ABSTRACT

SONG, LIDONG. Deep Learning for Power System Synthetics Data Generation and Anomaly Detection. (Under the direction of Dr. Ning Lu).

Electric power systems (EPS) have evolved significantly over the past few decades. The incorporation of distributed energy resources has made the power grid a more complex, unstable, and nonlinear structure than conventional power grids. Meanwhile, artificial intelligence (AI) and deep learning (DL) has been widely adopted recently due to their durability and versatility in managing complex nonlinearity on large-scale systems. This paper's work focuses on leveraging state-of-the-art DL technology to solve three power system problems: load profile super-resolution (LPSR), load profile inpainting for missing load data restoration and baseline estimate; False data injection attack (FDIA) and detection in distribution system with integration of battery energy storage system (BESS).

First, ProfileSR-GAN, a novel two-stage LPSR framework, is proposed to upsample low-resolution load profiles (LRLPs) to high-resolution ones (HRLPs). A GAN-based model is applied in the first stage to restore high-frequency components from the LRLPs. The GAN-based model uses weather data and LRLPs to reflect load-weather dependency. In the second stage, a polishing network directed by outline loss and switching loss removes unrealistic power fluctuations in generated HRLPs and improves point-to-point matching accuracy. New load shape evaluation measures are designed to evaluate the HRLPs' realisticness. Simulation findings show that the ProfileSR-GAN outperforms the state-of-the-art approaches in all shape-based metrics and can achieve equivalent point-to-point matching accuracy. After converting LRLPs to HRLPs, non-intrusive load monitoring can be significantly improved.

Next, a GAN-based load profile inpainting network (Load-PIN) is proposed to restore missing load data segments and estimate the baseline for a demand response event. The inputs

are time series load data before and after the inpainting period, together with explanatory variables (e.g., weather data). The innovation of the Load-PIN lies in the design of the GAN Generator. A Generator structure consisting of a coarse network and a fine-tuning network is proposed. The coarse network provides an initial estimation of the data. The fine-tuning network consists of self-attention blocks and gated convolution layers for adjusting the initial estimates. The Load-PIN is tested on three real-world datasets for two applications: patching missing data and deriving baselines of conservation voltage reduction (CVR) events. Compared with the state-of-the-art methods, the simulation results show that Load-PIN is more flexible and achieves 15-30% accuracy improvement.

Finally, a reinforcement learning (RL) based stealth false data injection attack (FDIA) model and a temporal graph convolution network (TGCN) based detection algorithm is proposed to mitigate the risk of false data injection attacks against the battery state of charge (SoC) of the distribution system with BESS integration. First, an agent based on deep Q-learning was developed to solve the contradiction between the high computational cost of nonlinear stealth FDIA constraints and real-time online deployment. A state estimation (SE) based bad data detection (BDD) environment is developed to interact with the RL agent, through which the agent learns to launch stealth SoC FDIA that can avoid being identified by SE-based BDD. Then, TGCN, a deep learning model that combines GCN and gated recurrent unit (GRU), is proposed to leverage the graph nature of the power grid and the temporal features of time-series measurements for FDIA detection. The experiment demonstrates that the Grid-TGCN model outperforms the state-of-the-art method in terms of accuracy and F1 score while keeping a lightweight model structure, making it a promising solution for deploying and using in large-scale complex power grids.

Deep Learning for Power System Synthetics Data Generation and Anomaly Detection

by
Lidong Song

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Electrical Engineering

Raleigh, North Carolina
2023

APPROVED BY:

_____	_____
Dr. Ning Lu Committee Chair	Dr. Wenyuan Tang
_____	_____
Dr. Huaiyu Dai	Dr. Yunan Liu

DEDICATION

To my mother Yunping Fang, my father Lin Song, and my wife-to-be Kailu Xiao.

BIOGRAPHY

Lidong Song was born in Henan, China. He received his B.S. degree in electrical engineering from China University of Mining & Technology - Beijing in 2016, and his M.S. degree in electrical engineering from Xi'an Jiaotong University in 2019. He is currently pursuing a Ph.D. degree in electrical engineering at North Carolina State University, Raleigh, USA. His research interests include deep learning and data-driven applications in distribution systems, such as load profile super-resolution, non-intrusive load monitoring, baseline load estimation, and cyber security.

