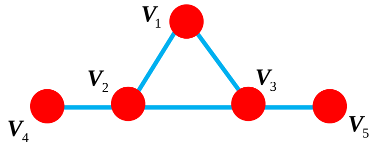


Fig. 2 The rightmost path extension strategy

Rightmost path extension strategy: This strategy follows the principle of depth-first search, and it is implemented over a graph using a recursive stack. In this stack, nodes are used as the basis for an extension, and the currently processed node is called the rightmost node. In general, there are two types of extensions: forward extensions and backward extensions, where forward extensions are used to form an edge to visit new nodes and backward extensions are the opposite. Note that this strategy always implements backward extensions before forward extensions to avoid missing edges. Fig. 2 gives an example of how the rightmost path extension traverses a graph. Assuming that we start from node V_1, one can randomly choose from its neighbors V_2 and V_3 for the next extension. Taking V_2 as an example, V_4 is next to be visited. Then, because V_4 does not have other neighbors, the strategy will go back to V_2 and visit V_3. At this moment, V_3 has two available neighbors, i.e., V_1 and V_5. Given that the extension between V_3 and V_1 is the backward extension, the rightmost path strategy will visit V_1 first and then V_5. The eventual visiting order of the edges is E_{12}, E_{24}, E_{23}, E_{31} and E_{35}.

Canonical DFS: the depth-first search of a graph is defined as a sequence of the extended edges, sorted in the depth-first search order. Continuing the previous example of Fig. 2, the sequence of E_{12}, E_{24}, E_{23}, E_{31} and E_{35} is the DFS of the graph. To make sure that each graph and subgraphs in the database can be represented by only a specific DFS during the mining process, the *total order of extended edges* is used to unify the expression of each graph. For the definition of *total order of extended edges*, readers are referred to [24] for details. For a graph, the DFS with the smallest *total order of extended edges* is the canonical DFS. Algorithm 2 presents how TKG mines the Top-K frequent cascading subgraphs from the constructed database $\boldsymbol{G}_D$, where *RightMostPathExtension*(*) and *isCanonical*(*) represent the strategy and method implementation for the corresponding targets as discussed.

Note that all the components included in Q_K are considered as critical CPS components, and they are denoted as $\boldsymbol{C}_c = \boldsymbol{v}^{(1)}{}_f \cup \boldsymbol{v}^{(2)}{}_f ... \cup \boldsymbol{v}^{(i)}{}_f ... \cup \boldsymbol{v}^{(K)}{}_f$. To further prioritize these critical components, we conduct the critical component evaluation as in Part B.

Algorithm 2: Generation of weighted cascading graph

```
Input:
$\boldsymbol{G}_D$
$K$
$Q_K$ : For storing the current top-k frequent subgraphs
$Q_C$ : For storing candidate subgraphs for next extension.
Output:
The set of frequent subgraphs: $\boldsymbol{G}_f = \{\boldsymbol{g}^{(i)}{}_f | i = 1,2,...\}$

Step 1  $minsup = 1$
Step 2  While $Q_C$ is not empty do
Step 3      $\boldsymbol{g} \leftarrow$ the subgraph with the highest support in $Q_C$
Step 4      $\varepsilon \leftarrow RightMostPathExtension(\boldsymbol{g})$
Step 5      For each extension $\in \varepsilon$ do
Step 6          $\boldsymbol{g}' \leftarrow \boldsymbol{g} \cup extension$
Step 7          If $sup(\boldsymbol{g}') \geq minsup$ and $isCanonical(\boldsymbol{g}')$
Step 8              $Q_K \leftarrow \boldsymbol{g}'$
Step 9              If $|Q_K| > K$
Step 10                 $minsup = min(\boldsymbol{sup}(\boldsymbol{g}^{(i)}{}_{CF}))$
Step 11             End
Step 12             $Q_C \leftarrow \boldsymbol{g}'$
Step 13         End
Step 14     End
Step 15 End
Step 16 Return $Q_K$
```

B. Critical Component Evaluation for Cyber-Physical Power Systems

In this part, we quantify the correlations between the identified critical components to further evaluate the criticality of each component from the perspectives of manifest and latent correlations as defined in Section II. For the convenience of calculation, we merge all the identified frequent subgraphs in $\boldsymbol{F}_g$ into a Cascading Characteristics Graph (CC-Graph).

