

Learning Cascading Failure Interactions by Deep Convolutional Generative Adversarial Network

Shuchen Huang

*Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, USA
shuang48@stevens.edu*

Junjian Qi

*Electrical and Computer Engineering
Stevens Institute of Technology
Hoboken, USA
jqj8@stevens.edu*

Abstract—In this paper, a cascading failure interaction learning method is proposed for real utility outage data. For better revealing the structure, we reorganize the failure interaction matrix based on Louvain community detection. A deep convolutional generative adversarial network (DCGAN) based method is then proposed to learn the implicit features for failure propagation in the interaction matrix. A systematic method is further developed to evaluate the performance of the learning method on missing interaction recovery and new interaction discovery. The effectiveness of the proposed method is validated on the 14-year real utility outage data from Bonneville Power Administration.

Index Terms—Blackout, cascading failure, community detection, deep convolutional generative adversarial network (DCGAN), interaction, utility outage data

I. INTRODUCTION

Large-scale blackouts have led to many component failures, significant economic losses, and severe social impacts [1]. Traditionally cascading failure study heavily relies on simulation models, such as Manchester model [2], hidden failure model [3], and OPA model [4]. Recently efforts have been made on extracting failure propagation patterns from simulated data. This includes branching process (BP) model [5], [6] and multi-type BP model [7] that extract high-level statistical information of outage propagation and the component interaction models such as influence graph [8], interaction network [9]–[11], and coupled interaction network [12] that enable an explicit study of the interactions between component outages.

Although there are many mechanisms in a cascading failure, the simulation models can only select a few of them and it is not clear how realistic the simulated cascades are. Therefore, recent efforts have been made to directly analyze real utility outage data. In [13] an influence graph is proposed to describe the statistics of cascading line outages for the 14-year real outage data from Bonneville Power Administration (BPA) [14], [15]. Then in [16] the same data is analyzed based on a generation-dependent interaction network estimated from the expectation maximization (EM) algorithm, and the unique challenges for directly analyzing real data are addressed.

This work was supported by the National Science Foundation under CAREER ECCS-2110211.

However, due to the scarcity of real outage data, there may be missing or undiscovered component interactions. Therefore, a method to mine the implicit information in the limited outage data and the statistically estimated component interactions is needed. Such challenges can be found in other real-life problems, such as protein-protein interaction detection [17], cellphone system prediction [18], and recommendation systems [19]. Various techniques have been proposed [20], such as non-negative matrix factorization (NNMF) [21] and similarity-based algorithms [17]. However, these methods are designed for undirected or other specific networks and may not work well for the component failure interaction problem that involves a weighted directed network [22].

Recently, a deep convolutional generative adversarial network (DCGAN) has been proposed [23], which is further utilized for learning the pattern features from images represented by real-world networks [22]. These networks are processed as binary images based on their adjacency matrices. However, the element values are neglected. Since the element values contain important information, the learning method in [22] needs to be improved to keep that information.

In this paper, a cascading failure interaction learning method based on DCGAN is proposed to reveal the cascading propagation patterns by extracting implicit features. The main contributions of this paper are summarized as follows.

- 1) We propose a DCGAN based learning method to extract the implicit failure propagation features in the interaction matrix estimated from real utility outage data. To improve the learning performance, a Louvain community detection based method is developed to reorganize the interaction matrix before DCGAN training.
- 2) We develop a systematic method to evaluate the performance of the learning method on missing interaction recovery and new interaction discovery. The noisy new interactions are filtered based on the degree features of the benchmark interaction matrix.

The remainder of this paper is organized as follows. Section II introduces the utility outage data and the failure interaction matrix estimated from the data. Section III explains the pre-processing of the interaction matrix based on community detection. Section IV proposes a DCGAN based method for

failure interaction learning. Section V discusses a systematic method for evaluating the performance of the learning method. Section VI validates the effectiveness of the proposed method, and finally, conclusions are drawn in Section VII.