ACKNOWLEDGMENTS

First and foremost, I am extremely grateful to my advisor, Prof. Ning Lu, for her invaluable advice, continuous support, and patience during my Ph.D. study. Her immense knowledge, open mind, and persistent exploration attitude have encouraged me all the time in my academic research and daily life. I feel so lucky to be a Ph.D. student under her guidance. Apart from my advisor, I also would like to thank Dr. David Lubkeman, for all the guidance, support, and instruction he provided me throughout my doctoral studies.

I would also express my deepest gratitude and appreciation to my committee, including Prof. Wenyuan Tang, Prof. Huaiyu Dai, and Prof. Yunan Liu, for their invaluable time and guidance. Their enthusiasm for research always inspires me to devote myself to my own research and the exploration of new fields.

I would like to extend my sincere thanks to my labmates at GridWrx. Special thanks to Dr. Yiyan Li. Our discussions and collaborations have been key factors in making my research progress in deep learning-related topics. Thanks to Bei Xv, and Victor Daldegan Paduani, I will always remember the great time we worked together. I also would like to thank Fuhong Xie, Ming Liang, Mingzhi Zhang, Rongxing Hu, Si Zhang, Han Pyo Lee, Qi Xiao, Yi Hu, Kai Ye, Charles Kelly, and Hyeonjin Kim. It truly has been a very good time in this lab with all of you.

Lastly, my family deserves endless gratitude. I would like to thank my parents whose love and guidance are with me in whatever I pursue. They are the ultimate role models. And a very special thanks to my loving and supportive wife-to-be Dr. Kailu Xiao, who has provided me with unending inspiration, and moral support during the past nine years. I could not accomplish this without your support and encouragrment.

TABLE OF CONTENTS

LIST OF TABLES

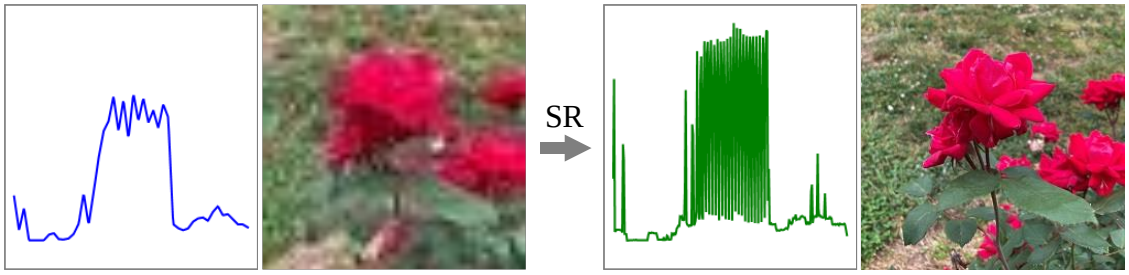
LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Load Profile Super Resolution

In recent years, there has been a steady increase in the deployment of smart meters, which has supported the explosion of data-driven research in power systems, particularly load consumption data-supported applications, such as load analysis, load forecasting, and load management. However, due to factors such as the cost of data communication and storage, most utilities only collect low-resolution smart meter data. A common practice to record the total energy consumed through smart meters is to collect the load power data with a sampling interval of 15 or 30 minutes, through which an average power consumption for the interval is computed. During this process, fast power variations within each sampling interval are lost, as shown in Figures 1.1 (a) and (c). This is called the smoothing effect of signal averaging.

In recent years, two global activities, electrification in the transportation sector [1, 2] and decarbonization [3] in the energy sector, greatly expedited the integration of distributed energy resources (DERs), such as solar, wind, batteries, controllable loads, and electric vehicle chargers. However, uncertainties and variabilities inherent in renewable generation outputs and battery charging/discharging actions will lead to more frequent significant and rapid power variations, causing circuit overloads and weakening the voltage stability [4]. Consequently, for high-DER penetration distribution grids, where fast power variations are visible, it is increasingly important to use HR load profiles for planning and operation studies such as quasi-static power flow analysis or non-intrusive load monitoring (NILM).



(a) LR load profile (b) LR image (c) HR load profile (d) HR image

Figure 1.1. An illustration of super-resolution problems

Recovering the HR data from the LR data is a data up-sampling process. In the past, interpolation is widely used to up-sample LR load profiles or patch the missing data. However, the main drawback of interpolation is that it cannot restore intra-interval power fluctuations.

