Electric Power Systems Research 239 (2025) 111184

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

A graph and diffusion theory-based approach for localization and recovery of false data injection attacks in power systems

Yixuan He, Jingyu Wang *, Chen Yang, Dongyuan Shi

State Key Laboratory of Advanced Electromagnetic Technology, Huazhong University of Science and Technology, Wuhan, 430074, Hubei, China

ARTICLE INFO

Keywords:
FDIA localization
Data recovery
Denoising diffusion graph models
Denoising diffusion implicit models
Line graph
Line message passing neural network

ABSTRACT

False Data Injection Attacks (FDIAs) pose a serious threat to power systems by interfering with state estimation and jeopardizing their safety and reliability. Detecting and recovering from FDIAs is thus critical for maintaining power system integrity. The increasing integration of renewable energy sources and the extensive use of power electronic devices introduce significant randomness in both power generation and loads, leading to significant power fluctuations and dynamic changes in power flows. These variations challenge the accuracy of existing FDIA detection and recovery methods. To address these challenges, an innovative data recovery framework is proposed, comprising two key stages: the FDIA localization stage and the FDIA data recovery stage. In the first stage, a Line Message Passing Neural Network (LMPNN) based FDIA localization model is employed to precisely identify the attacked data and generate a mask input for the recovery stage. In the data recovery stage, an FDIA data recovery model, named Denoising Diffusion Graph Models (DDGM), is designed to recover data with minimal error while conforming to the physical laws of the grid. Both models utilize node graph and line graph representations to depict measurements on buses and branches. By leveraging an optimized graph neural network, and inviting a loop-structured framework that combines a denoising diffusion model with a graph neural network these models effectively extract data features and inherent dynamic properties, enabling superior localization of FDIAs both in node and edge spaces and ensuring accurate recovery of compromised data even in the presence of high uncertainty and significant power fluctuations. By incorporating physical laws through a customized loss function embedding Kirchhoff's circuit laws into the training process of DDGM, the model ensures the recovered data to be physically consistent with power system dynamics. Experimental validations on IEEE 39-bus and 118-bus test systems, under conditions of high fluctuations in generation and loads, demonstrate that the proposed models outperform existing methods, achieving significant improvements in accuracy and robustness.

1. Introduction

With the development of the smart grid, traditional power systems are deeply coupled with information and communication systems, revolutionizing into the Cyber-Physical Power System (CPPS). This integration facilitates real-time monitoring, increases observability and controllability, and enhances the safety and stability of power systems. However, the reliance on networked communication systems has made CPPSs more susceptible to cyber attacks. False Data Injection Attack (FDIA) is a typical attack elaborately designed for interfering with power system state estimation by injecting false data vectors into measurement vectors [1]. This attack can bypass the Bad Data Detection (BDD) module, resulting in wrong state estimates, which may further lead to erroneous decision-making in control center and thus severely disrupt the normal operation of power systems.

Due to the high stealth and destructive consequences of FDIA, the CPPS community has devoted much effort to its defensive countermeasures. To prevent tampered measurements from affecting subsequent energy management applications, an intuitive countermeasure is to first accurately detecting and locating the attacked measurements, and then recovering these measurements. This approach has two basic requirements: accurate detection and localization of the attack and precise recovery of the original data. Several studies have followed this sequential approach to attack detection and recovery [2–6].

However, the increased use of renewable energy and power electronic devices [7] has led to more pronounced power fluctuations, reducing the accuracy of FDIA data recovery. Data recovery accuracy involves minimizing errors between the recovered and original data (statistical accuracy) and ensuring the recovered data adheres to physical principles like Kirchhoff's circuit laws physical accuracy. Current

* Corresponding author.
E-mail addresses: yixuanhe@hust.edu.cn (Y. He), jywang@hust.edu.cn (J. Wang), yangchen1003@hust.edu.cn (C. Yang), dongyuanshi@hust.edu.cn (D. Shi).

https://doi.org/10.1016/j.epsr.2024.111184
Received 29 July 2024; Received in revised form 17 October 2024; Accepted 21 October 2024
Available online 5 November 2024

methods face challenges due to inaccurate FDIA localization algorithms and data recovery algorithms that fail to maintain both statistical and physical accuracy, particularly during significant fluctuations in new energy sources.

To address these issues, the paper proposes an innovative FDIA data recovery framework that enhances the accuracy of both FDIA localization and data recovery while ensuring compliance with physical laws. This framework includes two models: a Line Message Passing Neural Network (LMPNN) for FDIA localization and a Denoising Diffusion Graph Model (DDGM) for data recovery. The FDIA localization model integrates an LMPNN and a hidden space to produce a mask that serves as input for the data recovery model. The DDGM features a two-stage circular structure incorporating a Denoising Diffusion Implicit Model (DDIM), LMPNN, a hidden space, and physical constraints, allowing the generation of recovered data that aligns with the ongoing operational dynamics of the smart grid. By combining the diffusion models with graph-based methodologies that concurrently process data from both branches and buses, our research establishes a robust framework that excels in FDIA detection and data recovery in smart grids. The four main contributions of this paper are outlined as follows:

1. Innovative FDIA Localization and Data Recovery Framework: This paper is one of the few studies on FDIA localization and data recovery that considers power fluctuations. The proposed data recovery framework, comprised of the FDIA localization model and data recovery model, offers a notable improvement in accuracy, providing a more robust defense against cyberattacks and enhancing grid security.
2. Node Graph and Line Graph Representation for Smart Grids, and LMPNN for FDIA Localization: To capture the inherent dynamic characteristics of power flows on branches and power injections on buses in smart grids, a node graph and its corresponding line graph representation of smart grid features are proposed. Additionally, this paper proposes a modified graph neural network called LMPNN, enabling simultaneous processing of data from branches and buses in both node and edge space. This advanced graph representation allows for more accurate information extraction and updating.
3. DDGM for FDIA Data Recovery: Expanding upon the successful application of DDIM in image generation, its utility is extended to tackle the data recovery challenges posed by FDIA. Additionally, this paper proposes a new model, DDGM, which inherits the robustness of DDIM in stable and high-quality sample generation.
4. Physics-Informed Data-Driven Approach: By integrating the fundamental physical principles of power systems, specifically Kirchhoff's circuit laws, into the loss function of DDGM, the recovered data more accurately conforms to the data relationships and physical laws governing the current state of the power system. This paper is one of the few studies that improves data recovery accuracy while ensuring adherence to real-world operational dynamics in FDIA scenarios.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work and draws out the rationale behind choosing the method in this paper. Section 3 introduces the basic knowledge of FDIA, DDIM, and MPNN. Section 4 describes the proposed data recovery framework in detail. Section 5 validates the effectiveness and superiority of the proposed models via experiments on the IEEE 39-bus and 118-bus test systems. Finally, Section 6 concludes this paper.

2. Related work

Broadly, FDIA data recovery research consists of two key steps: accurately detecting and locating the attacked measurements, and then recovering these measurements based on the detection results.

FDIA detection and localization leverage methodologies rooted in both model-driven approaches and machine learning techniques [8,9]. Model-driven methods, such as applying the Square-Root Extended Kalman Filter (SREKF) to FDIA detection [10], require accurate modeling of the smart grid and precise detection criteria. However, these methods often struggle to ensure comprehensive and accurate results due to the complex interaction process of the power system and the massive state space. In contrast, machine learning methods, which are less constrained by model complexity, can achieve superior performance. Prominent deep learning approaches include recurrent neural networks (RNN) like Long Short-Term Memory (LSTM) [11] and Convolutional Neural Networks (CNN) [12]. With rapid advancements in deep learning technology, many FDIA detection researches based on machine learning have achieved relatively high accuracy. Despite these advancements, existing FDIA detection methods are often tailored to specific smart grid configurations and are effective only for stationary data [13]. This limitation makes them less adaptable to changing operating conditions caused by power fluctuations. To overcome the generalization issues and maintain FDIA detection accuracy under conditions of power fluctuation, it is necessary to consider the intrinsic physical dynamics and interactions of measurements. Given the connectivity and interactions within smart grids, power system topologies can be modeled as graphs, with each grid bus and branch represented as a node and an edge, respectively [14]. Therefore, applying a graph neural network to exploit these inherent graph structures, such as using the Graph Convolutional Attention Network (GCAT) for FDIA detection [15], is a rational approach. However, the existing methods only process the measurements in the node space, neglecting edge data representing power flows on branches. Our innovative approach addresses this gap by developing a node graph and a corresponding line graph to process data in both the node and edge spaces via a modified graph neural network. By considering both power flows on branches and power injections on buses, as well as their physical implications, our method enhances the feature extraction of smart grids. This comprehensive processing can further enhance the FDIA detection and localization performance.