II. CASCADING FAILURE INTERACTION ESTIMATION USING REAL UTILITY OUTAGE DATA

In this paper, the 14-year real outage data from BPA in the Transmission Availability Data System (TADS) since January 1999 is used for cascading failure analysis and prediction [14], [15]. Since cascading is defined as the uncontrolled successive loss of system elements by North American Electric Reliability Corporation (NERC), only 10,942 automatic outages are applied for analysis. The outage data is grouped into different cascades and generations. One cascade corresponds to one cascading failure sample while one generation corresponds to one stage in a cascade. Each cascade starts with initial outages in generation 0 followed by outages grouped into further generations until the cascade stops. This can be done based on the gaps in start time between successive outages [24]. Let $\mathcal{F}_g^{(m)}$ be the set of the failed components in generation g of cascade m . Assume there are M cascades listed as:

	generation 0	generation 1	generation 2	...
cascade 1	$\mathcal{F}_0^{(1)}$	$\mathcal{F}_1^{(1)}$	$\mathcal{F}_2^{(1)}$	...
cascade 2	$\mathcal{F}_0^{(2)}$	$\mathcal{F}_1^{(2)}$	$\mathcal{F}_2^{(2)}$	...
...	...	...	...	...
cascade M	$\mathcal{F}_0^{(M)}$	$\mathcal{F}_1^{(M)}$	$\mathcal{F}_2^{(M)}$	...

There are $n = 582$ components (lines) and $M = 6,687$ cascades with 10,779 automatic outages. Based on this data, we can estimate the component interactions, which are organized into an interaction matrix. Due to obvious evolution among generations and high heterogeneity among cascades, an interaction matrix is estimated for any two consecutive generations based on the EM algorithm developed in [16]. Since the largest generation number is 109, there are 108 interaction matrices ($\mathbf{B}_g \in \mathbb{R}^{n \times n}$, $g = 0, \dots, 107$). The elements of $\mathbf{B}_g$ are the empirical probabilities that component j fails in generation $g+1$ following component i failure in generation g . Because the interaction matrix $\mathbf{B}_0 = [b_{ij}] \in \mathbb{R}^{n \times n}$ ($b_{ij} \in [0, 1]$) between generation zero and generation one involves the largest number of outages and failure interactions, in this paper we only focus on $\mathbf{B}_0$. Note that the elements with very small values (those with $b_{ij} < 10^{-6}$) are set to be zero.

III. COMMUNITY DETECTION AND INTERACTION MATRIX PRE-PROCESSING FOR BETTER STRUCTURE

The $\mathbf{B}_0$ matrix is treated as a single tunnel image $\mathcal{I}_0$, where pixel values are the corresponding element values of $\mathbf{B}_0$. To highlight its structure, in Fig. 1(a) we show $\mathcal{I}_0$ by ignoring its element values. It is seen that $\mathcal{I}_0$ does not have a clear structure which will make the interaction learning very challenging since the learning performance depends on the structure of the image. To address this problem we rearrange $\mathbf{B}_0$ so that the corresponding image has a better structure.

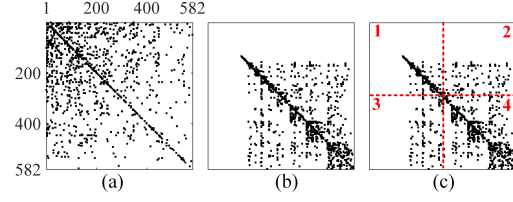


Fig. 1. Images that only keep the structures for: (a) $\mathcal{I}_0$; (b) $\tilde{\mathcal{I}}$; and (c) $\tilde{\mathcal{I}}^k$, $k = 1, \dots, 4$.

We first assign the components to different communities where the components in the same community have dense connections while those from different communities have sparse connections. The communities are obtained by the Louvain community detection technique [25] for the weighted directed interaction network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ obtained by treating $\mathbf{B}_0$ as the adjacency matrix, where $\mathcal{V} = \{1, 2, \dots, n\}$ is the set of network nodes as components and $\mathcal{E}$ is the set of links.

Specifically, the Louvain community detection method optimizes a modularity measure Q that is defined as [26]:

$$Q = \frac{1}{2b} \sum_i \sum_j \left(b_{ij} - \frac{\sum_j b_{ij} \sum_i b_{ij}}{2b} \right) \delta(c_i, c_j), \quad (1)$$

where $2b = \sum_i \sum_j b_{ij}$ is the sum of the link weights in $\mathcal{G}$, c_i and c_j are the communities that nodes i and j are assigned to, and δ is the Kronecker delta function. The highest Q value of a network appears when the optimal community assignment is achieved. After employing the Louvain method to $\mathcal{G}$ in Gephi, $C = 24$ communities are detected and the node sets of the communities are denoted by $\mathcal{V}_1, \dots, \mathcal{V}_C$.

Then, to obtain an image with a good structure that the connection density increases along the diagonal, the communities are sorted in ascending order. Components are further reordered based on the community sequence and the interactions between components are mapped based on the new component order, maintaining the actual interactions between components. By doing so the matrix $\mathbf{B}_0$ is converted to a new interaction matrix $\tilde{\mathbf{B}}_0 = [\tilde{b}_{ij}] \in \mathbb{R}^{n \times n}$ for which the corresponding image $\tilde{\mathcal{I}}$ as in Fig. 1(b) has a better structure.