In image processing, the super-resolution (SR) [5] refers to the technique of generating HR images from LR images, as illustrated in Figures 1.1 (c) and (d). A wide variety of deep-learning methods have been developed for image SR, for example, ultrasound imaging [6], line-fitting [7], and iris recognition [8]. Similarly, in audio signal processing, audio super-resolution (ASR) is used to recover the HR audio signals from the LR signals using deep-learning models [9].

Motivated by image and audio SR, we define load profile super-resolution (LPSR) as a technique for generating realistic HR load profiles from LR load profiles. In power system applications, the development of LPSR is still in its infancy stage. In [10], Liu et al. developed super-resolution perception (SRP) for processing smart meter data. SRP combines Convolution Neural Network (CNN) with supervised learning based on Mean Square Error (MSE) loss. However, the MSE-based supervised learning algorithms can introduce unrealistic details and cause over-smoothing in the reconstructed HR data. This drawback has also been widely observed in image recovery studies [11].

Generative Adversarial Network (GAN) [12] based methods are widely used in solving image SR problems. Because the adversarial training between Generator and Discriminator can effectively capture the inherent probability distribution of the HR data, GAN-based methods can usually obtain more realistic SR results than the CNN or MSE-based approaches [13].

In the power system domain, GAN has been used for generating load profiles [14, 15], wind farm outputs [16], simulating load forecasting uncertainty [17], synthesizing appliance power signatures [18], and appliance-level energy disaggregation [19]. However, to the best of our knowledge, using GAN-based methods for LPSR is still an uncharted area. Chapter 2 proposes a novel two-stage LPSR framework to up samples low-resolution load profiles to high-resolution ones. Simulation findings show that the ProfileSR-GAN outperforms the state-of-the-art approaches in all shape-based metrics and can achieve equivalent point-to-point matching accuracy.

1.2 Customer Baseline Load Estimation

The reliability and flexibility of power system operation face severe challenges due to the growing uncertainties from the supply side because of the fast-increasing penetration of renewable energy sources. Demand Response (DR) aims at improving the operational efficiency of power plants and grids, and it constitutes an effective means of reducing grid risk during peak periods to ensure the safety of power supplies. One key challenge related to DR is customer baseline load (CBL) calculation. A fair and accurate baseline provides helpful information for resource planners and system operators who wish to implement DR programs. Most current CBL estimation methods for incentive-based demand response rely heavily on historical data and are unable to adapt to the cases when the load patterns in the DR event day are not similar enough to those in non-DR days.

One category of CBL estimation methods that are early developed and widely adopted by ISOs is average-based methods. Examples of averaging approaches include the direct average of X previous days; the average of the highest X of Y days (HighXofY); the average of the middle X of Y days (MidXofY); and an average of the lowest X of Y days Significant disadvantage of these averaging-based methods is that they can lead to significant errors when applied to residential customers. This is because residential load patterns cannot be maintained at the same stable level as commercial and industrial loads. Moreover, aggregated residential load patterns tend to have more robust heterogeneity when compared to industrial and commercial loads because residential electricity consumption directly relates to random human activities. Therefore, the loads are more vulnerable to changes in natural and social factors.

To solve the heterogeneity issue in residential CBL estimation, regression-based methods are developed, including Support Vector Regression [20-22], Gaussian Process Regression [23, 24], Lasso regression [25, 26], Quantile regression [27], and Linear Regression [28, 29]. To improve the CLB performance, control group and clustering-based methods [20, 21, 25, 29, 30] are introduced and combined with regression tools. However, these regression-based methods are usually highly customized and too sophisticated to reproduce, which is limited the wide deployment in industrial applications.

At the same time, the booming of machine learning techniques and the availability of large-scale smart meter data motivates machine learning-based methods. [31] proposed using the Long Short Term Memory (LSTM) network to predict the baseline load for incentive-based demand response. [32] adopted the stacked auto-encoders (SAEs) under the federated learning (FL) framework.

This paper proposes a novel Generative Infilling Network (GIN) combing the Gated Convolution Neural Network [33] and the multi-head attention machoism proposed in Transform Network [34]. Metrics evaluation demonstrates that the proposed GIN method outperforms the state-of-the-art method, which achieves higher accuracy with a smaller network scale. Algorithm design and implementation details are explained in Chapter 3.