Definition 7 (CC-Graph) For a given frequent subgraph set $\boldsymbol{F}_g$, the corresponding CC-Graph is defined as in (9)-(12)

$$\boldsymbol{G}_{CC} = \langle \boldsymbol{V}_{CC}, \boldsymbol{E}_{CC}, \boldsymbol{w}_{CC}, \boldsymbol{\phi}_{CC} \rangle \tag{9}$$

$$\boldsymbol{V}_{CC} = \bigcup_{i=1}^{I} \boldsymbol{v}^{(i)}{}_f \tag{10}$$

$$\boldsymbol{E}_{CC} = \bigcup_{i=1}^{I} \boldsymbol{e}^{(i)}{}_f \tag{11}$$

$$\boldsymbol{w}^{(t_x)} = \left\{ w_{\boldsymbol{E}_{CC}} \middle| w_{\boldsymbol{E}_{CC}} = f(\boldsymbol{E}_{CC}) \right\} \tag{12}$$

The definition of CC-Graph is similar to the definition of $\boldsymbol{G}_{CF}(t_x)$. Note that $\boldsymbol{G}_{CC}$ is not necessarily a connected graph. In $\boldsymbol{G}_{CC}$, all the edges represent the manifest correlations among the identified critical components. To quantify the manifest correlations, we define the manifest correlation coefficient as in Definition 8.

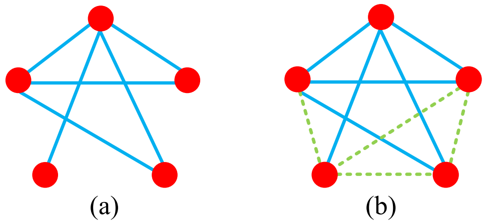


Fig. 3 (a) Example of CC-Graph. (b) Latent correlation calculation

Definition 8 (Manifest Correlation Coefficient) For an edge $e_p = (v_q, v_r) \in \boldsymbol{E}_{CC}$, the manifest correlation coefficient is defined as in (13)-(15):

$$\boldsymbol{C}_{CF}(t_x) = \left\{ R\left(C^{(1)}{}_{CF}(t_x), C^{(2)}{}_{CF}(t_x), ..., C^{(N)}{}_{CF}(t_x) \right) \right\} \tag{13}$$

$$\boldsymbol{C}_D = \left\{ \boldsymbol{C}_{CF}(t_x) \middle| t_0 \leq t_x \leq t_u \right\} \tag{14}$$

$$M_{Ce_p}(\boldsymbol{C}_D) = \frac{\left| \left\{ \boldsymbol{C}_{CF}(t_x) \middle| \exists v_q \rightarrow v_r \right\} \right|}{\left| \left\{ \boldsymbol{C}_{CF}(t_x) \middle| v_q \in \boldsymbol{C}_{CF}(t_x) \right\} \right|} \tag{15}$$

By calculating the manifest correlation for all edges in $\boldsymbol{G}_{CC}$, the CC-Graph is updated to $\boldsymbol{G}_{CC} = \langle \boldsymbol{V}_{CC}, \boldsymbol{E}_{CC}, \boldsymbol{w}_{CC} \boldsymbol{M}_{\boldsymbol{C}}{}^T, \boldsymbol{\phi}_{CC} \rangle$, where $\boldsymbol{M_C}^T$ are the sets of M_{Ce_p} for all edges and they share the same mapping relationship $\boldsymbol{\phi}_{CC}$ as for $\boldsymbol{w}_{CC}$. Then, to thoroughly investigate the cascading characteristics of CPS, we evaluate the latent correlations among the identified critical components. Fig. 3(a) is an example of a CC-graph, where the blue edges represent the manifest correlations. For the nodes that are not directly connected, they may or may not have latent correlations, as demonstrated in the green edges in Fig. 3(b). To examine the latent correlation features, we extend $\boldsymbol{G}_{CC}$ to a full connection graph $\boldsymbol{G}'_{CC} = \left\langle \boldsymbol{V}_{CC}, \boldsymbol{E}'_{CC}, \boldsymbol{w}_{CC}\boldsymbol{M_C}^T \oplus \boldsymbol{L_C}^T, \boldsymbol{\phi}'_{CC} \right\rangle$, where $\boldsymbol{L_C}^T = \{L_{Ce_q} | e_q \in \boldsymbol{E}'_{CC}\}$, and the latent correlation coefficient of the extended edges are calculated by (16).