Without losing its structure, the reorganized matrix $\tilde{\mathbf{B}}_0$ is partitioned into four sub-matrices $\tilde{\mathbf{B}}_0^k = [\tilde{b}_{ij}^k]$, $k = 1, \dots, 4$ with the same dimension to reduce computational memory during training. The numbers of nonzero elements in $\tilde{\mathbf{B}}_0^k$, $k = 1, \dots, 4$ are 276, 114, 124, and 674, respectively. Fig. 1(c) shows the four sub-images $\tilde{\mathcal{I}}_0^k$, $k = 1, \dots, 4$ generated from these sub-matrices and how $\tilde{\mathbf{B}}_0$ is partitioned.

IV. FAILURE INTERACTION LEARNING BY DCGAN

Here, a DCGAN-based method is developed to learn the missing or undiscovered failure interactions. The interaction features of the TADS real outage data are hidden in the four structural sub-images obtained in Section III. Without losing major features, a small portion of interactions are changed in each sub-image by the proposed image generation method to generate a series of perturbed images. To capture

the interaction features of these images, DCGAN uses an adapted generative adversarial network (GAN), which has a generator and a discriminator that confront each other in the adversarial training [27]. We adjust the network settings for specific image structures to obtain a better performance. At the end of the training, the learned features are presented in the predicted images, helping reveal the missing or undiscovered interactions. The details of the perturbed image generation method and the DCGAN training will be introduced below.

A. Perturbed Image Generation

For each sub-matrix $\tilde{B}_0^k$, $k = 1, \dots, 4$, the corresponding matrix of the sub-image $\tilde{I}_0^k$, we define an *interaction set* $\tilde{\mathcal{E}}^k = \{(i, j) | \tilde{b}_{ij}^k \neq 0\}$ and a *blank set* $\tilde{\mathcal{H}}^k = \{(i, j) | \tilde{b}_{ij}^k = 0\}$. The set of interactions in $\tilde{B}_0$ is $\tilde{\mathcal{E}} = \tilde{\mathcal{E}}^1 \cup \dots \cup \tilde{\mathcal{E}}^4$. To evaluate the learning performance in recovering the missing interactions, some interactions are chosen as known missing interactions and are deleted for DCGAN to recover. Specifically, the interaction set is divided into two sets with $\tilde{\mathcal{E}}^k = \tilde{\mathcal{E}}_o^k \cup \tilde{\mathcal{E}}_p^k$ and $\tilde{\mathcal{E}}_o^k \cap \tilde{\mathcal{E}}_p^k = \emptyset$ where $\tilde{\mathcal{E}}_o^k$ is the *observed set* and $\tilde{\mathcal{E}}_p^k$ is the *probe set*. Setting the elements in $\tilde{B}_0^k$ corresponding to the probe set to be zero, an observed matrix $\hat{B}_0^k$ is obtained for which the corresponding image is $\hat{I}_0^k$. By learning the interaction features from the observed interaction set in $\hat{B}_0^k$, the learning method will capture the major propagation patterns and recover the missing interactions. In this paper, the probe set accounts for 10% of the interaction set.

Since only one image corresponding to the observed matrix is not sufficient for training, each observed matrix $\hat{B}_0^k$, $k = 1, \dots, 4$, is utilized as the sample base for generating similar failure interaction images. To benefit the implicit interaction learning, the perturbed images to be generated should share similar features of the sub-images $\hat{I}_0^k$, $k = 1, \dots, 4$ and at the same time have good sample diversity. To achieve this, small changes to $\hat{B}_0^k$ by further interaction deleting and adding are proposed to generate perturbed images. The interactions to be deleted are selected randomly while the interactions to be added need to be chosen carefully to avoid providing many false and extra features for interaction learning.

Since the components in the same community have closer bonds, implicit interactions within one community have a higher existence likelihood. This type of interaction is added to provide image diversity and assist the learning method to find implicit interactions based on the component bonds.

Specifically, each perturbed matrix and its corresponding image are generated from $\hat{B}_0^k$ through two steps.