1.3 Attack and Anomaly Detection for Smart Distribution Grid With BESS

Smart grid's innovative technology will revolutionize our society, economy, and environment, which will be achieved by integrating information and communication technologies (ICT) into the power grid, evolving it into a cyber-physical system (CPS). In this process, however, security concerns have taken a back place. With the numerous ICT components, the smart grid's security has been severely weakened by the increasing number of cyber-attacks. This created a great deal of concern regarding the dependability and security of the ever-desired smart grid in light of the tremendous economic and stability concerns.

With the development of energy storage technology, the cost of energy storage units, such as lithium batteries and lead-acid batteries, has decreased significantly. The Battery Energy Storage System (BESS) is, therefore, widely deployed in the distribution network. It plays a crucial role in grid operation, such as energy management and demand response under the grid-connected state mode, or works as a grid-forming unit to realize volt-var control and frequency regulation of the distribution network under island mode. Due to the vast energy storage capacity of battery cells deployed on the grid, malicious attacks that cause failures will result in immeasurable losses and safety risks. Consequently, cyberattacks against BESS have aroused widespread concern and interest.

In light of the aforementioned context, this paper conducts an in-depth analysis of the distribution network's hidden False Data Injection Attack (FDIA). A stealth FDIA algorithm based on reinforcement learning is proposed against the current State Estimation (SE) based Bad Data Detection (BDD) system. Then, considering the graph nature of the power grid, a Graph Convolutional Network (GCN) based anomaly detection algorithm is designed. The experimental results demonstrate that the proposed RL-based attack is capable of evading the detection of existing BDD. The GCN-based anomaly detection algorithm identifies concealed FDIA effectively.

CHAPTER 2 LOAD PROFILE SUPER-RESOLUTION

2.1 Background

This chapter presents a novel two-stage load profile super-resolution (LPSR) framework, ProfileSR-GAN, to upsample the low-resolution load profiles (LRLPs) to high-resolution load profiles (HRLPs). It is a common practice for utilities to downsample smart meter measurements from high-resolution (HR) to low-resolution (LR) (e.g., 15-, 30-, or 60-minute)[35]. This will significantly lower the costs of communicating, storing, and processing the data collected from millions of smart meters. But at the same time, it brings certain negative effects, causing a lot of helpful user load information to be lost in the down-sampling process. At the same time, however, it causes many useful user load information to be lost in the downsampling process. Therefore, it is necessary to design a way to recover this lost high-resolution information from low-resolution data, which can better guide the supplier's service strategy and achieve the effect of energy saving and efficient operation.

The LPSR problem is formulated as a Maximum-a-Posteriori problem. In the first stage, a GAN-based model is adopted to restore high-frequency components from the LRLPs. To reflect the load-weather dependency, aside from the LRLPs, the weather data is added as an input to the GAN-based model. In the second stage, a polishing network guided by outline loss and switching loss is novelly introduced to remove the unrealistic power fluctuations in the generated HRLPs and improve the point-to-point matching accuracy. To evaluate the realisticness of the generated HRLPs, a new set of load-shape evaluation metrics is developed. Simulation results show that: i) ProfileSR-GAN outperforms the state-of-the-art methods in all shape-based metrics and can achieve comparable performance with those methods in point-to-point matching accuracy, and ii) after applying ProfileSR-GAN to convert LRLPs to HRLPs, the

performance of a downstream task, non-intrusive load monitoring, can be significantly improved. This demonstrates that ProfileSR-GAN is a compelling new mechanism for restoring high-frequency components in down-sampled time-series data sets and improves the performance of downstream tasks that require HR load profiles as inputs.

We present a two-stage load profile super-resolution (LPSR) framework, ProfileSR-GAN. In the first stage, a GAN-based model is adopted to restore high-frequency components from the low-resolution load profiles (LRLPs). To reflect the load-weather dependency, aside from the LRLPs, the weather data is added as an input to the GAN-based model. The LPSR problem is formulated as a Maximum-a-Posteriori problem. To make the generated HRLPs more realistic, we use the method introduced in [11, 36] to construct the Generator loss function so that adversarial and feature-matching losses can be used to recover the high-frequency components missed in the downsampling process effectively. In the second stage, a polishing network is designed to connect to the GAN model to remove unrealistic power fluctuations from the generated HRLPs. A new set of load shape evaluation metrics is developed for evaluating the realisticness of the generated profiles and for comparing the performance with other state-of-the-art algorithms.