FDIA data recovery is usually the next step after attack localization, yet this crucial step has received relatively limited attention in research. However, a few studies have considered both the detection and recovery of cyberattacks. Both [16,17] utilize Denoising Autoencoders (DAEs) to recover the damage caused by cyberattacks. [18] proposed an attention-based temporal convolutional Denoising Autoencoder to identify FDIA locations and replace compromised measurements with reconstructed values. Besides, two main approaches to data recovery are currently applied in data imputation scenarios. The first approach involves traditional missing data imputation, primarily based on statistical learning. Examples include the iterative estimation algorithm based on singular value decomposition (SVD) [19], the mean–variance method [2], and the partial canonical identity (PCI) matrix method [20], etc. However, they require substantial computing power compared to deep learning algorithms and cannot guarantee immediate data recovery. Furthermore, in scenarios where a significant volume of data is attacked or where some attacked data is misjudged as correct, interpolation performance can be severely affected. The second approach involves the utilization of RNN such as LSTM [21] and Gate Recurrent Unit (GRU) [22]. Recent research has also applied variational auto-encoder (VAEs) and Generative Adversarial Networks (GANs) [23] to data recovery by predicting measurements and replacing attacked data with corresponding predicted values. These networks rely on training with continuous historical data and focus on future trend prediction, enabling more accurate results under steady-state conditions. However, they often overlook the intrinsic physical correlations of the measured data. Consequently, if the system state experiences significant fluctuations, the prediction results may not align well with the current state. Moreover, the performance of DAEs is constrained by the quality and quantity of the training data, often requiring substantial labeled

datasets. VAEs suffer from inherent shortcomings, such as the difficulties of tuning hyperparameters and generalizing specific generative model structures to different databases. The training process of GANs is also problematic, characterized by low stability and susceptibility to issues such as mode collapse and oscillation. This instability becomes particularly problematic when dealing with extensive missing data, as the GAN generator may struggle to obtain sufficient global information, making it difficult to accurately recover attacked data.

To effectively address the challenge of FDIA data recovery, new methods are needed to improve accuracy. Denoising Diffusion Implicit Models (DDIM) [24], widely used in image generation and restoration [25], offer significant advantages over existing generative models in power systems. Diffusion models can produce realistic measurements with high randomness, making them suitable for generating diverse, high-quality data samples and resulting in a more stable training process that is less susceptible to issues like mode collapse or oscillation [26]. Recent research has explored data imputation algorithms based on diffusion models, such as Structured State Space Diffusion models (SSSD) [27], which have shown promising results. However, due to the randomness of the sample generation process of diffusion models, the data they generate may not align with the data relationships inherent in the current state of the system, and the results may be physically inconsistent. Our innovative approach addresses these challenges by integrate DDIM with a modified graph neural network, similar to the FDIA localization model, and incorporate the physical constraints of power systems. This integration not only provides more diverse and higher-quality predictions but also leverages the inherent graph structures of power systems to better fit the current operational state and physical laws. Consequently, the effectiveness and reliability of FDIA data recovery are significantly enhanced.

3. Preliminary knowledge

3.1. False data injection attack

Smart grid modeling relies on the continuous stream of real-time measurements and static system data, which mainly includes system parameters and configurations. Power system measurements $\boldsymbol{z}$ collected by SCADA systems can be expressed as

$$\boldsymbol{z} = h(\boldsymbol{x}) + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{z} \in \mathbb{R}^m$ is the measurement vector, $\boldsymbol{x} \in \mathbb{R}^n$ the state vector, $h(\cdot)$ the nonlinear measurement function following the ac power flow model, and $\boldsymbol{e} \in \mathbb{R}^m$ the measurement error vector. In practice, the BDD module is typically employed to identify and eliminate outliers in data. The residual between the measured and estimated values is used for BDD. If this residual, denoted as $\boldsymbol{r} \in \mathbb{R}^m$, is less than the threshold τ, the measurements are considered acceptable and pass through the BDD module, which can be expressed as

$$\|\boldsymbol{r}\| = \|\boldsymbol{z} - h(\hat{\boldsymbol{x}})\| < \tau \tag{2}$$

FDIA aims to inject the attack vector $\boldsymbol{a} \in \mathbb{R}^m$ into $\boldsymbol{z}$. After tampering, the measurement vector becomes $\boldsymbol{z}^a = \boldsymbol{z} + \boldsymbol{a}$, and the state vector becomes $\hat{\boldsymbol{x}}^a = \hat{\boldsymbol{x}} + \boldsymbol{c}$, where $\boldsymbol{c}$ is the bias of state vector. In this case, the residual is expressed as

$$\|\boldsymbol{r}^a\| = \|\boldsymbol{z}^a - h(\hat{\boldsymbol{x}}^a)\| = \|\boldsymbol{z} + \boldsymbol{a} - h(\hat{\boldsymbol{x}} + \boldsymbol{c})\| \tag{3}$$

To make FDIA bypass the BDD module, $\boldsymbol{a}$ should be designed elaborately to ensure $\|\boldsymbol{r}^a\| < \tau$. Under normal circumstances, attackers only need to know part of the network topology and target a subset of the nodes to launch an effective attack [3]. It can be expressed as

$$\boldsymbol{z}_i^a = \begin{cases} \boldsymbol{z}_i + a & \text{if } i \in \Omega_a \\ \boldsymbol{z}_i & \text{otherwise} \end{cases} \tag{4}$$

where Ω_a is the attack area, and i stands for the ith measurement.

3.2. Denoising Diffusion Implicit Models (DDIM)

The denoising diffusion model includes the forward diffusion process, denoted as q, and the backward generating process, denoted as p. The diffusion process q adds Gaussian noise to the data until it becomes pure noise data, forming an approximate prior distribution. Conversely, the generating process p gradually removes the noise, transforming the pure noise data back into samples that resemble the original data. Unlike Denoising Diffusion Probabilistic Models (DDPM) [28], which relies on Markov chains, Denoising Diffusion Implicit Models (DDIM) [24] is based on non-Markov chains. This non-Markovian approach allows the model to skip steps in the denoising process without needing to access all past states before the current state, thus enhancing the training speed by 10 to 50 times. DDIM's diffusion process q is shown in Eq. (5),

$$q_\sigma\left(x_{1:T}|x_0\right) := q_\sigma\left(x_T|x_0\right)\prod_{t=2}^{T} q_\sigma\left(x_{t-1}|x_t, x_0\right) \tag{5}$$

where $q_\sigma\left(x_T|x_0\right) = N\left(\sqrt{\alpha_T}x_0, \left(1-\alpha_T\right)I\right)$ and for all $t > 1$,

$$q_\sigma\left(x_{t-1}|x_t, x_0\right) = N\left(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}}, \sigma_t^2\mathrm{I}\right) \tag{6}$$

As shown in Eq. (6), the posterior distribution is still Gaussian distribution. The forward diffusion process derives from Bayes' rule, shown as follows.

$$q_\sigma\left(x_t|x_{t-1}, x_0\right) = \frac{q_\sigma\left(x_{t-1}|x_t, x_0\right)q_\sigma\left(x_t|x_0\right)}{q_\sigma\left(x_{t-1}|x_0\right)} \tag{7}$$

A single step of the generating process p is defined as

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon_\theta^{(t)}\left(x_t\right)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\varepsilon_\theta^{(t)} + \sigma_t\varepsilon_t \tag{8}$$

where $\sigma_t = \eta\cdot\sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}\sqrt{1-\frac{\bar{\alpha}_t}{\bar{\alpha}_{t-1}}}$. In Eq. (8), $\frac{x_t - \sqrt{1-\bar{\alpha}_t}\varepsilon_\theta^{(t)}\left(x_t\right)}{\sqrt{\bar{\alpha}_t}}$ is the predicted x_0, $\sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\varepsilon_\theta^{(t)}$ is the direction pointing to x_t, and $\sigma_t\varepsilon_t$ is random noise. When $\eta = 1$, the generating process follows the framework of DDPM. Conversely, when $\eta = 0$, it becomes a process characteristic of DDIM.

The denoising diffusion model, such as DDIM, is highly flexible. It can simulate complex probability distributions and dynamically adjust its structure during training to better fit the data. This flexibility gives it broad application prospects for generating complex data, such as high-quality images and audio.

3.3. Message Passing Neural Network (MPNN)

Message Passing Neural Network (MPNN) is a commonly used graph neural network framework [29]. The graph neural network used to train an MPNN includes the node features, denoted as x_v, and the edge features, denoted as e_{vw}. MPNN consists of two stages: the message-passing stage and the readout stage. During the message-passing stage, information is generated based on the node features and transmitted according to the network's topology. The message passing function and update function are shown in Eqs. (9) and (10), respectively.

$$\boldsymbol{m}_v^{t+1} = \sum_{\omega\in N_v} M\left(\boldsymbol{h}_v^t, \boldsymbol{h}_w^t, \boldsymbol{e}_{vw}\right) \tag{9}$$

$$\boldsymbol{h}_v^{t+1} = U\left(\boldsymbol{h}_v^t, \boldsymbol{m}_w^{t+1}\right) \tag{10}$$

where $\boldsymbol{m}_v^t$ is the information generated at the time step t, $\boldsymbol{h}_v^t$ is the hidden state of nodes, $N(v)$ represents all the neighbors of the node v, M_t is the message function that generates information based on nodes and edges, and U_t is the aggregate function that combines the information passed to the nodes with the nodes' own features to

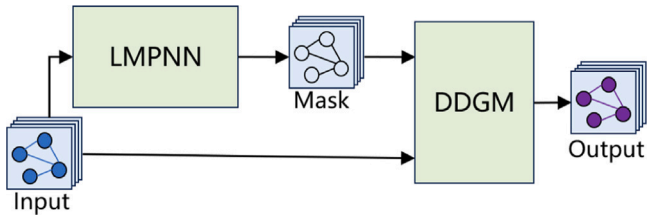

Fig. 1. The data recovery framework.

generate new node features. In the readout stage, this information is mapped to a feature vector describing the graph features based on the entire graph's characteristics, as shown in the following equation.

$$\hat{\boldsymbol{y}} = R\left(\{\boldsymbol{h}_v^T | v \in G\}\right) \tag{11}$$

where R is the readout function that aggregates all node states. MPNN can effectively learn the spatial features of large networks with a lot of data by updating the feature representation of nodes by passing messages on the graph.