- 1) For an observed matrix $\hat{B}_0^k$, 5% of the interactions in $\tilde{\mathcal{E}}^k$ are randomly selected and set to be 0.
- 2) Let $|S|$ be the number of elements in a set S . $0.05|S|$ number of new interactions are added with values as 0.068, which is the mean value of all interactions in $\tilde{\mathcal{E}}$. The components i and j of each interaction are selected based on the community assignment obtained in Section III. A component $i \in \mathcal{V}_l$ for $\forall l = 1, \dots, C$ is first randomly selected, and the components in the same community as component i are assigned a larger

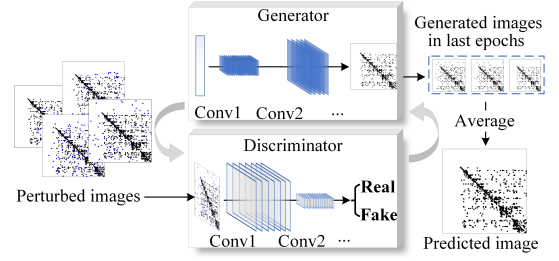


Fig. 2. DCGAN for cascading failure interaction learning.

weight than the other components. Then component j is randomly decided by the assigned weights.

We repeat the above steps until 1,000 perturbed images for each $\hat{I}_0^k$, $k = 1, \dots, 4$, are generated.

B. DCGAN for Failure Interaction Learning

The essential part of the proposed learning method is the DCGAN training. By importing the perturbed images obtained in Section IV-A, the DCGAN method is executed four times to learn the hierarchy of each interaction sub-image and to capture the propagation patterns. The DCGAN training process for cascading failure interaction learning is shown in Fig. 2. During training, the perturbed images that contain important interaction features from the outage data are imported as “real” images for the discriminator to learn the failure propagation patterns through the updating of the generator and discriminator. The generator presents the learned interaction features in the generated images in each epoch. Eventually, a predicted image is provided by the generator after the entire training.

1) *Generator*: A generator generates S fake interaction images similar to the perturbed matrices. A uniform distribution is the input of the generator, and with the fractional-strided convolutions, the generator converts the distribution into fake images, imitating the interaction features that actual perturbed images share. After receiving recognition feedback from the well-trained discriminator about the fake interaction patterns, the parameters of the generator are updated to generate more convincing images that have closer propagation patterns. At the end of DCGAN training, the final generated image whose matrix has the average values of the elements of the matrices corresponding to the S generated images in every epoch.

The final interaction features are obtained from the generator in the last L epochs. The L generated images are collected and the elements of the final predicted matrix $\bar{B}_0^k = [\bar{b}_{ij}^k]$ are the average element values of the corresponding matrices of the L images (the elements less than 10^{-6} are removed). According to the results in Section VI, the averaging does not cause serious information loss. Stacking four $\bar{B}_0^k$ in the way shown in Fig. 1(c), the entire predicted matrix $\bar{B}_0 = [\bar{b}_{ij}]$ and its image $\bar{I}_0$ are obtained. Also, we define a *predicted interaction set* $\bar{\mathcal{E}} = \{(i, j) | \bar{b}_{ij} \neq 0\}$.

2) *Discriminator*: A discriminator decides whether the input images are perturbed images with real interaction features or fake images produced by the generator. It uses strided

convolutions to extract more details of the real interaction features for better detecting fake images. In the last convolution, the output of the discriminator is flattened and put into a single Sigmoid output, and the discriminator provides the classification results about the received interaction images.

By adversarial training and the well-trained generator, the implicit interaction features embodied in the specific data distributions of the perturbed images are obtained by DCGAN. Hence, DCGAN can capture the propagation patterns and reveal the interactions that match the major patterns.

3) *Implementation details for cascading failure interaction learning*: For each failure interaction sub-matrix, we set $S = 500$ and $L = 20$ for the DCGAN learning model. Due to the different structures presented in the four interaction sub-images, the settings of the filter and the hyper-parameters of DCGAN are different. Based on our numerical experiments, larger convolution filters work better for more sparse images. Therefore, for $\hat{I}_0^2$ that has the least nonzero elements, the third and fourth convolutional filters of its generator are set as 6×6 and 7×7 , and the first and second filters of the discriminator are set as 7×7 and 6×6 . The corresponding filters for the other sparse image $\hat{I}_0^3$ are set as 6×6 and 5×5 , while for $\hat{I}_0^1$ and $\hat{I}_0^4$, they are 5×5 . Moreover, we set different hyper-parameters for different sub-images. For $\hat{I}_0^2$, the learning rate and the batch size are, respectively, chosen as 0.00015, 128 for better performance. Differently, for $\hat{I}_0^1$, $\hat{I}_0^3$, and $\hat{I}_0^4$, the learning rate is 0.0002, the batch size is 64. The epoch numbers for each sub-image are 100, 120, 120, and 120.