Definition 9 (Latent Correlation Coefficient) For an edge $e_q = (v_q, v_r) \in \boldsymbol{E}'_{CC}$, the latent correlation coefficient L_{Ce_q} is defined as in (16):

$$L_{Ce_q} = \frac{\left|\left\{\boldsymbol{G}_{CF}(t_x) \middle| \exists v_q \Rightarrow v_r\right\}\right|}{|\boldsymbol{G}_D|} \qquad (16)$$

Based on the extended CC-Graph, we propose the node criticality index to quantify the importance of each identified critical component. The definition of NC-index is given as follows.

Definition 10 (NC-Index) For a critical component $v_q \in \boldsymbol{V}_{CC}$, the NC-index of v_q is denoted as $\boldsymbol{N}_{Cq}$, and is calculated by (17).

$$\boldsymbol{N}_{Cq} = \sum_{\boldsymbol{E}_y} M_{Ce_y} + \sum_{\boldsymbol{E}_y} L_{Ce_y} \qquad (17)$$

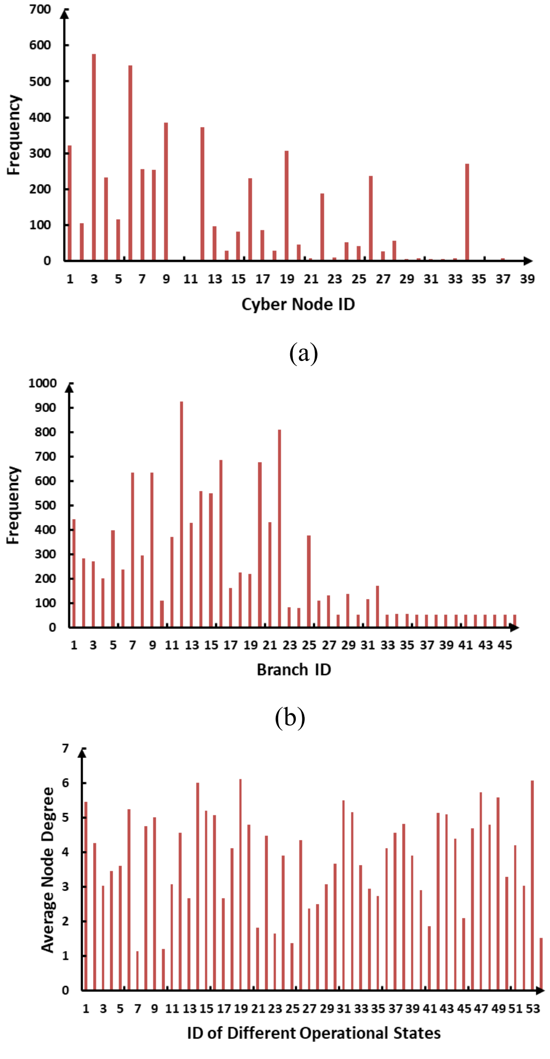
Where $\boldsymbol{E}_y = \{e_1, e_2, ... e_y, ..., e_Y\}$ consists of all the edges that are connected to v_q including the extended edges. For each critical component, $\boldsymbol{N}_{Cq}$ evaluates its criticality considering both its manifest and latent correlations. The higher the $\boldsymbol{N}_{Cq}$ value, the more important the component is for enhancing CPS security.

IV. Case Studies

In this section, we implement the proposed methodology to the IEEE 39-bus test system. The modeling details and cyber-physical cascading model can be found in our previous work [27], which contains 78 nodes in total. In this paper, we simulate the cascading model for 54 weeks and collect 2,483 cascading chains. For each week, we construct a weighted cascading graph to form the graph database. The simulations are conducted in Python running on a laptop, which is equipped with an Intel i7-8750H CPU @ 2.2 GHz and 16 GB RAM.