Firstly, we formulated the LPSR problem as a Maximum-a-Posteriori problem that can be solved using GAN-based approaches. We added weather data as an input to the GAN model to reflect the load-weather dependency. Secondly, we proposed connecting a polishing network to the GAN model to remove unrealistic power fluctuations in the GAN-generated HR load profiles. This significantly improves the ProfileSR-GAN performance on the point-to-point matching accuracy. Thirdly, we designed the performance evaluation metrics for evaluating the realisticness of the generated HR profiles. Our simulation results show that ProfileSR-GAN

outperforms the state-of-the-art algorithms. As an effective mechanism for restoring high-frequency components, ProfileSR-GAN can improve the performance of downstream tasks (e.g., NILM) that require HR load profiles as inputs.

The rest of the chapter is organized as follows. Section 2.2 formulates the LPSR problem, and Section 2.3 introduces the ProfileSR-GAN framework. Experimental results are presented in Sections 2.4 and 2.5. Section 2.6 concludes the chapter.

2.2 Load Profile Super Resolution Problem Formulation

Let P^{HR} represent the HR measurements with N data points. P^{LR} is the set of LR measurements down-sampled from P^{HR} by averaging α continuous samples, where α is called the *scale-up factor*. Thus, P^{LR} has M data points and $M = N/\alpha$. The down-sampling process can be expressed as

$$P_m^{\mathrm{LR}} = \frac{1}{\alpha} \sum_{n=\alpha(m-1)+1}^{\alpha m} P_n^{\mathrm{HR}} + \eta_m \quad \forall m \in M, \forall n \in N \tag{2.1}$$

where η represents noises caused by disturbances in the data acquisition process; n and m are the index of the HR and LR measurements, respectively.

As shown in Figure 2.1, when a 1-minute load profile is down-sampled to 5-, 15-, and 30-minute load profiles, the high-frequency components will be filtered out because of the smoothing effect in signal averaging. The smoothing effect becomes more apparent when α increases. Compared with the 1-minute load profile, the 15- and 30-minute load profiles have lower load peaks and contain slower power variations. In addition, individual device on/off and cycling behaviors are no longer distinguishable. In the following sections, we will introduce the ProfileSR-GAN for restoring the intra-interval power variations from LR load profiles.

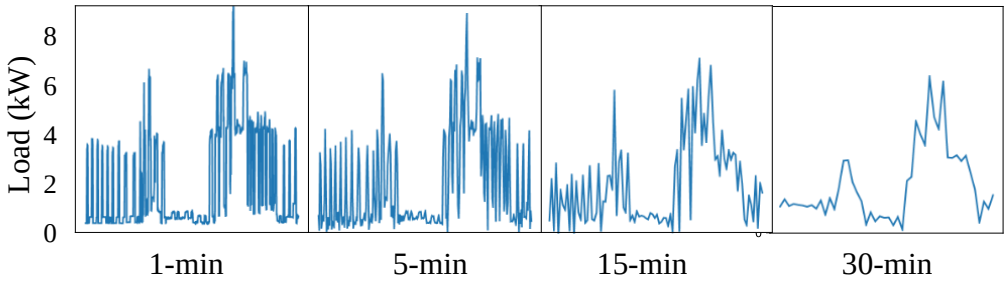


Figure 2.1. Daily load profiles with different granularity

Strengthening of Smart Grid functionalities has become the need of the 21st Century. Security evolves to be the primary concern at the deployment level of Smart Grids. Cyber security threats and vulnerabilities in smart grid networks must be addressed before the deployment of the Smart Grid. Our proposed intrusion detection scheme identifies anomalies in the Smart Grid traffic and detects attacks like flooding, which causes Denial of Service in Smart Grid Networks.

In this paper, we plan to leverage deep learning techniques for the clustering of traffic data and outlier detection for the data transmitted between the utility data center and the field devices.

Super-resolution algorithms originate from the image processing domain for processing 2-dimensional images. The 1-dimensional LPSR problem can be formulated as a Maximum a Posteriori (MAP) estimation problem as introduced in [37].

From the Bayesian rule, we have

$$p\left(P^{\mathrm{HR}} \mid P^{\mathrm{LR}}\right) = \frac{p\left(P^{\mathrm{LR}} \mid P^{\mathrm{HR}}\right) p(P^{\mathrm{HR}})}{p\left(P^{\mathrm{LR}}\right)} \tag{2.2}$$

where $p(P^{\mathrm{HR}}| P^{\mathrm{LR}})$ is the conditional probability of P^{HR} given P^{LR}, $p(P^{\mathrm{LR}}| P^{\mathrm{HR}})$ is the conditional probability of P^{LR} given P^{HR}, and $p(P^{\mathrm{HR}})$ and $p(P^{\mathrm{LR}})$ is the prior probability of the HR and LR load profiles, respectively.