V. PERFORMANCE EVALUATION

The purpose of interaction learning is to reveal the intrinsic propagation pattern and the implicit component failure interactions that are missing or undiscovered. Therefore, whether the learning method can effectively *recover missing interactions* or *discover new interactions* are crucial for evaluation.

Missing interactions refer to the existing interactions that are intentionally removed. The evaluation of the recovery of these interactions is easier since their original locations and values are known. Some of the estimated interactions can be deleted and after training their predicted values can indicate the recovery ability of the learning method.

In contrast, the undiscovered interactions refer to those that do not exist in the estimated interaction matrix due to the scarcity of real outage data. Their number and likelihood scale are difficult to decide based on the scarce data. Hence, a benchmark interaction matrix generated based on much more cascades than those in the real outage data is needed to help verify the discovered new interactions.

Due to distinct characteristics of the two different types of interactions, the recovery and discovery abilities of the proposed learning method are evaluated separately below.

A. Recovery of Missing Interactions

Good recovery ability is demonstrated when removed interactions in the probe set are recovered with predicted values that match well with the known interaction values. Furthermore,

the probe set in the predicted matrices should have higher values than the blank set in general, avoiding a disordered learning method that predicts many improbable interactions.

1) *Recovery rate of the probe set*: From $\bar{B}_0^k$, $k = 1, \dots, 4$, the *predicted interaction set* is decided as $\bar{\mathcal{E}}^k = \{(i, j) | \bar{b}_{ij}^k \neq 0\}$, based on which we define a recovery rate R_1 as:

$$R_1 = \frac{|\bar{\mathcal{E}}^k \cap \tilde{\mathcal{E}}_p^k|}{|\tilde{\mathcal{E}}_p^k|}. \quad (2)$$

The higher the recovery rate is, the better the recovery ability of the proposed learning method.

2) *Metrics comparing probe and blank sets*: Instead of only focusing on the probe set, the overall value level of the probe set interactions compared with those of the blank set needs to be evaluated. Two scale-independent metrics, the area under the receiver operating characteristic curve (AUC) [28] and the probe set rank (PSR) inspired by the metrics in [29] are used to compare the overall values of the probe and blank sets.

- AUC depends on the results of z independent comparisons: for each comparison, two predicted values $\bar{b}_{i_1 j_1}^k$ where $(i_1, j_1) \in \tilde{\mathcal{E}}_p^k$ and $\bar{b}_{i_2 j_2}^k$ where $(i_2, j_2) \in \mathcal{H}^k$ are randomly selected for comparison. There are three outcomes: i) $\bar{b}_{i_1 j_1}^k > \bar{b}_{i_2 j_2}^k$, ii) $\bar{b}_{i_1 j_1}^k = \bar{b}_{i_2 j_2}^k$ or $|\bar{b}_{i_1 j_1}^k - \bar{b}_{i_2 j_2}^k| \leq 10^{-3}$, and iii) $\bar{b}_{i_1 j_1}^k < \bar{b}_{i_2 j_2}^k$. Let z_1 and z_2 respectively be the number of the first and second outcomes in z comparisons, and AUC is calculated as:

$$\text{AUC} = \frac{z_1 + z_2}{z}, \quad (3)$$

where z is set to be 2,000 to ensure that most interactions in the probe set are included in comparisons.

- PSR indicates the overall value ranking of the elements of $\bar{B}_0^k$ in the probe set among those in both the probe and blank sets. After sorting the predicted values $\bar{b}_{ij}^k$ in both the probe and blank sets in an ascending order, a series of ranks $r_i^k, i = 1, 2, \dots, |\tilde{\mathcal{E}}_p^k| + |\mathcal{H}^k|$ are obtained. Let $\mathcal{P}$ be a set of the i 's that correspond to a probe set element. Then, focusing on the ranks of the probe set elements, PSR is calculated as:

$$\text{PSR} = \frac{1}{|\tilde{\mathcal{E}}_p^k|} \sum_{i \in \mathcal{P}} \frac{r_i^k}{|\tilde{\mathcal{E}}_p^k| + |\mathcal{H}^k|}. \quad (4)$$

Higher values of AUC and PSR indicate that the learning method tends to assign higher values of $\bar{b}_{ij}^k$ for the probe set than the blank set, thus being able to recover the probe set while not producing many new interactions in the blank set.