4. Data recovery framework against FDIA

In this section, a framework for data recovery under FDIAs is proposed, as illustrated in Fig. 1. The framework synergistically incorporates two pivotal models: the LMPNN-based FDIA localization model, as depicted in Fig. 3, and the DDGM-based data recovery model , as depicted in Fig. 4. The FDIA localization model incorporates LMPNN and the hidden space, primarily aimed at accurately distinguishing between compromised and normal measurements to generate a mask. This mask, as shown in Fig. 1, is subsequently fed into DDGM together with the measurements. The DDGM, constituting a two-stage circular structure, embodies innovation by integrating elements of DDIM, LMPNN, a hidden space, and physical constraints. And it can be applied to data recovery under FDIA effectively.

4.1. Node graph, line graph and LMPNN

In smart grids, the connection of various interconnected power facilities (e.g., loads, generators and distribution substation) can be abstracted to corresponding network topologies. Each grid bus and branch can be represented as a node and an edge, respectively, while the measurements on the buses and branches can be abstracted as node and edge features, respectively. Thus, an N-bus power system is modeled as an undirected node graph, noted as $G = (X(\boldsymbol{x}), E(\boldsymbol{e})) \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$, where the bus (node) set $X(\boldsymbol{x}) \in \mathbb{R}^{4\times N\times 1}$ is constituted of the active and reactive power injections on buses, voltage volumes, and phase angles of buses, and the branch (edge) set $E(\boldsymbol{e}) \in \mathbb{R}^{4\times N\times N}$ is comprised of the adjacency matrix of the active and reactive power flows injected into and out of branches. Based on the node graph structure, the bus data associated with nodes of graphs can be processed.

The hidden space $H(\boldsymbol{h}) \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$ is initialized with Gaussian white noise, where the amplitude distribution follows the Gaussian distribution and the power spectral density is characterized by uniformly distributed noise.

The LMPNN is designed to extract the data features and identify data relational anomalies of the measurements. Before the graph convolution operation, the network topology information and hidden space information are aggregated to enhance the model's capacity to discern and represent the underlying data relationships accurately. The operation is shown in Eq. (12)–(15),

$$N_s\left(\boldsymbol{x}_v^s, \boldsymbol{h}_{xv}^s\right) = \left\langle \boldsymbol{x}_v^s, \boldsymbol{h}_{xv}^s \right\rangle \tag{12}$$

$$\boldsymbol{n}_w^{s+1} = \sum_{v\in N(w)} conv N_s\left(\boldsymbol{x}_v^s, \boldsymbol{h}_{xv}^s\right) \tag{13}$$

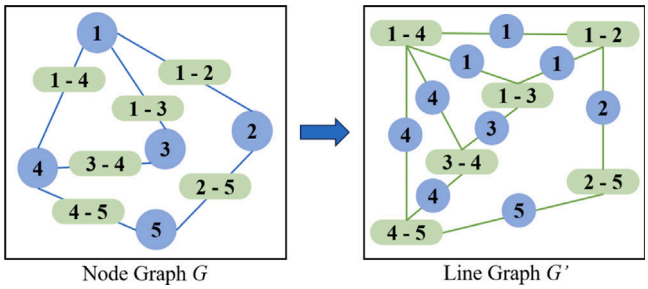

Fig. 2. The process of generating a line graph from a node graph.

$$A_s\left(\boldsymbol{e}_v^s, \boldsymbol{h}_{ev}^s\right) = \left\langle \boldsymbol{e}_v^s, \boldsymbol{h}_{ev}^s \right\rangle \tag{14}$$

$$\boldsymbol{a}_w^{s+1} = \sum_{v\in N(w)} conv A_s\left(\boldsymbol{e}_v^s, \boldsymbol{h}_{ev}^s\right) \tag{15}$$

where $\boldsymbol{x}_v^s$ represents the updated information of node v after the time step s, $\boldsymbol{h}_{xv}^s$ denotes the corresponding information of the hidden space for node v after the same time step, $\boldsymbol{e}_v^s$ indicates the updated information of edge v after the time step s, $\boldsymbol{h}_{ev}^s$ the corresponding information of the hidden space for edge v after the same time step, $conv$ indicates the operation of convolution, and $\langle\cdot\rangle$ represents concatenating node or edge information with corresponding hidden space information. The aggregated node and edge information undergoes further refinement through CNN. Subsequently, the updated node set $N_s(\boldsymbol{n}^s)$ and edge adjacency matrix set $A_s(\boldsymbol{a}^s)$ are obtained. Next, the node information undergoes refinement through GCN, which aggregates the features of adjacent nodes and edges. It is operated as follows,

$$\boldsymbol{n}_i^{s+1} = MultiHead\left(\left\langle \sum_j \widetilde{\boldsymbol{a}}_{1ij}^s \boldsymbol{n}_{fj}^s, \sum_j \widetilde{\boldsymbol{a}}_{2ij}^s \boldsymbol{n}_{fj}^s, \sum_j \widetilde{\boldsymbol{a}}_{3ij}^s \boldsymbol{n}_{fj}^s, \sum_j \widetilde{\boldsymbol{a}}_{4ij}^s \boldsymbol{n}_{fj}^s \right\rangle\right) \tag{16}$$

where the corner mark f represents the measurement type, and $\widetilde{\boldsymbol{a}}_{fij}$ represents the data of the f measurement type of the branch connecting bus i and j after standardization. Four types of branch data (active and reactive power flows injected into and out of transmission lines) are sequentially fused with bus data before being concatenated. Multi-Head Attention [30] is incorporated following GCN. By assigning different weights to information in different locations, the network is adept at processing data from diverse perspectives, thereby enhancing its capability to extract data relationship information. Through these methodologies, the updated bus measurements are achieved.

The branch feature updating process is carried out after the bus features are updated. To initiate the update of branch features, a line graph is constructed, wherein each grid branch and bus are abstracted as a node and an edge, respectively, while the measurements associated with branches and buses are represented as node and edge data correspondingly. This line graph $G' = L(G) = \left(E'\left(\boldsymbol{e}'\right), X'\left(\boldsymbol{x}'\right)\right)$ functions as a dual representation of G, which represents the adjacency of the buses. To construct G', the edges in G are converted to the nodes in G', while the nodes in G are converted to the edges in G'. Namely, $E'\left(\boldsymbol{e}'\right) = \{\boldsymbol{x}(e) | e \in E\}$. H takes a similar operation, and gets $H'\left(\boldsymbol{h}'\right) = L(H)$. Fig. 2 is an example of generating a line graph from an original node graph.

The following equations demonstrate the aggregation of topology information and hidden space information within the line graph.

$$A'_s\left(\boldsymbol{e}_v'^s, \boldsymbol{h}_{e'v}'^s\right) = \left\langle \boldsymbol{e}_v'^s, \boldsymbol{h}_{e'v}'^s \right\rangle \tag{17}$$

$$\boldsymbol{a}_w'^{s+1} = \sum_{v\in N(w)} conv A_s\left(\boldsymbol{e}_v'^s, \boldsymbol{h}_{e'v}'^s\right) \tag{18}$$

$$N'_s\left(\boldsymbol{x}_v'^s, \boldsymbol{h}_{x'v}'^s\right) = \left\langle \boldsymbol{x}_v'^s, \boldsymbol{h}_{x'v}'^s \right\rangle \tag{19}$$

$$\boldsymbol{n}'^{s+1}_{w} = \sum_{v \in N'(w)} convN_s\left(\boldsymbol{x}'^{s}_{v}, \boldsymbol{h}'^{s}_{x'v}\right) \tag{20}$$

The update operation for the branch features is defined as follows.

$$\boldsymbol{a}'^{s+1}_{i} = MultiHead\left(\left\langle \sum_j \widetilde{\boldsymbol{n}}'^{s+1}_{1ij}\, \boldsymbol{a}'^{s}_{fj}, \sum_j \widetilde{\boldsymbol{n}}'^{s+1}_{2ij}\, \boldsymbol{a}'^{s}_{fj}, \sum_j \widetilde{\boldsymbol{n}}'^{s+1}_{3ij}\, \boldsymbol{a}'^{s}_{fj}, \sum_j \widetilde{\boldsymbol{n}}'^{s+1}_{4ij}\, \boldsymbol{a}'^{s}_{fj} \right\rangle\right) \tag{21}$$

Then, convert the updated $\boldsymbol{a}'$ to $\boldsymbol{a}$, that is, transfer the line graph back to node topology graph: $G = (N(\boldsymbol{n}), A(\boldsymbol{a})) = L^{-1}(G')$. Through the aforementioned methods, the update of branch data is achieved.