The objective function of an LPSR problem is to maximize the probability of the occurrence of P^{HR} for a given P^{LR}. Using the MAP estimation method presented in [37], the estimated HR profile, $\hat{P}^{HR}$, can be obtained by

$$\hat{P}^{HR} = \arg\max_{\hat{P}^{HR}} \left(\log p\left(P^{LR} \mid \hat{P}^{HR} \right) + \log p\left(\hat{P}^{HR} \right) \right) \tag{2.3}$$

where $\hat{P}^{HR}$ is the estimated HR load profile.

The conditional probability of observing P^{LR} given $\hat{P}^{HR}$ under a Gaussian noise (i.e., η in (2.1) has a zero mean with variance σ) can be calculated as

$$p\left(P^{LR} \mid \hat{P}^{HR} \right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-\left\| \hat{P}^{LR} - P^{LR} \right\|_2^2}{2\sigma^2} \right) \tag{2.4}$$

where $\hat{P}^{LR}$ is the LR load profile derived from $\hat{P}^{HR}$.

Substituting (2.4) into (2.3), we have

$$\hat{P}^{HR} = \arg\min_{\hat{P}^{HR}} \left(\left\| \hat{P}^{LR} - P^{LR} \right\|_2^2 - \lambda \log p\left(\hat{P}^{HR} \right) \right) \tag{2.5}$$

where$\lambda = 2\sqrt{2\pi}\sigma^3$ is the regularization factor.

In (2.5), the second term represents the prior knowledge of P^{HR}. Traditionally, the MSE-based method tries to find a $\hat{P}^{HR}$ by minimizing the distance between P^{LR} and $\hat{P}^{LR}$ (i.e., $\|\hat{P}^{LR} - P^{LR}\|_2^2$). However, in practice, there is only one LR observation for a HR profile. As a result, solely relying on $\|\hat{P}^{LR}\text{-}P^{LR}\|_2^2$ can make the LPSR problem ill-posed, i.e., (2.3) may not have a unique solution [5]. For example, it is possible that the same LR profile can be obtained by down-sampling two different HR profiles.

Therefore, it is essential to constrain the solution space by introducing the prior knowledge of P^{HR} into the SR problem formulation, as formulated by the second term in (2.5). Note that we refer to prior knowledge as the information a learner already has before learning a

new problem [38]. In this paper, two types of prior knowledge are used. First, weather data and LR profiles are used as inputs to account for the known dependency of electricity consumption on the weather. Second, the two terms used in the generator loss function (refer to Section II.C): *adversarial loss* and *feature-matching loss*, represent prior knowledge of the shape characters extracted by the discriminator network from the actual waveforms.

2.3 ProfileSR-GAN Network Design And Implementation Details

2.3.1 Generative Adversarial Network

A GAN model consists of two components: a generator network (G) and a discriminator network (D), as shown in Figure 2.2. A latent vector **z,** usually a Gaussian noise, is used as the input to generate the target output $G(\mathbf{z})$. Then, the generator output, $G(\mathbf{z})$, which is the generated data, and the real data, **x**, are sent to D. The goal of D is to distinguish which data sets are real and which are fake.

The training of a GAN model is an alternative and adversarial process: G tries to generate samples $G(\mathbf{z})$ that can fool D; D learns to identify $G(\mathbf{z})$ from **x** by assigning larger probabilities to **x** and smaller ones to $G(\mathbf{z})$. As introduced in [12], this process is formulated as a minimax game

$$\min_G \max_D V(D,G) = \mathrm{E}_{\mathbf{x}\sim p_{(\mathbf{x})}}[\log D(\mathbf{x})] + \mathrm{E}_{\mathbf{z}\sim p_{(\mathbf{z})}}[\log(1-D(G(\mathbf{z})))] \tag{2.6}$$

where $V(D,G)$ is the reward function, $p_{(\mathbf{x})}$ and $p_{(\mathbf{z})}$ are the probability distributions of training data and latent vector, E is the expectation operator.