B. Discovery of New Interactions

The proposed DCGAN method can discover new interactions from the learned failure propagation patterns. To examine these newly generated interactions, a benchmark interaction matrix is estimated from a large number of cascades. Although the real outage data is limited, the generation-dependent interaction model proposed in [16] can generate as many cascades as needed to better capture and further extend what

has been observed in real outage data. This model is used to generate $M_{\max} \gg M$ cascades based on $\mathbf{B}_0$. From these $M_{\max}$ cascades a new interaction matrix $\hat{\mathbf{B}}_0$ is estimated using the EM algorithm to capture the interactions between generation zero outages and generation one outages. Note that the elements less than 10^{-6} are removed.

After pre-processing $\hat{\mathbf{B}}_0$ in the same way as in Section III, a benchmark matrix $\mathbf{B}_0 = [\hat{b}_{ij}]$ and its corresponding images $\tilde{\mathcal{I}}_0$ are obtained. Define a *benchmark interaction set* $\hat{\mathcal{E}} = \{(i, j) | \hat{b}_{ij} \neq 0\}$ and a *benchmark blank set* $\hat{\mathcal{H}} = \{(i, j) | \hat{b}_{ij} = 0\}$. Compared with $\mathbf{B}_0$, the set of new interactions revealed in $\hat{\mathbf{B}}_0$ is called the *benchmark new interaction set* $\hat{\mathcal{E}}_n = \hat{\mathcal{E}} \setminus \hat{\mathcal{E}}$.

Further, a benchmark interaction network $\hat{\mathcal{G}}(\mathcal{V}, \hat{\mathcal{E}})$ is generated by treating $\hat{\mathbf{B}}_0$ as the adjacency matrix. The sets of the components that appear as the source and destination nodes in $\hat{\mathcal{E}}_n$ are respectively denoted by $\mathcal{V}_n^s$ and $\mathcal{V}_n^d$. The 10th percentile of the out-degree (in-degree) of the components in $\mathcal{V}_n^s$ ($\mathcal{V}_n^d$) is denoted by p_{10}^+ (p_{10}^-). Since p_{10}^+ and p_{10}^- provide important information about the features of the new interactions, they are used as references to filter the noises. Specifically, for a predicted new interaction in $\hat{\mathcal{E}} \setminus \hat{\mathcal{E}}$, if the out-degree of the source component is not greater than p_{10}^+ or if the in-degree of the destination component is not greater than p_{10}^- , it is removed. The remaining predicted new interactions are denoted by $\hat{\mathcal{E}}_n$, which can be classified into two different types below.

- If the new interaction in $\hat{\mathcal{E}}_n$ also appears in $\hat{\mathcal{E}}$, it is called a *verifiable new interaction*. To indicate if the new interactions in the benchmark matrix can be discovered by the learning method, we define a *discovery rate* as:

$$R_2 = \frac{|\hat{\mathcal{E}}_n \cap \hat{\mathcal{E}}|}{|\hat{\mathcal{E}}_n|}. \quad (5)$$

The higher R_2 is, the better the discovery ability the learning method has.

- If the new interaction in $\hat{\mathcal{E}}_n$ does not appear in $\hat{\mathcal{E}}$, it is called a *unverifiable new interaction*. We define *noise rate* R_3 as the ratio between the number of unverifiable new interactions and $|\hat{\mathcal{H}}|$:

$$R_3 = \frac{|\hat{\mathcal{E}}_n \setminus \hat{\mathcal{E}}|}{|\hat{\mathcal{H}}|}. \quad (6)$$

The smaller R_3 is, the better the noise suppression of the learning method is when discovering new interactions. Although these unverifiable new interactions cannot be verified by the benchmark matrix, they may provide useful information about failure propagation which could be further verified with an improved benchmark methodology.

VI. RESULTS

All computations are performed on a computer with Intel Core (TM) i7-11700 CPU. DCGAN is implemented in Python 3.6.13 based on Tensorflow-cpu 2.3.0. A traditional NNMF method [21] is performed on MATLAB R2020b with function

TABLE I
RECOVERY RATE OF THE PROBE SET

k	$ \hat{\mathcal{E}}_p^k $	$ \hat{\mathcal{E}}^k \cap \hat{\mathcal{E}}_p^k $	R_1
1	28	28	1
2	11	11	1
3	12	11	0.92
4	67	67	1

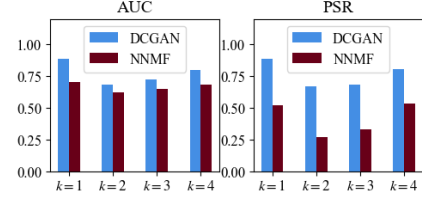


Fig. 3. AUC and PSR for DCGAN and NNMF.