A. Feature Analysis of the Weighted Graph Database

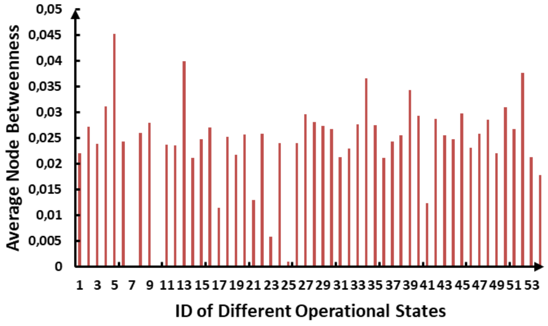
From the graph database perspective, Fig. 4(a) and (b) present the frequencies of cyber-physical components in the database. The frequencies reflect the extent to which the components contribute to the cascading process. At the cyber system layer, the 5 most critical cyber components are nodes 3, 6, 12, 1, and 19, while at the physical system layer the 5 most critical components are branches 12, 22, 16, 20, and 7. To analyse the constructed weighted graph database, we adopt the average node degree and average node betweenness to describe the graph features of the graph database. The average node degree defines the average amount of nodes connected to a selected node, and it reflects the connectivity of the graph. A high average node degree means that the information or resource can be exchanged in a more efficient manner. On the other hand, the average node betweenness in a graph reflects the extent to which nodes act as bridges in the transmission of information or resources. This metric measures the importance of each node as an intermediary on the shortest paths connecting other pairs of nodes within the network, on average. The results of the graph feature are given in Fig. 4(c) and (d). By analyse the results, one can observe that the weighted cascading graphs under different operational states exhibit distinctly different characteristics. In Fig. 4(c), the average node degree scales from 1.143 (operational state 7) to 6.119 (operational state 19), while in Fig. 4 (d), the highest value (0.045309) is 278 times bigger than the smallest value (0.000255). Such significant variation further proves our argument in the Introduction that the experimental results under one single operational state may not be applicable under different system statuses.



(a)

(b)

(c)



(d)

Fig. 4 (a) Frequency of cyber nodes in graph database (b) Frequency of physical branches in graph database. (c) Average node degree of weighted cascading graphs (d) Average node betweenness of weighted cascading graphs

B. The Identification of Critical Components and the CC-Graph

In this part, we present the construction results of CC-Graph using the methodology proposed in Section III. During the implementation process of the TKG algorithm, we investigate the impact of different K values on the number of identified critical components. In Fig. 5, as the K value increases, the number of critical components increases along with it. However, the increasing rate has a visible decrease at K=40. On the other hand, in Fig. 6, we present the relationship between K value and the structural entropy [31] of CC-Graph. The structural entropy $E_{entropy}(\boldsymbol{G}_{CC})$ is used to quantify the information amount contained in each CC-Graph that is constructed based on a given K value, and it can be calculated by following equation (18)

$$E_{entropy}(\boldsymbol{G}_{CC}) = -\sum_{i=1}^{I}\left(P_d(\boldsymbol{v}^{(i)}{}_f) \times \log_2 P_d(\boldsymbol{v}^{(i)}{}_f)\right) \tag{18}$$

where $P_d(\boldsymbol{v}^{(i)}{}_f)$ is the probability distribution of the degree of node $\boldsymbol{v}^{(i)}{}_f$. When the structural entropy of a graph is higher, it indicates that the graph is more complex and contains more information. In our case, it is desirable to analyze the CC-Graph with the highest structural entropy, because it means that the corresponding CC-Graph contains the most thorough information of components correlation. In Fig. 6, one can observe that the $E_{entropy}(\boldsymbol{G}_{CC})$ quickly increases when K is small and is eventually stabilized. This process indicates that as the K increases, the CC-Graph contains more information of component correlation. Also, when the K increases beyond a certain point, the increase of K will not add new information to the CC-Graph and only causes small changes to the $E_{entropy}(\boldsymbol{G}_{CC})$. Therefore, when K value is too low, some critical components correlation information may be missed in the CC-Graph. On the other hand, when K value is too high, it does not add new and useful information to the CC-Graph while it also increases the cost of defending critical components. Based on the discussion above, the optimal K value is determined when the corresponding $E_{entropy}(\boldsymbol{G}_{CC})$ reaches the maximum. In Fig. 6, the optimal K is 40.