The LMPNN module now completes the aggregation, transfer, and update of the bus and branch information by learning bus and branch embeddings. In the readout stage, the hidden space information undergoes updating, preservation, and subsequent utilization for the subsequent training step, as illustrated below.

$$R\left(\boldsymbol{h}^{s+1}_{xw}\right) = \sum conv\left\langle \boldsymbol{n}^{s+1}_{w}, \boldsymbol{h}^{s}_{xw} \right\rangle \tag{22}$$

$$R\left(\boldsymbol{h}^{s+1}_{ew}\right) = \sum conv\left\langle \boldsymbol{a}^{s+1}_{w}, \boldsymbol{h}^{s}_{ew} \right\rangle \tag{23}$$

The classical MPNN framework primarily focuses on updating node information, neglecting edge information. This limitation hinders its ability to extract data relational features and aggregate information. In contrast, the proposed LMPNN allocates equal emphasis to both node and edge information, thus facilitating the efficient aggregation of bus measurements, branch measurements, and hidden space information. This comprehensive approach ensures a more precise alignment with data relationships, consequently yielding superior accuracy in the results.

4.2. Physical principles learning methodology

In this work, the compatibility of the recovered data with the physical laws governing the power grid by embedding the underlying physical relationships between power system measurements is assessed into the model's loss function. This approach penalizes solutions that deviate from these physical laws, ensuring accurate reconstruction of the input data. The physics-based loss terms serve as regularization agents, thus eliminating the need for traditional regularization techniques. The loss function is modeled through the following equation

$$L = \lambda_{phy} L_{phy} + \lambda_{data} L_{data} \tag{24}$$

where L_{phy} denotes the physical error term, L_{data} denotes the statistical error term, and the hyperparameters λ_{phy} and λ_{data} weigh the importance of the two error terms in the loss function. By selecting appropriate values for these hyperparameters, the model can balance fitting the data accurately while constraining the predicted physical quantities within the range allowed by the laws of physics.

In power flow calculations, the bus power balance equation is utilized to articulate the power balance relationship of each bus in the power system, as shown in Eq. (25)–(26).

$$P_i - P_{Gi} + P_{Li} = 0 \tag{25}$$

$$Q_i - Q_{Gi} + Q_{Li} = 0 \tag{26}$$

where P_i and Q_i respectively represent the active and reactive power injection of bus i, P_{Gi} and Q_{Gi} respectively represent the active and reactive power injection of the generator connected to bus i, and P_{Li} and Q_{Li} respectively represent the active and reactive power injection of the load connected to bus i.

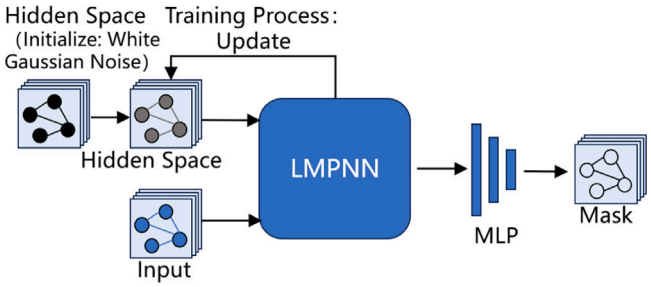

Fig. 3. The FDIA localization model for generating masks.

The branch power balance equation is used to describe the power balance relationship of each branch in the power system, as shown in Eq. (27)–(28).

$$P_{ij} + P_{ji} = 0 \tag{27}$$

$$Q_{ij} + Q_{ji} = 0 \tag{28}$$

where P_{ij} and Q_{ij} respectively represent the active and reactive power flow from bus i to bus j, and P_{ji} and Q_{ji} respectively represent the active and reactive power flow from bus j to bus i.

According to the power balance equation, in an ideal state without noise, the physical constraint satisfies

$$\sum \left(P_i - P_{Gi} + P_{Li} + P_{ij} + P_{ji} + Q_i - Q_{Gi} + Q_{Li} + Q_{ij} + Q_{ji}\right) = 0 \tag{29}$$

So the physical error term L_{phy} is defined as

$$L_{phy} = \sum \left(P_i - P_{Gi} + P_{Li} + P_{ij} + P_{ji} + Q_i - Q_{Gi} + Q_{Li} + Q_{ij} + Q_{ji}\right) \tag{30}$$

L_{phy} force the reconstructed measurements to follow Kirchhoff's law. The smaller the L_{phy}, the more consistent the recovered data is with the physical laws of the power system.

4.3. LMPNN-based FDIA localization model

Under normal circumstances, an attacker targets only part of the nodes, resulting in the compromise of only some measurements whose specific locations remain unknown. The proposed FDIA localization model is designed to detect and precisely locate the data affected by FDIA at any given time.

As shown in Fig. 3, the FDIA localization model is comprised of a LMPNN, a hidden space and a Multilayer Perceptron (MLP). The LMPNN, as shown in Section 4.1, is utilized to extract features from measurements in node and edge space. The hidden space $H \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$ is initially populated with Gaussian white noise and subsequently updated and recorded in a Parameter List of the model for use in the following training step. Consequently, the initial hidden space information for each new training step comprises data from the previous step rather than starting anew with Gaussian white noise. Since only a small portion of data in each sample is attacked, and the attacked data varies, continuous training iterations cause the hidden space data to increasingly approximate the LMPNN output with correct input data. This iterative process facilitates the accuracy and convergence of the model. During testing, the hidden space information is set to the preserved hidden space data from the last training iteration and is not updated. Therefore, incorporating the hidden space, which contains features and data correlations of correct data, into the model improves its ability to locate attacked measurements.

Finally, the extracted features are fed into an MLP consisting of three fully connected layers. The output is then activated by the activation function. The FDIA localization result is presented as

$$s = \underbrace{[1, 1, 0, 1, \ldots, 1, 0, 1, 1, 1]}_{length=m} \tag{31}$$

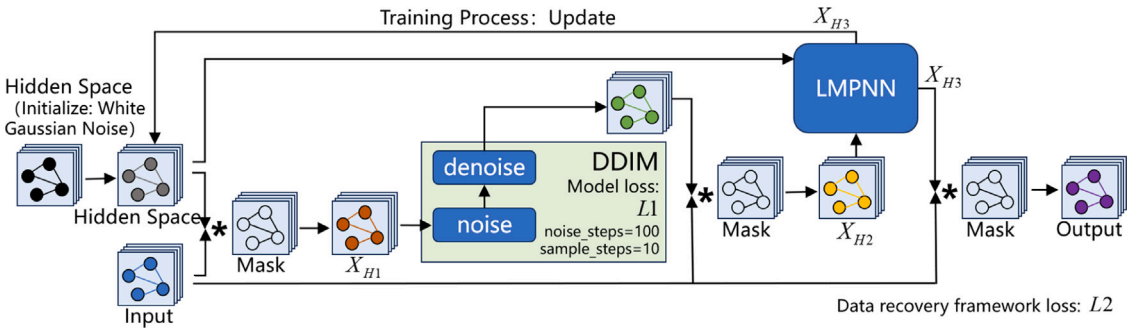

Fig. 4. Denoising diffusion graph circulation models.

where numerical output of "1" denotes no attack, while "0" signifies the occurrence of an attack under FDIA. Here, m represents the number of measurements within a given sampling time instance, with each output value corresponding to a specific measurement. Finally, convert $s \in \{0,1\}^{1\times m}$ to the topology structure $M \in \{0,1\}^{4\times N\times 1+4\times N\times N}$ corresponding to the network topology, and let M be the mask of DDGM, as shown in Fig. 4.

4.4. DDGM

The framework of DDGM is shown in Fig. 4. Training commences by initializing the hidden space $H \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$ with Gaussian white noise. Subsequently, the input data is replaced by the hidden space information H in accordance with the mask information M, which is defined as

$$X_{H1i,j,k} = \begin{cases} X_{i,j,k} & \text{if } M_{i,j,k} = 1 \\ H_{i,j,k} & \text{if } M_{i,j,k} = 0 \end{cases} \tag{32}$$

The process involves feeding the hidden space $X_{H1} \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$ into DDIM to execute the data noising and denoising phases. Specifically, the noising step is set to 100, while the denoising step is set to 10. To evaluate the performance of DDIM and facilitate training, a loss function $L1$ is employed, allowing for the relatively independent optimization of the DDIM component. The objective function of DDIM can be directly defined as that of DDPM [24], as shown in Eq. (33),

$$\begin{aligned} L1 = E_{x_{0:T}\sim q_{\sigma,\tau}(x_{0:T})}\Bigg[\sum_{t\in\bar{\tau}} D_{\mathrm{KL}}\left(q_{\sigma,\tau}\left(x_t \mid x_0\right) \parallel p_\theta^{(t)}\left(x_0 \mid x_t\right)\right) \\ + \sum_{i=1}^{L} D_{\mathrm{KL}}\left(q_{\sigma,\tau}\left(x_{\tau_{i-1}} \mid x_{\tau_i}, x_0\right) \mid p_\theta^{(\tau_i)}\left(x_{\tau_{i-1}} \mid x_{\tau_i}\right)\right)\Bigg)\Bigg] \end{aligned} \tag{33}$$

where each KL divergence is between two Gaussians with variance independent of θ. The introduction of $L1$ into the model accelerates the convergence speed of both the DDIM component and the overall model. Adhering to a similar principle as outlined in Eq. (32), the original input data is substituted with data generated by DDIM in accordance with the mask information M, yielding $X_{H2} \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$.