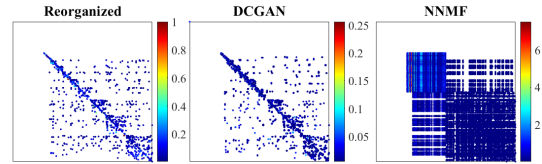


Fig. 4. Predicted images from DCGAN and NNMF.

nnmf for comparison. We take the average metric values of the last L epochs as the metric results of DCGAN and take the average metric values over 200 runs as the results of NNMF. The elements less than 10^{-6} are removed from NNMF predicted matrices. The learning performance is evaluated following the evaluation method proposed in Section V.

A. Recovery Ability

The details about the recovery of the probe set are listed in Table I. The recovery rates for $\bar{\mathbf{B}}_0^1$, $k = 1, \dots, 4$ are all close to one, and only one missing interaction in $\bar{\mathbf{B}}_0^3$ is not recovered, indicating good learning performance.

The AUC and PSR of the proposed DCGAN method compared with NNMF are shown in Fig. 3. The recovery ability of DCGAN outperforms NNMF in all four sub-matrices. The highest AUC and PSR of DCGAN are close to 0.9 for the first sub-matrix. For the most sparse images when $k = 2, 3$, the AUC and PSR of DCGAN are still higher than 0.7.

Fig. 4 shows the final predicted images of DCGAN and NNMF. To show the major structure in the images, only interactions greater than 0.043, which is the 10th percentile of all interactions in $\mathbf{B}_0$, are displayed. It is seen that the DCGAN predicted image matches very well with the reorganized image $\tilde{\mathcal{I}}_0$, while the predicted image from NNMF has a lot of noise.

B. Discovery Ability Comparison

The $M_{\max}$ in Section V-B is decided as $130M$ under which the number of interactions in the benchmark matrix saturates. The p_{10}^+ and p_{10}^- in Section V-B are 2 and 3, respectively. With noise filtering for both DCGAN and NNMF, their R_2 and R_3

TABLE II
DISCOVERY ABILITY COMPARISON

Metric	DCGAN	NNMF
R_2	0.7329	0.2715
R_3	0.0980	0.0598

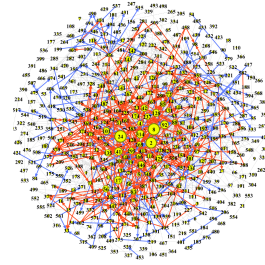


Fig. 5. $\hat{G}$ with highlighted verifiable new interactions.

values are listed in Table II. With $R_2 = 0.73$ and $R_3 = 0.10$, the proposed DCGAN method has a good discovery ability without producing too much noise, and a total of 332 verifiable new interactions are obtained. By contrast, although NNMF has a lower noise rate, it has a much lower R_2 value and thus cannot discover as many new interactions.

The benchmark interaction network $\hat{G}$ is shown in Fig. 5. The size of the node is positively related to the sum of the out-degree and in-degree of the node. The red arrows denote the verifiable new interactions and the blue arrows denote the new interactions in the benchmark interaction network that cannot be discovered by the proposed method. Most verifiable new interactions are between the components with a relatively large number of interactions, indicating the necessity of failure prevention for those components with dense interactions.

VII. CONCLUSION

In this paper, a DCGAN based cascading failure interaction learning method is proposed to recover the missing interactions and discover new interactions based on limited real utility outage data. The proposed method is evaluated on 14-year real utility outage data from BPA based on the proposed systematic evaluation method. The results show that the proposed learning method can well capture the failure interaction patterns and outperform the traditional NNMF method, successfully recovering the missing interactions and discovering new interactions without producing a lot of noise.

REFERENCES

- [1] K. Sun, Y. Hou, W. Sun, and J. Qi, *Power System Control under Cascading Failures: Understanding, Mitigation, and Restoration*. Wiley-IEEE Press, Jan. 2019.
- [2] D. S. Kirschen, D. Jayaweera, D. P. Nedic, and R. N. Allan, "A probabilistic indicator of system stress," *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1650–1657, Aug. 2004.
- [3] J. Chen, J. S. Thorp, and I. Dobson, "Cascading dynamics and mitigation assessment in power system disturbances via a hidden failure model," *Int. J. Elec. Power*, vol. 27, no. 4, pp. 318–326, May 2005.
- [4] H. Ren, I. Dobson, and B. A. Carreras, "Long-term effect of the n-1 criterion on cascading line outages in an evolving power transmission grid," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1217–1225, Aug. 2008.