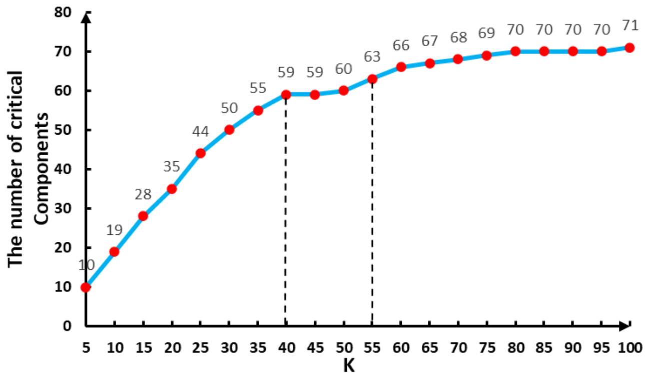


Fig. 5 The number of identified critical components under different K

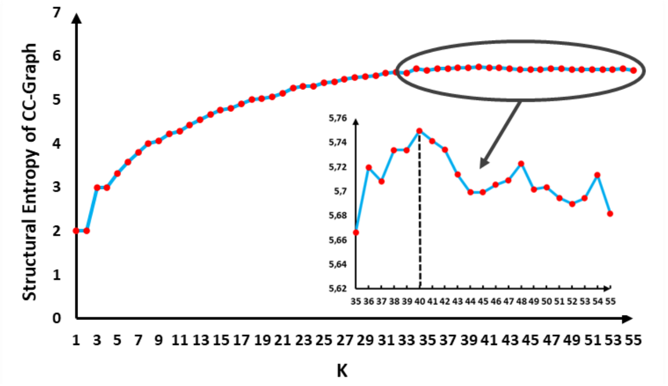


Fig. 6 The relationship between K value and the structural entropy of CC-Graph

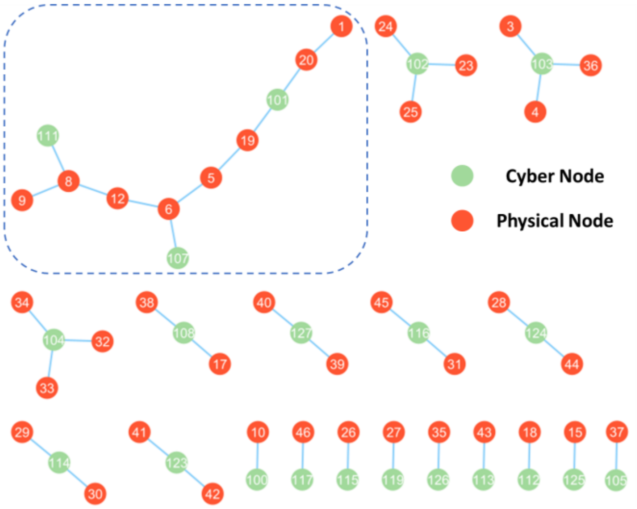
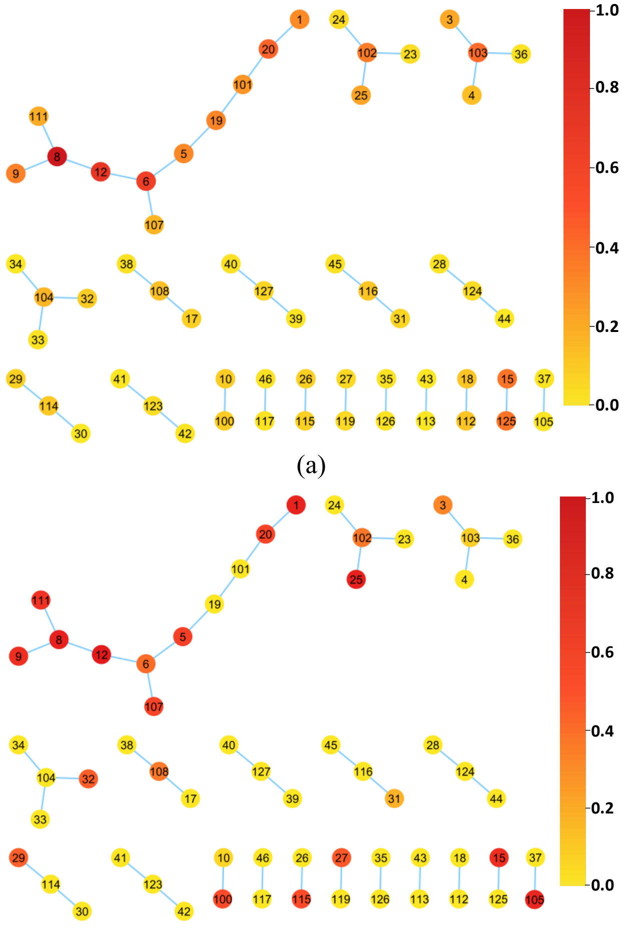


Fig. 7 Constructed CC-Graph when K=40

Fig. 7 presents the generated CC-Graph when K=40. In this graph, there are in total 21 critical cyber nodes and 38 critical physical branches. For each pair of nodes that are directly connected, apparent manifest correlations exist. For each pair of nodes that are indirectly connected but have an accessible path, latent correlations exist. Note that the latent correlations in Fig. 7 only consider the mined frequent cascading subgraphs. They frequently appear in the graph database, and it does not prove that there are no latent correlations between those node pairs having no accessible path. For example, node 1 and node 24 on the top of the CC-Graph are not directly or indirectly connected, but there is still a possible latent correlation between them. From a global perspective, the CC-Graph in Fig 6 is not a connected graph, and the node degree of each node is not high (the maximum value is 3). It indicates that the range of the frequent cascading patterns is not extensive. However, by observing the marked area, this is a comparatively large connected graph, which indicates that if any node in this area fails, it may cause a severe impact on the system operation. In the next part, we will further quantify the criticality of each node in Fig. 7 by using the proposed NC-Index.