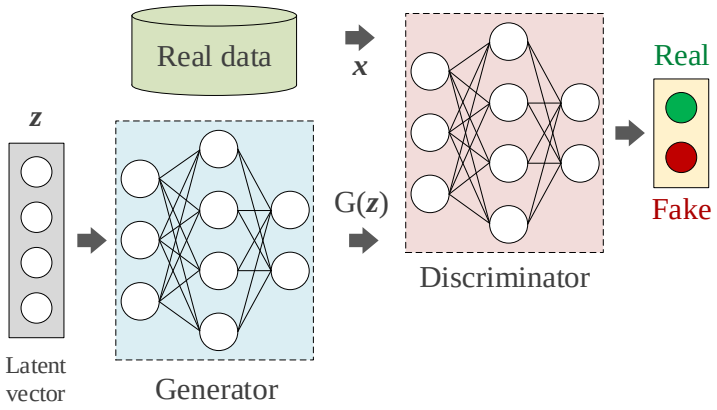


Figure 2.2. The original generative adversarial network (GAN) model

2.3.2 Profilesr-GAN

This section introduces architecture design, generator loss function selection, and polishing network loss function selection of the proposed ProfileSR-GAN framework.

As shown in Figure 2.3, ProfileSR-GAN is a two-stage process. In the first stage, LR profiles and their corresponding weather data are used as inputs of the GAN-based model to generate HR profiles through adversarial training. In the second stage, a polishing network will remove unrealistic power fluctuations from the GAN-generated HR profiles.

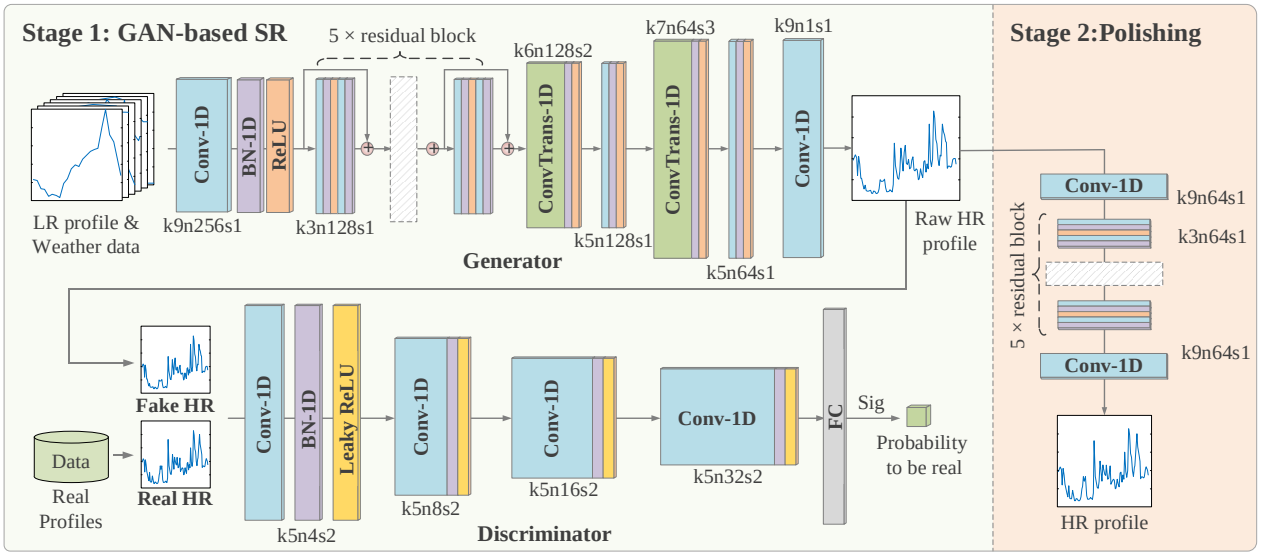


Figure 2.3. The two-stage ProfileSR-GAN architecture with corresponding kernel size (k), number of feature maps (n), and stride (s) indicated for each convolution layer

The generator network is a deep CNN. First, convolution layers are used to extract high-level features from the input data. Then, we implement transpose convolution layers to recover the HR profiles. Inspired by [36], we use ReLU as the activation function. Residual blocks are inserted between two consecutive convolutional layers [39] to alleviate the gradient diminishing issue. We also adopt batch normalization following each convolutional layer [40] to enhance the training process.