Input X_{H2} into LMPNN, and get $X_{H3} \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$. The LMPNN structure in this part is slightly different from that in Section 4.1, as shown in Eq. (34)–(41).

$$N_s\left(\boldsymbol{x}_v^s, \boldsymbol{m}_{xv}^s, \boldsymbol{h}_{xv}^s\right) = \left\langle \boldsymbol{x}_v^s, \boldsymbol{m}_{xv}^s, \boldsymbol{h}_{xv}^s \right\rangle \tag{34}$$

$$\boldsymbol{n}_w^{s+1} = \sum_{v\in N(w)} convN_s\left(\boldsymbol{x}_v^s, \boldsymbol{m}_{xv}^s, \boldsymbol{h}_{xv}^s\right) \tag{35}$$

$$A_s\left(\boldsymbol{e}_v^s, \boldsymbol{m}_{ev}^s, \boldsymbol{h}_{ev}^s\right) = \left\langle \boldsymbol{e}_v^s, \boldsymbol{m}_{ev}^s, \boldsymbol{h}_{ev}^s \right\rangle \tag{36}$$

$$\boldsymbol{a}_w^{s+1} = \sum_{v\in N(w)} convA_s\left(\boldsymbol{e}_v^s, \boldsymbol{m}_{ev}^s, \boldsymbol{h}_{ev}^s\right) \tag{37}$$

$$A'_s\left(\boldsymbol{e}_v'^s, \boldsymbol{m}_{e'v}'^s, \boldsymbol{h}_{e'v}'^s\right) = \left\langle \boldsymbol{e}_v'^s, \boldsymbol{m}_{e'v}'^s, \boldsymbol{h}_{e'v}'^s \right\rangle \tag{38}$$

$$\boldsymbol{a}_w'^{s+1} = \sum_{v\in N(w)} convA_s\left(\boldsymbol{e}_v'^s, \boldsymbol{m}_{e'v}'^s, \boldsymbol{h}_{e'v}'^s\right) \tag{39}$$

$$N'_s\left(\boldsymbol{x}_v'^s, \boldsymbol{m}_{x'v}'^s, \boldsymbol{h}_{x'v}'^s\right) = \left\langle \boldsymbol{x}_v'^s, \boldsymbol{m}_{x'v}'^s, \boldsymbol{h}_{x'v}'^s \right\rangle \tag{40}$$

$$\boldsymbol{n}_w'^{s+1} = \sum_{v\in N'(w)} convN_s\left(\boldsymbol{x}_v'^s, \boldsymbol{m}_{x'v}'^s, \boldsymbol{h}_{x'v}'^s\right) \tag{41}$$

where $\boldsymbol{m}_{xv}$ is the bus mask, $\boldsymbol{m}_{ev}$ is the branch mask, $\boldsymbol{m}'_{x'v}$ and $\boldsymbol{m}'_{e'v}$ are the masks converted to line graph: $M'\left(m'\right) = L\left(M\right)$. Splicing masks to the input facilitates LMPNN in finding data connections more effectively based on correct data.

Following the same principle as Eq. (32), replace the initially input data with X_{H3} based on the mask information M, and then get the result $R \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$ of this training step. Similarly to the FDIA localization model, throughout the training phase, the hidden space information is updated with X_{H3}, and recorded in the Parameter List of the model following each training iteration, while it remains static during testing.

Let the correct data of training samples prior to the attack, denoted as $C \in \mathbb{R}^{4\times N\times 1+4\times N\times N}$, serve as the training label. Based on physical principles learning method in Section 4.2 and Eq. (24), the whole loss of DDGM is defined as

$$L2 = \lambda_{phy} L_{phy} + \lambda_{data} \cdot \sum (R - C)^2 \tag{42}$$

Based on trial and error, we have found that the best choice for the hyperparameters are $\lambda_{phy} = 10^{-4}$ and $\lambda_{data} = 1$. The values effectively balance the trade-off between fitting the data accurately and maintaining adherence to the physical laws.

5. Experiments and discussions

In this section, numerical experiments are conducted on the IEEE 39-bus and 118-bus test systems to validate the effectiveness of the FDIA localization model and DDGM. The superiority of the proposed models over existing approaches is also discussed.

5.1. Experimental setup

The experiments are performed on a high-performance server with two Intel Xeon Gold 6248R 3.00 GHz CPUs, 128 GB RAM, and two NVIDIA RTX 3090 GPU. The operating system is Ubuntu Linux 22.04. The deep learning models are implemented based on the Pytorch 1.8.1 library with CUDA Toolkit 11.1 installed.

For the FDIA detection experiment on the IEEE 39-bus and 118-bus systems, the experimental setup includes two training epochs and 6000 training steps. The dataset comprises 48,000 training samples, 6000 validation samples, and 6000 test samples. For the data recovery experiment on the two same systems, the setup includes three training epochs and 13,500 training steps, with 72,000 training samples, 9000 validation samples, and 9000 test samples. In both experiments, each

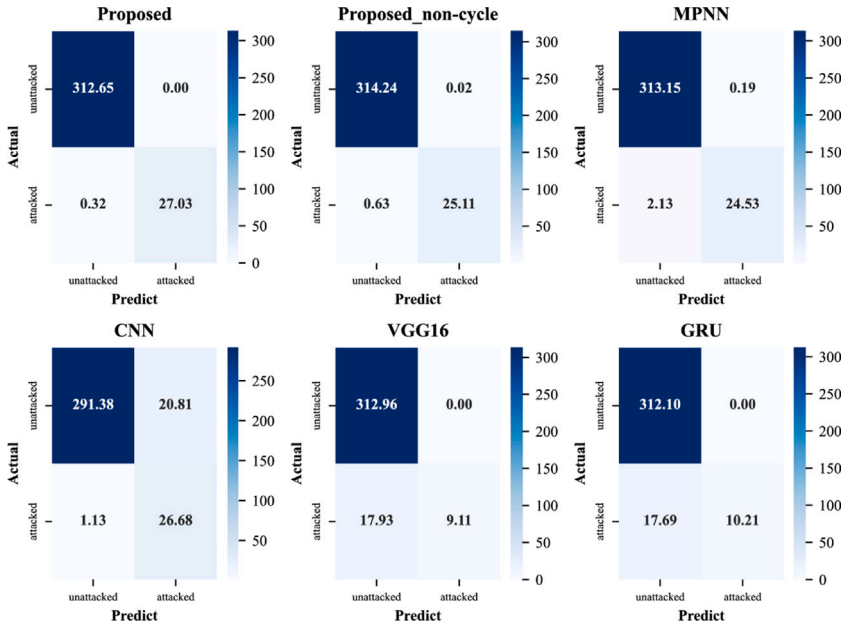

Fig. 5. The confusion matrices for tests on the IEEE 39-bus.

sample contains measurements under FDIA at a single sampling time instance, with the amplitude of the attack vector c randomly selected, ranging from 0.0002 to 0.2 after scaling. To ensure variability, FDIA is applied to the magnitude or phase angle of 1–2 arbitrary nodes in the system (typically only one of the two values is tampered with, while in rare cases, both magnitude or phase angle may be altered simultaneously), tampering with the corresponding measurements, including the nodal active and reactive power injections, voltage volumes, and phase angles of branches, and the active and reactive power flows injected into the transmission lines. To assess the model's robustness under power fluctuation, simulation cases are conducted with loading conditions randomly selected within 50%–150% of the base load, generation conditions within 50%–150% of the base generator level, and measurement noise is set to a maximum of 5% of the raw value. All data samples are generated through simulations using MATPOWER.

5.2. Accuracy of FDIA localization model

The accuracy is used as the metric to evaluate the performance of the proposed FDIA detection model. It reflects the proportion of correct detections out of the total measurements, which is defined as

$$Accuracy = \frac{TP_{Avg} + TN_{Avg}}{TOT_{Per}} \tag{43}$$

where TP_{Avg}, TN_{Avg}, and TOT_{Per} denote the number of measurements that correctly detected attacks per sample on average, the number of measurements that correctly detected no attacks per sample on average, and the total number of measurements per sample, respectively.