C. The Critical Components Evaluation Results



(a)

(b)

Fig. 8 (a) CC-Graph with only manifest correlation coefficients (b) CC-Graph with NC-Index

In Fig. 8 and Table I, we present the calculation results of all the indices we proposed in Section III. In Fig. 8(a), we only consider the manifest correlation. The ranking results of the manifest correlation coefficients are decided by two factors, i.e., the evident support in the historical data and the node degree of the nodes in the CC-Graph. Fig. 8(a) also proves this point and the nodes with a higher degree are comparatively more critical than the other low-degree nodes. Also, the results indicate that the most high-ranked components are in the largest subgraph. This indicates that these nodes have tighter connections with the other nodes and a wider range to propagate the failures. The detailed ranking information is given in Table I. In Fig. 8(b), we jointly consider the manifest correlations and latent correlations. Compared with Fig. 8(a), the most critical components are still mainly distributed in the largest subgraph. However, part of the critical components from the largest subgraph rank lower, while some components from smaller subgraphs rank higher. This is because the latent correlation considers the global relationships among components, and it quantifies the risk of indirectly triggering a cascading failure.

TABLE I
Ranking of critical components considering different indices

Ranking	Considering only manifest correlation coefficient	Considering NC-Index
1	8 (physical)	12 (physical)
2	12 (physical)	8 (physical)
3	6 (physical)	25 (physical)
4	20 (physical)	1 (physical)
5	103 (cyber)	105 (cyber)
6	15 (physical)	9 (physical)
7	125 (cyber)	15 (physical)
8	102 (cyber)	111 (cyber)
9	9 (physical)	5 (physical)
10	19 (physical)	20 (physical)

D. Performance Comparison with the Literature

In this part, we compare the proposed method with the existing literature to prove its effectiveness. We compare the performance of methods from two aspects: load loss and network efficiency. For the load loss, we implement each method to identify the top-5 critical components for the CPS of IEEE 39-bus system as explained in [27]. Then, we traverse the possible combination of those components and disconnect them to observe the load loss after the cascading failures. For each method, we record the highest load loss. Similarly, we use the same approach to calculate the network efficiency of each remaining network after the cascading failures. It is worth mentioning that unlike load loss, the network efficiency only indicates the topological features of the network, and it quantifies the network connectivity.

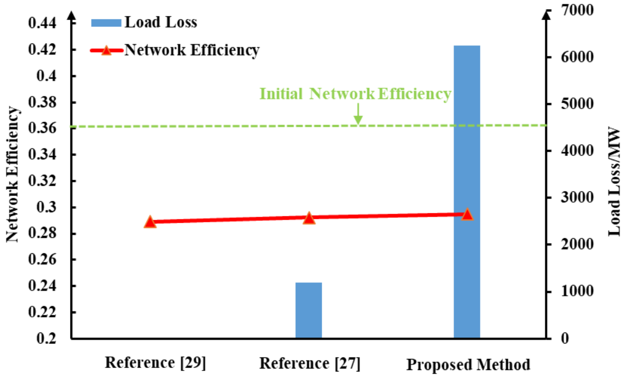


Fig. 9 The comparison results.

We compare the proposed method with reference [27] and [32], where [27] considered the cascading failure data and identified the critical components based on the proposed index while [32] evaluated the nodes importance for power system from the perspective of centrality measures. In Fig. 9, we present the comparison results. From the perspective of load loss, one can observe that the removal of the critical components indentified by the proposed method can cause a much higher load loss, while there is no load loss in the results of [32]. The reason behind the results is that reference [32] neglects the node heterogeneity of CPS and only consider the topological aspects of networks. In real industrial scenarios, we place greater emphasis on factors that can directly lead to security issues and financial losses, such as load loss. Besides, by analyzing the network efficiency results, one can observe that there is a clear decrease in all three methods compared with the initial network. However, by combining the results of load loss and network efficiency, the critical components identified by our method can cause more catastrophic cascading failures by inflicting a comparable degree of damage on the network. The comparison results effectively confirm the precision of our method in identifying critical components compared with the existing literature.