- [5] I. Dobson, J. Kim, and K. R. Wierzbicki, "Testing branching process estimators of cascading failure with data from a simulation of transmission line outages," *Risk Anal.*, vol. 30, no. 4, pp. 650–662, Apr. 2010.
- [6] J. Qi, I. Dobson, and S. Mei, "Towards estimating the statistics of simulated cascades of outages with branching processes," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 3410–3419, Aug. 2013.
- [7] J. Qi, W. Ju, and K. Sun, "Estimating the propagation of interdependent cascading outages with multi-type branching processes," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 1212–1223, Mar. 2017.
- [8] P. Hines, I. Dobson, and P. Rezaei, "Cascading power outages propagate locally in an influence graph that is not the actual grid topology," *IEEE Trans. Power Syst.*, vol. 32, no. 2, pp. 958–967, Mar. 2017.
- [9] J. Qi, K. Sun, and S. Mei, "An interaction model for simulation and mitigation of cascading failures," *IEEE Trans. Power Syst.*, vol. 30, no. 2, pp. 804–819, Mar. 2015.
- [10] W. Ju, J. Qi, and K. Sun, "Simulation and analysis of cascading failures on an NPCC power system test bed," in *2015 IEEE Power & Energy Society General Meeting*. IEEE, Jul. 2015, pp. 1–5.
- [11] J. Qi, J. Wang, and K. Sun, "Efficient estimation of component interactions for cascading failure analysis by EM algorithm," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3153–3161, May 2018.
- [12] L. Wang, J. Qi, B. Hu, and K. Xie, "A coupled interaction model for simulation and mitigation of interdependent cascading outages," *IEEE Trans. Power Syst.*, vol. 36, no. 5, pp. 4331–4342, Sept. 2021.
- [13] K. Zhou, I. Dobson, Z. Wang, A. Roitershtein, and A. P. Ghosh, "A markovian influence graph formed from utility line outage data to mitigate large cascades," *IEEE Trans. Power Syst.*, vol. 35, no. 4, pp. 3224–3235, Jul. 2020.
- [14] I. Dobson, B. A. Carreras, D. E. Newman, and J. M. Reynolds-Barredo, "Obtaining statistics of cascading line outages spreading in an electric transmission network from standard utility data," *IEEE Trans. Power Syst.*, vol. 31, no. 6, pp. 4831–4841, Nov. 2016.
- [15] Bonneville Power Administration Transmission Services Operations & Reliability website. [Online]. Available: <http://transmission.bpa.gov/Business/Operations/Outages>
- [16] J. Qi, "Utility outage data driven interaction networks for cascading failure analysis and mitigation," *IEEE Trans. Power Syst.*, vol. 36, no. 2, pp. 1409–1418, Mar. 2021.
- [17] I. A. Kovács *et al.*, "Network-based prediction of protein interactions," *Nat. Commun.*, vol. 10, no. 1, pp. 1–8, Mar. 2019.
- [18] V. D. Blondel, A. Decuyper, and G. Krings, "A survey of results on mobile phone datasets analysis," *EPJ Data Sci.*, vol. 4, no. 1, p. 10, Aug. 2015.
- [19] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, Feb. 2019.
- [20] P. Wang, B. Xu, Y. Wu, and X. Zhou, "Link prediction in social networks: the state-of-the-art," *Sci. China Inf. Sci.*, vol. 58, no. 1, pp. 1–38, Jan. 2015.
- [21] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Comput. Stat. Data Anal.*, vol. 52, no. 1, pp. 155–173, Sept. 2007.
- [22] X.-W. Wang, Y. Chen, and Y.-Y. Liu, "Link prediction through deep generative model," *Iscience*, vol. 23, no. 10, p. 101626, Oct. 2020.
- [23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, Nov. 2015.
- [24] I. Dobson, "Estimating the propagation and extent of cascading line outages from utility data with a branching process," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2146–2155, Nov. 2012.
- [25] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *J. Stat. Mech.*, vol. 2008, no. 10, p. P10008, Oct. 2008.
- [26] A. Arenas, J. Duch, A. Fernández, and S. Gómez, "Size reduction of complex networks preserving modularity," *New J. Phys.*, vol. 9, no. 6, p. 176, Jun. 2007.
- [27] I. Goodfellow *et al.*, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, Jun. 2014.
- [28] L. Lü and T. Zhou, "Link prediction in complex networks: A survey," *Physica A*, vol. 390, no. 6, pp. 1150–1170, Mar. 2011.
- [29] T. Zhou, J. Ren, M. Medo, and Y.-C. Zhang, "Bipartite network projection and personal recommendation," *Phys. Rev. E*, vol. 76, no. 4, p. 046115, Oct. 2007.