The accuracy of the proposed model and the mainstream detection methods on the IEEE 39-bus and 118-bus systems are shown in Table 1. In these tables, $Accuracy$ denotes the average classification accuracy across 6000 test samples, while VAR indicates the variance in accuracy across these samples. The confusion matrices for tests on the IEEE 39-bus and 118-bus systems, reflecting misclassifications, are shown in Figs. 5 and 6. The values in positions (1,1), (1,2), (2,1), and (2,2) of these matrices correspond to the following: the average number of correctly classified non-attacked measurements per sample, the average number of non-attacked measurements misclassified as attacked, the average number of attacked measurements misclassified as non-attacked, and the average number of correctly classified attacked measurements, respectively.

Table 1
The Test Results of Different Models for FDIA Localization.

Model	IEEE 39-bus system		IEEE 118-bus system	
	Accuracy	VAR($\times 10^{-6}$)	Accuracy	VAR($\times 10^{-6}$)
Proposed	99.91%	1.20	99.87%	1.34
Proposed_non-cycle	99.81%	2.11	99.70%	3.05
MPNN	99.32%	7.15	99.48%	7.98
CNN	93.55%	64.72	93.48%	58.96
VGG16	94.73%	48.85	97.72%	30.20
GRU	94.80%	92.91	97.97%	64.30

In Table 1, Figs. 5 and 6, "Proposed_non-cycle" indicates the deletion of hidden space information updating process in FDIA detection model. VGG16 [31] is a deep neural network composed of 16 cascading layers, organized into five convolutional groups followed by an MLP. It is designed to extract deeper features from the data and theoretically performs better than standard CNNs.

Table 1 demonstrates that the proposed FDIA localization model achieves very good performance on both the IEEE 39-bus and 118-bus systems, with an accuracy exceeding 99.8%, and a smaller variance. It can be concluded that Graph Neural Networks (GNNs) significantly outperform commonly used methods such as Convolutional Neural Networks (CNNs, including VGG16) and Sequential Neural Networks (GRU) in this context. This superiority is due to the intricate linkage between the attribute information of power system measurements and their corresponding topological information. GNNs facilitate the synchronous learning of both structural and attribute information, enabling accurate identification of attacked measurements based on their topological relationships. This capability remains robust even when power fluctuations cause substantial changes in the overall measurements.

To further validate the superiority of the proposed model, a t-test was performed to compare the Loop Message Passing Neural Network (LMPNN) with two alternatives: LMPNN without the loop structure (i.e., without hidden space information updates) and the classical Message Passing Neural Network (MPNN). The results, shown in Table 2, reveal that all test statistics are greater than zero, indicating that LMPNN consistently achieves higher mean accuracy than both LMPNN

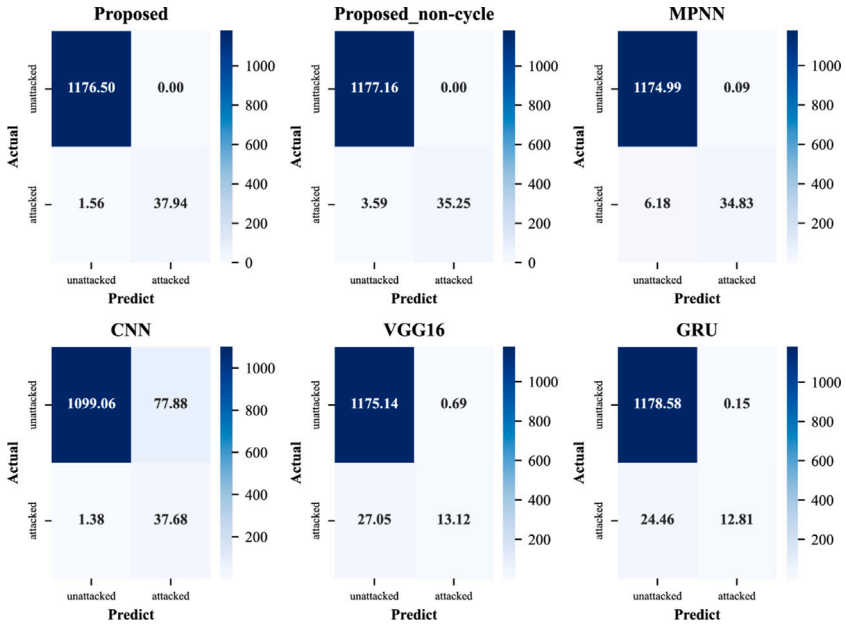

Fig. 6. The confusion matrices for tests on the IEEE 118-bus.

Table 2
The T-Test Results for FDIA Localization.

	IEEE 39-bus system		IEEE 118-bus system	
	Statistic	*P*-value	Statistic	*P*-value
Proposed - Proposed_non-cycle	2.73	6.36×10^{3}	3.24	1.02×10^{-3}
Proposed - MPNN	21.81	5.77×10^{-82}	29.15	3.35×10^{-80}

without the loop structure and MPNN. Additionally, the P-values are far below the 0.1 threshold, leading to the rejection of the null hypothesis of equal means, thus confirming that the mean accuracy of LMPNN is significantly better. These results highlight the effectiveness of the loop structure and underscore LMPNN's ability to concurrently extract and integrate relational information from both the edge and node spaces.

In summary, the proposed model introduces an innovative approach by leveraging LMPNN to effectively capture the complex data relationships within power systems and enhances FDIA localization accuracy under conditions of power fluctuation and SCADA noise. This demonstrates the model's superior capability in handling real-world challenges in power system monitoring and protection.

5.3. Performance of DDGM

Ablation studies and contrast experiments are conducted to demonstrate the validity of the proposed model.

5.3.1. Ablation study

The ablation study's training loss changes for the first three epochs on the IEEE 39-bus and 118-bus systems are depicted in Fig. 7. The corresponding test results are presented in Tables 3 and 4. The loss is calculated using Eq. (42), where a value closer to 0 indicates a data recovery result closer to the label, or, the correct data prior to the attack.

The training results depicted in Fig. 7 reveal that DDGM demonstrates a commendable convergence rate and achieves a low asymptotic loss on both the IEEE 39-bus and 118-bus systems. The variant "Proposed non-cycle" denotes the removal of the hidden space information updating process from DDGM. The loss of "Proposed non-cycle" is slightly higher than that of DDGM, validating the effectiveness of the hidden space circulation structure. In contrast, MPNN exhibits a considerably larger loss than DDGM, with a slower convergence speed. This discrepancy is attributable to MPNN's inherent limitations in generating diverse, high-quality recovery data samples and its inability to address power flows on branches (edges) within the smart grid. Similarly, DDIM also displays a significantly higher loss and slower convergence speed than DDGM, mainly due to its inability to ensure that the recovered data samples accurately reflect the current power flow state. Thus, the rationality of the model structure is demonstrated.

5.3.2. Contrast experiments

The convergence stability of FDIA data recovery for different models is compared, and the training loss changes for the first three epochs on the IEEE 39-bus and 118-bus systems are shown in Fig. 8. The loss is calculated using Eq. (42). In this section, the paper compares advanced methods that have demonstrated strong performance in data imputation scenarios. The models include SSSD [27], a newly proposed neural network based on diffusion models, and Wasserstein GAN with Gradient Penalty (WGAN-GP) [32], which is based on GAN and Wasserstein Distance. Deep Convolutional Generative Adversarial Networks (DCGAN) [33], combining CNNs within a GAN framework, also exhibit superior stability and generation quality over classical GANs, and it is also compared. Furthermore, the paper evaluates the Graph Recurrent Imputation Network (GRIN) [34], a recent model employing graph neural networks for multivariate time series imputation. GP-VAE [35], which merges Deep Variational AutoEncoders (VAEs) with Gaussian process (GP) to model low-dimensional dynamics, and ConvGRU, which combines the temporal modeling capabilities of GRU with CNNs to effectively extract data correlation features, are also included in the comparison.

From Fig. 8, it is evident that the proposed DDGM demonstrates superior performance in both the IEEE 39-bus and 118-bus systems. Specifically, DDGM exhibits a faster convergence speed and significantly lower asymptotic loss compared to other models. Although models like SSSD and GP-VAE achieve relatively satisfactory results, they still fall short when compared to DDGM. In contrast, commonly used data recovery models such as WGAN-GP and DCGAN exhibit significantly lower training stability, characterized by apparent oscillations. Furthermore, models such as GRIN, ConvGRU, VGG, and CNN

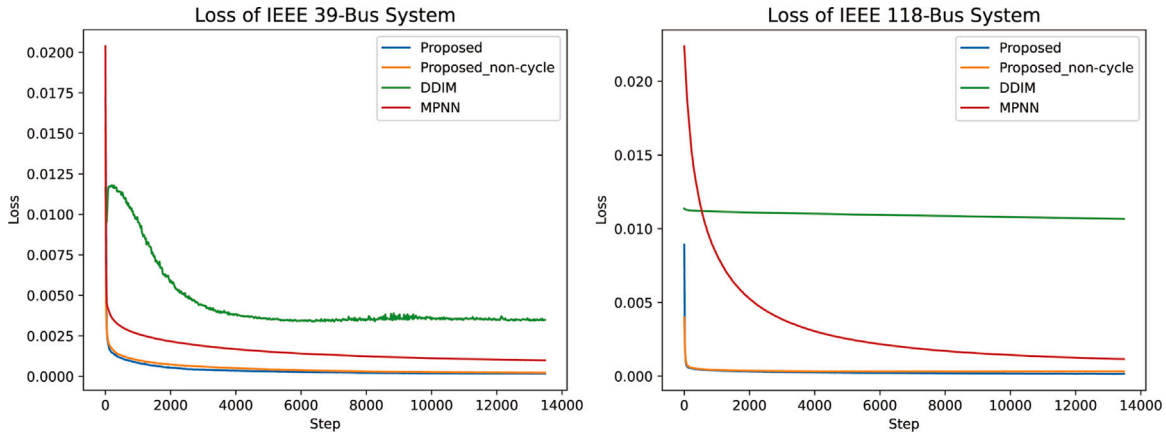

Fig. 7. Training convergence curves of ablation study.