V. Conclusion

This paper proposes a graph data mining-based critical components identification model named GraphCCI, which evaluates the criticality of CPS components from the perspectives of manifest and latent correlations. First, we abstract the cascading failure data under different operational states into a weighted cascading graph database. Then, the TKG algorithm is adopted to identify the frequent subgraphs in the constructed graph database. Meanwhile, the definition of CC-Graph is proposed to model the overall cascading features based on the graph mining results. Finally, the NC-Index is proposed to evaluate the criticality of each CPS component. Our case study reveals that the cyber-physical system shows different cascading features under different system conditions. Verifications on the IEEE 39-bus test system demonstrate the effectiveness of our method. The identification results can provide an important reference to enhance CPS security and prevent cascading failures and even a blackout.

Acknowledgements

This work was partially supported by EU Horizon Europe eFORT Project with Grant Agreement No 101075665, Dutch Research Council's RESCUE Project with Grant Agreement NWO ESI.2019.006, and China Scholarship Council.

References

[1] R. M. Lee, M. J. Assante and T. Conway, "Analysis of the Cyber Attack on the Ukrainian Power Grid" USA, Mar. 18, 2016 [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2016/05/20081514/E-ISAC_SANS_Ukraine_DUC_5.pdf

[2] M. J. Assante, R. M. Lee and T. Conway, "Modular ICS Malware" USA, Aug. 2, 2017 [Online]. Available: https://ics.sans.org/ics-library.

[3] K. Proska, J. Wolfram, J. Wilson, D. Black, K. Lunden, D. Kapellmann Zafra, N. Brubaker, T. Mclellan, C. Sistrunk, "Sandworm Disrupts Power in Ukraine Using a Novel Attack Against Operational Technology" Nov. 09, 2023 [Online]. Available: https://www.mandiant.com/resources/blog/sandworm-disrupts-power-ukraine-operational-technology

[4] V. S. Rajkumar, A. Ştefanov, A. Presekal, P. Palensky and J. L. R. Torres, "Cyber Attacks on Power Grids: Causes and Propagation of Cascading Failures," *IEEE Access*, vol. 11, pp. 103154-103176, September 2023.

[5] J. Xu, R. Yao and F. Qiu, "Mitigating Cascading Outages in Severe Weather Using Simulation-Based Optimization," *IEEE Trans. Power Syst.* vol. 36, no. 1, pp. 204-213, January 2021.

[6] Y. K. Wu, Y. C. Chen, H. L. Chang and J. S. Hong, "The Effect of Decision Analysis on Power System Resilience and Economic Value During a Severe Weather Event," *IEEE Trans. Ind. Appl.*, vol. 58, no. 2, pp. 1685-1695, March-April 2022

[7] Y. Liu, S. Gao, J. Shi, X. Wei and Z. Han, "Sequential-Mining-Based Vulnerable Branches Identification for the Transmission Network Under Continuous Load Redistribution Attacks," *IEEE Trans on Smart Grid*, vol. 11, no. 6, pp. 5151-5160, November 2020.

[8] Y. Liu, S. Gao, J. Shi, X. Wei, Z. Han and T. Huang, "Pre-Overload-Graph-Based Vulnerable Correlation Identification Under Load Redistribution Attacks," *IEEE Trans on Smart Grid*, vol. 11, no. 6, pp. 5216-5226, November 2020.

[9] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic Cascade of Failures in Interdependent Networks," *Nature*, vol. 464, pp. 1025–1028, April 2010.

[10] Amin Abedi, Ludovic Gaudard, Franco Romerio, "Review of Major Approaches to Analyze Vulnerability in Power System", *Reliab. Eng. Syst. Saf.*, vol. 183, pp. 153-172, 2019

[11] Y. Wang, C. Liu, M. Shahidehpour, and C. Guo, "Critical Components for Maintenance Outage Scheduling Considering Weather Conditions and Common Mode Outages in Reconfigurable Distribution Systems," *IEEE Trans. Smart Grid*, vol. 7, no. 6, pp. 2807–2816, November 2016.

[12] J. V. Milanovic and W. Zhu, "Modeling of Interconnected Critical Infrastructure Systems Using Complex Network Theory," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4637–4648, September 2018.

[13] S. Fattaheian-Dehkordi, M. Fotuhi-Firuzabad, and R. Ghorani, "Transmission System Critical Component Identification Considering Full Substations Configuration and Protection Systems," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 5365–5373, September 2018.

[14] Y. Zhao, S. Liu, Z. Lin, L. Yang, Q. Gao, and Y. Chen, "Identification of Critical Lines for Enhancing Disaster Resilience of Power Systems with Renewables based on Complex Network Theory," *IET Gener. Transm. Distrib.*, vol. 14, no. 20, pp. 4459–4467, 2020.