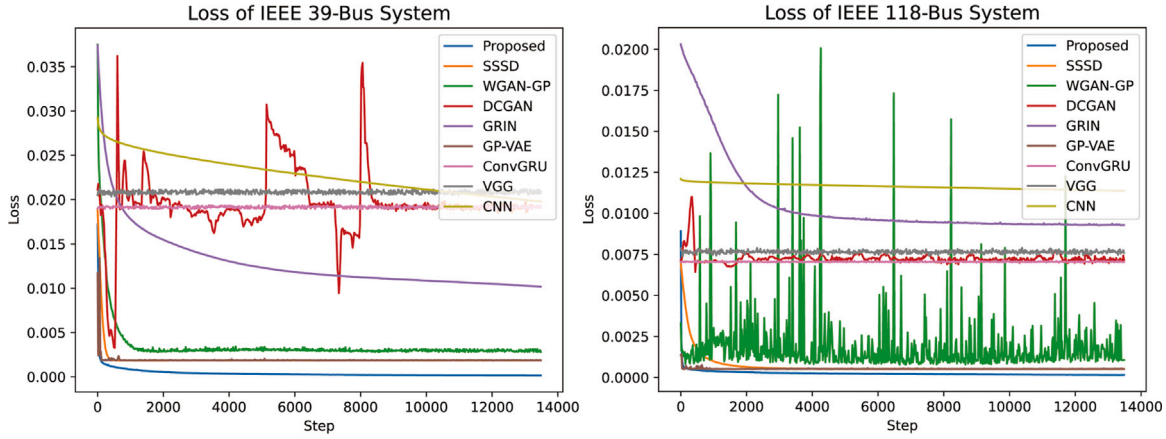

Fig. 8. Training convergence curves of different data recovery models.

exhibit much higher asymptotic losses and either fail to converge or display extremely slower convergence speeds.

Fig. 9 and Fig. 10 respectively show the heatmaps of partial measurements (including active power, reactive power, voltage, and phase angle of the nodes) at a certain moment for the IEEE 39-bus and IEEE 118-bus systems. This includes the normalized data before the attack, after the attack, and after recovery. It is evident from the figures that the heatmap of the recovered data is very close to that of the un-attacked data.

Tables 3 and 4 are the test results of different models on IEEE 39-bus and 118-bus systems, respectively. The total loss (calculated using Eq. (42)), the physical error term (calculated using Eq. (30)), and the variance of the loss for the 9000 test samples are presented. The data demonstrates that DDGM achieves the lowest asymptotic loss among advanced models, indicating superior data recovery accuracy. Therefore, the FDIA data recovery based on DDGM outperforms other advanced models, demonstrating the advantage of integrating DDIM and GNN for this application scenario. The integration ensures robust and reliable recovery of data compromised by FDIA, even under varying power flow conditions. Similar conclusions are evident from the tests on the 118-bus system. Thus, the contrast experiments confirm the proposed model's enhanced data recovery capabilities and its suitability for dynamic power grid environments.

To assess the compatibility of the recovered data with the physical laws governing the power grid, Section 4.2 defines physical constraint loss. Due to the addition of noise as a measurement error, the physical constraint loss for the samples is inherently greater than 0. According to Eq. (30), the average physical constraint loss for non-attack test samples on IEEE 39-bus and 118-bus systems is 3.2328 and 1.2806, respectively. Notably, the physical constraint loss for the DDGM is lower than that for other models, and is close to that for non-attack test samples, indicating that the recovered data better conforms to the data relationships and physical laws of the current state.

Table 3
The Test Results of Different Models on IEEE 39-bus System for Data recovery.

Model	Loss($\times 10^{-3}$)	Physical constraint loss	VAR($\times 10^{-9}$)
Proposed	0.158	4.8835	3.42
Proposed_non-cycle	0.264	4.8231	8.17
DDIM	3.379	11.5858	213.49
MPNN	0.989	5.5720	41.02
SSSD	1.824	7.9782	96.58
WGAN-GP	2.794	8.3598	196.97
DCGAN	3.261	27.8929	283.45
GRIN	10.178	9.4811	1216.56
GP-VAE	1.850	7.5975	113.59
ConvGRU	18.941	19.5167	8696.57
VGG16	20.257	22.5465	12163.55
CNN	19.775	18.0829	10385.56

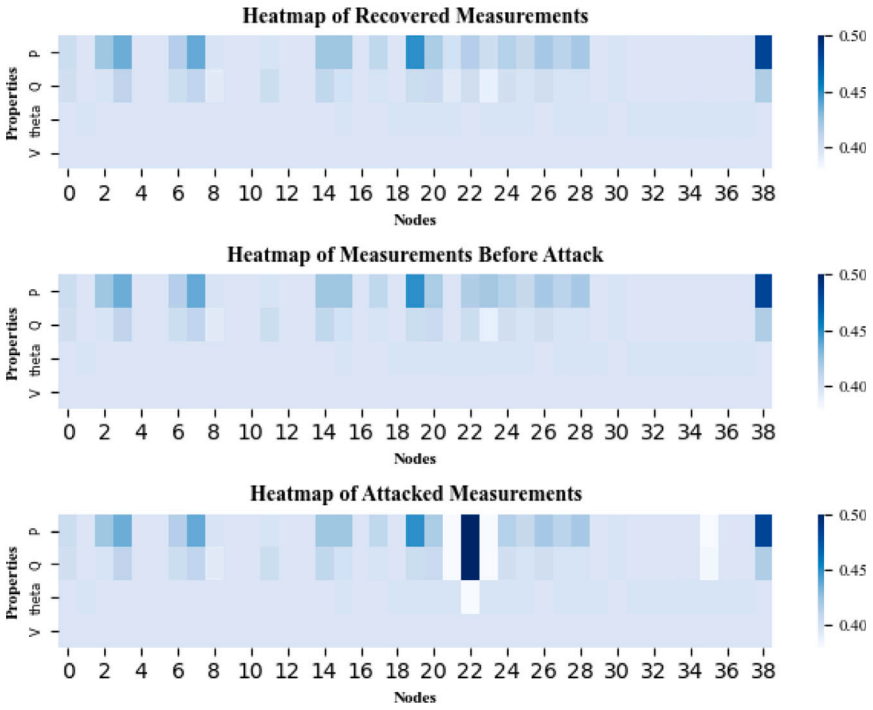

Fig. 9. Heatmap of part of the measurements on the IEEE 118-bus.

Fig. 10. Heatmap of part of the measurements on the IEEE 118-bus.

In addition, the test results indicate that the neural network algorithms are fast enough to meet the real-time requirements for data recovery, according to the test time shown in Tables 3 and 4.

In conclusion, DDGM exhibits the capability to capture the potential distribution of incoming data and the inherent dynamics of the smart grid. This adaptability enables it to contend with the uncertainty stemming from power fluctuations to some extent. It not only excels in data

Table 4
The Test Results of Different Models on IEEE 118-bus System for data recovery.

Model	Loss($\times 10^{-3}$)	Physical constraint loss	VAR($\times 10^{-9}$)
Proposed	0.143	1.8553	2.78
Proposed_non-cycle	0.325	2.4778	9.92
DDIM	10.669	11.7044	899.96
MPNN	1.241	7.6995	59.10
SSSD	0.516	6.0163	21.32
WGAN-GP	0.770	6.8616	36.09
DCGAN	6.432	11.5473	746.98
GRIN	9.279	18.1233	1629.32
GP-VAE	0.516	5.9710	18.82
ConvGRU	6.995	11.2753	892.14
VGG16	7.522	11.9913	1105.32
CNN	11.411	11.5473	3840.13

recovery accuracy but also ensures that the recovered data adheres to the physical constraints of the power grid, reinforcing its applicability and effectiveness in real-world scenarios.

6. Conclusion

This paper proposes an FDIA data recovery framework comprising two key models: the FDIA localization model and the data recovery model. Within the FDIA localization model, an advanced graph neural network named LMPNN is designed to effectively detect and pinpoint FDIAs utilizing measurements obtained from both branches and buses. This advancement significantly enhances the accuracy of FDIA localization in complex power system environments. Additionally, the proposed recovery model employs a physics-informed two-stage circular structure, integrating elements of DDIM and LMPNN. This framework adeptly captures intricate data correlations, ensuring that the recovered data closely adheres to the underlying physical laws governing power grid operations. Case studies conducted on the IEEE 39-bus and 118-bus systems showcase the robustness of our proposed framework compared to alternative techniques. Notably, our framework demonstrates consistently high FDIA localization and data recovery performance even under uncertain high-level power fluctuations.