[15] L. Li, H. Wu, Y. Song, and Y. Liu, "A State-Failure-Network Method to Identify Critical Components in Power Systems," *Electr. Power Syst. Res.*, vol. 181, pp. 1–10, January 2020.

[16] Q. Gao, Y. Wang, X. Cheng, J. Yu, X. Chen, and T. Jing, "Identification of Vulnerable Lines in Smart Grid Systems based on Affinity Propagation Clustering," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5163–5171, June 2019

[17] Y. Jia, Z. Xu and L. Lai, et al, "Risk-based Power System Security Analysis considering Cascading Outages," *IEEE Trans. Ind. Informat.*, vol.12, no.2, pp.872-882, March 2016.

[18] Y. Liu, N. Zhang and D. Wu, et al, "Searching for Critical Power System Cascading Failures with Graph Convolutional Network," *IEEE Trans. Contr. Network Syst.*, vol.8, no.3, pp.1304-1313, September 2021.

[19] X. Wu, D. Wu and E. Modiano, "Predicting Failure Cascades in Large Scale Power Systems via the Influence Model Framework," *IEEE Trans. Power Syst.*, vol. 36, no. 5, pp. 4778–4790, September 2021.

[20] Z. Zhang, S. Huang and S. Mei, et al., "Key Branches Identification for Cascading Failure based on Q-learning Algorithm," *in Proc. IEEE Int. Conf. Power Syst. Technol.*, Wollongong, NSW, Australia, 2016, pp. 1–6.

[21] Z. Zhang, R. Yao and S. Huang, "An Online Search Method for Representative Risky Fault Chains based on Reinforcement Learning and Knowledge Transfer," *IEEE Trans. Power Syst.*, vol. 35, no. 3, pp. 1856–1867, May 2020.

[22] A. B. Birchfield, T. Xu, K. M. Gegner, K. S. Shetye and T. J. Overbye, "Grid Structural Characteristics as Validation Criteria for Synthetic Networks," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3258-3265, 2017

[23] X. P. Ji, B. Wang, D. Liu, T. Zhao, "Review on Interdependent Networks Theory and Its Applications in the Structural Vulnerability Analysis of Electrical Cyber-physical System" *Proc. CSEE*, vol. 36, no. 17, pp. 4521–4532, September 2016

[24] Fournier-Viger, P., Cheng, C., Lin, J. C.-W., Yun, U., Iran, U. "TKG: Efficient Mining of Top-K Frequent Subgraphs," *in Proc. of 7th Intern. Conf. on Big Data Analytics (BDA 2019)*, Springer, Cham, 2019, pp. 209-226.

[25] Xifeng Yan and Jiawei Han, "GSpan: Graph-based Substructure Pattern Mining," *in 2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, Maebashi City, Japan, 2002, pp. 721-724

[26] Y. Cai, Y. Cao, Y. Li, T. Huang and B. Zhou, "Cascading Failure Analysis Considering Interaction Between Power Grids and Communication Networks," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 530-538, January 2016.

[27] Y. Liu, I. Semertzis, A. Stefanov, P. Palensky, "Critical Components Identification for Cyber-Physical Power Systems considering Time-Varying Operational States," *in Proceedings of the 9th Workshop on Modeling and Simulation of Cyber-Physical Energy Systems*, USA, 2021, pp. 1–7.

[28] V. Tseng, C. Wu, P. Fournier-Viger, P. S. Yu, "Efficient Algorithms for Mining Top-K High Utility Itemsets," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 54–67, Jan. 2016.

[29] X. Yan; J. Han, "gSpan: graph-based substructure pattern mining," *in 2002 IEEE International Conference on Data Mining, 2002. Proceedings*, Japan, 2002, pp. 721-724.

[30] M. Kuramochi and G. Karypis. "Frequent subgraph discovery," *in Proceedings 2001 IEEE International Conference on Data Mining*, USA, 2001, pp. 313-320.

[31] B. Wang, H. Tang, C. Guo, Z. Xiu, "Entropy optimization of scale-free networks' robustness to random failures," *Physica A: Statistical Mechanics and its Applications*, vol 363, no. 2, pp. 591-596, 2006.

[32] F. Cadini, E. Zio C. Petrescu, "Using Centrality Measures to Rank the Importance of the Components of a Complex Network Infrastructure," *in Critical Information Infrastructure Security: Third Int. Workshop, CRITIS 2008*, Italy, 2009, pp. 155-167.