CRediT authorship contribution statement

Yixuan He: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Jingyu Wang:** Writing – review & editing, Supervision, Resources, Methodology, Formal analysis, Conceptualization. **Chen Yang:** Writing – review & editing, Validation. **Dongyuan Shi:** Writing – review & editing, Supervision, Resources.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jingyu Wang reports financial support was provided by National Natural Science Foundation of China. Jingyu Wang reports a relationship with National Natural Science Foundation of China that includes: funding grants. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China under Grant No. 52207107.

Data availability

Data will be made available on request.

References

[1] Y. Liu, P. Ning, M.K. Reiter, False data injection attacks against state estimation in electric power grids, ACM Transactions on Information and System Security 14 (1) (2011) 13.
[2] M. Jorjani, H. Seifi, A.Y. Varjani, H. Delkhosh, An optimization-based approach to recover the detected attacked grid variables after false data injection attack, IEEE Trans. Smart Grid 12 (6) (2021) 5322–5334, http://dx.doi.org/10.1109/TSG.2021.3103556.
[3] J. Ruan, G. Liang, J. Zhao, J. Qiu, Z.Y. Dong, An inertia-based data recovery scheme for false data injection attack, IEEE Trans. Ind. Inform. 18 (11) (2022) 7814–7823, http://dx.doi.org/10.1109/TII.2022.3146859.
[4] J. Pei, J. Wang, D. Shi, P. Wang, Detection and imputation based two-stage denoising diffusion power system measurement recovery under cyber-physical uncertainties, IEEE Trans. Smart Grid (2024) 1–15, early access.
[5] X. Hu, X. Wang, B. Shabbir, Y. Ma, Detection localization and recovery of false data injection attacks on power grids based on SA-DCNN and AE-LSTM, in: 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence, Vol. 3, ICIBA, 2023, pp. 1105–1109, http://dx.doi.org/10.1109/ICIBA56860.2023.10165322.
[6] D. Xue, X. Jing, H. Liu, Detection of false data injection attacks in smart grid utilizing ELM-based OCON framework, IEEE Access 7 (2019) 31762–31773, http://dx.doi.org/10.1109/ACCESS.2019.2902910.
[7] D. Hu, Y. Dong, J. Wang, D. Shi, Detection of false data injection attacks in smart grids under power fluctuation uncertainty based on deep learning, in: 2023 International Conference on Power System Technology, PowerCon, 2023, pp. 1–6, http://dx.doi.org/10.1109/PowerCon58120.2023.10331092.
[8] A.S. Musleh, G. Chen, Z.Y. Dong, A survey on the detection algorithms for false data injection attacks in smart grids, IEEE Trans. Smart Grid 11 (3) (2020) 2218–2234, http://dx.doi.org/10.1109/TSG.2019.2949998.
[9] H.T. Reda, A. Anwar, A. Mahmood, Comprehensive survey and taxonomies of false data injection attacks in smart grids: attack models, targets, and impacts, Renew. Sustain. Energy Rev. (Jul.) (2022) 163.
[10] X. Luo, M. Bai, X. Wang, X. Sun, Square-root extended Kalman filter-based detection of false data injection attack in smart grids, in: 2021 IEEE 5th Conference on Energy Internet and Energy System Integration, EI2, 2021, pp. 2376–2381, http://dx.doi.org/10.1109/EI252483.2021.9713070.
[11] Y. Zhao, X. Jia, D. An, Q. Yang, LSTM-based false data injection attack detection in smart grids, in: 2020 35th Youth Academic Annual Conference of Chinese Association of Automation, YAC, 2020, pp. 638–644, http://dx.doi.org/10.1109/YAC51587.2020.9337674.
[12] Y. He, L. Li, H. Qian, S. Yao, CNN-GRU based fake data injection attack detection method for power grid, in: 2022 2nd International Conference on Electrical Engineering and Control Science, IC2ECS, 2022, pp. 408–411, http://dx.doi.org/10.1109/IC2ECS57645.2022.10087906.
[13] A. Ashok, M. Govindarasu, V. Ajjarapu, Online detection of stealthy false data injection attacks in power system state estimation, IEEE Trans. Smart Grid 9 (3) (2018) 1636–1646, http://dx.doi.org/10.1109/TSG.2016.2596298.
[14] W. Xia, Y. Li, L. Yu, D. He, Locational detection of false data injection attacks in the edge space via hodge graph neural network for smart grids, IEEE Trans. Smart Grid (2024) 1, http://dx.doi.org/10.1109/TSG.2024.3389948.
[15] W. Xia, D. He, L. Yu, Locational detection of false data injection attacks in smart grids: A graph convolutional attention network approach, IEEE Internet Things J. 11 (6) (2024) 9324–9337, http://dx.doi.org/10.1109/JIOT.2023.3323565.
[16] M. Kesici, M. Mohammadpourfard, K. Aygul, I. Genc, Deep learning-based framework for real-time transient stability prediction under stealthy data integrity attacks, Electr. Power Syst. Res. 221 (2023) 109424, http://dx.doi.org/10.1016/j.epsr.2023.109424, URL: https://www.sciencedirect.com/science/article/pii/S0378779623003139.
[17] M. Elimam, Y.J. Isbeih, S.K. Azman, M.S.E. Moursi, K.A. Hosani, Deep learning-based PMU cyber security scheme against data manipulation attacks with WADC application, IEEE Trans. Power Syst. 38 (3) (2023) 2148–2161, http://dx.doi.org/10.1109/TPWRS.2022.3181353.
[18] Y. Raghuvamsi, K. Teeparthi, Detection and reconstruction of measurements against false data injection and DoS attacks in distribution system state estimation: A deep learning approach, Measurement 210 (2023) 112565, http://dx.doi.org/10.1016/j.measurement.2023.112565, URL: https://www.sciencedirect.com/science/article/pii/S026322412300129X.
[19] R. Mazumder, T. Hastie, R. Tibshirani, Spectral regularization algorithms for learning large incomplete matrices, J. Mach. Learn. Res. 11 (2010) 2287–2322.
[20] N. Jain, A. Gupta, V.A. Bohara, PCI-MDR: Missing data recovery in wireless sensor networks using partial canonical identity matrix, IEEE Wirel. Commun. Lett. 8 (3) (2019) 673–676, http://dx.doi.org/10.1109/LWC.2018.2882403.

[21] F. ALmutairy, R. Shadid, S. Wshah, Identification and correction of false data injection attacks against AC state estimation using deep learning, in: 2020 IEEE Power & Energy Society General Meeting, PESGM, 2020, pp. 1–5, http://dx.doi.org/10.1109/PESGM41954.2020.9282037.

[22] J.J.Q. Yu, A.Y.S. Lam, D.J. Hill, Y. Hou, V.O.K. Li, Delay aware power system synchrophasor recovery and prediction framework, IEEE Trans. Smart Grid 10 (4) (2019) 3732–3742, http://dx.doi.org/10.1109/TSG.2018.2834543.

[23] Y. Li, Y. Wang, S. Hu, Online generative adversary network based measurement recovery in false data injection attacks: A cyber-physical approach, IEEE Trans. Ind. Inform. 16 (3) (2020) 2031–2043, http://dx.doi.org/10.1109/TII.2019.2921106.

[24] J. Song, C. Meng, S. Ermon, Denoising diffusion implicit models, 2022.

[25] A. Lugmayr, M. Danelljan, A. Romero, F. Yu, R. Timofte, L. Van Gool, Repaint: Inpainting using denoising diffusion probabilistic models, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 11451–11461, http://dx.doi.org/10.1109/CVPR52688.2022.01117.

[26] Y. Pang, J. Mao, L. He, H. Lin, Z. Qiang, An improved face image restoration method based on denoising diffusion probabilistic models, IEEE Access 12 (2024) 3581–3596, http://dx.doi.org/10.1109/ACCESS.2024.3349423.

[27] J.L. Alcaraz, N. Strodthoff, Diffusion-based time series imputation and forecasting with structured state space models, Trans. Mach. Learn. Res. (2023) URL: https://openreview.net/forum?id=hHiIbk7ApW.

[28] J. Ho, A. Jain, P. Abbeel, Denoising diffusion probabilistic models, 2020.

[29] J. Gilmer, S.S. Schoenholz, P.F. Riley, O. Vinyals, G.E. Dahl, Neural message passing for quantum chemistry, 2017.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, 2023.

[31] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015.

[32] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein GANs, 2017.

[33] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

[34] A. Cini, I. Marisca, C. Alippi, Filling the gaps: Multivariate time series imputation by graph neural networks, 2022.

[35] V. Fortuin, D. Baranchuk, G. Rätsch, S. Mandt, GP-VAE: Deep probabilistic time series imputation, 2020.