The architecture summarized by Radford et al. [41] is used for constructing the discriminator network. The activation function is LeakyReLU. The discriminator network is trained to solve the maximization problem defined by (2.4). It contains four convolutional layers with an increasing number of kernels from 4 to 32. This allows us to compress the input profiles to high-level feature maps. Finally, the resulting feature maps will go through a fully connected (FC) layer and a sigmoid function to obtain the probability for real/fake classification. The polishing network is also a deep CNN bearing similar network structures as the generator, except that the two up-sampling transpose convolution layers are removed, and the number of kernels is reduced.

2.3.3 Loss Function Design For GAN Networks

Let θ_G be the parameter of the generator network. The generator loss, L_G, is minimized to find an optimal θ_G by

$$\min_{\theta_G} L_G\left(G_{\theta_G}\left(P^{\mathrm{LR}}\right), P^{\mathrm{HR}}\right) \tag{2.7}$$

$$L_G = L_{cont} + \lambda_1 L_{advs} + \lambda_2 L_{feat} \tag{2.8}$$

where L_{cont} is the content loss; L_{advs} is the adversarial loss; L_{feat} is the feature-matching loss; λ_1 and λ_2 are the weight coefficients.

To compute the content loss, L_{cont}, MSE is used to calculate the point-to-point distance between the generated and the ground truth HR profiles as

$$L_{cont} = \frac{1}{N}\left\| G_{\theta_G}(P^{\mathrm{LR}}) - P^{\mathrm{HR}} \right\|_2^2 \tag{2.9}$$

By minimizing the MSE, the generator is incentivized to find the maximum likelihood estimation of ground truth. However, relying solely on MSE-based L_{cont} leads to over-conservative results in LPSR. As shown in Figure 2.4, the generated HR profile is overly smooth, so it cannot restore high-frequency, significant power variations. To resolve this issue, two loss terms, L_{advs} and L_{feat}, are used in (2.8).

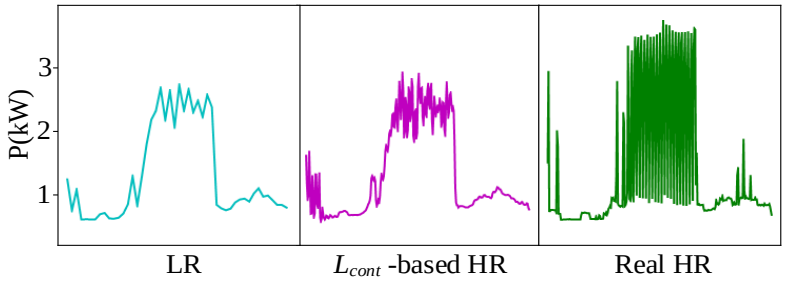


Figure 2.4. The LR profile, HR profile generated only based on L_{cont}, and real HR profile

The discriminator in GAN is trained to distinguish the fake from the real by minimizing the discriminator loss function, L_D, calculated as

$$L_D = -[\log D_{\theta_D}(P^{\mathrm{HR}}) + \log(1 - D_{\theta_D}(G_{\theta_G}(P^{\mathrm{LR}})))] \tag{2.10}$$

where θ_D is the parameters of the discriminator networks.

Let the second term related to θ_G in (2.10) be the adversarial loss L_{advs}. We have

$$L_{advs} = \log(1 - D_{\theta_D}(G_{\theta_G}(P^{\mathrm{LR}}))) \tag{2.11}$$

By minimizing L_{advs}, the generator network favors solutions that cannot be distinguished as "fake" by the discriminator network to make the generated HR more realistic. Inspired by [12], we rewrite (2.11) as

$$L_{advs} = -\log(D_{\theta_D}(G_{\theta_G}(P^{\mathrm{LR}}))) \tag{2.12}$$

to provide sufficient gradients and add robustness to the training process.

The feature-matching loss, L_{feat}, is defined as the distance between high-level feature maps extracted by the hidden layers of the discriminator network [42]. It is calculated as

$$L_{feat} = \sum_{j=1}^{J} \left\| \varphi_j\left(G_{\theta_G}(P^{\mathrm{LR}})\right) - \varphi_j\left(P^{\mathrm{HR}}\right) \right\|^2 \tag{2.13}$$

where $\varphi_j(\cdot)$ represents the output of the j^{th} intermediate convolution layer of the discriminator network, given real/fake HR profiles as inputs. J is the number of intermediate layers involved in the loss function. As shown in Figure 2.5, the extracted feature maps are pretty different between real and fake profiles. Because high-frequency significant power variations can be embedded in those hidden features, using L_{feat} can train the generator to generate more realistic HR profiles.

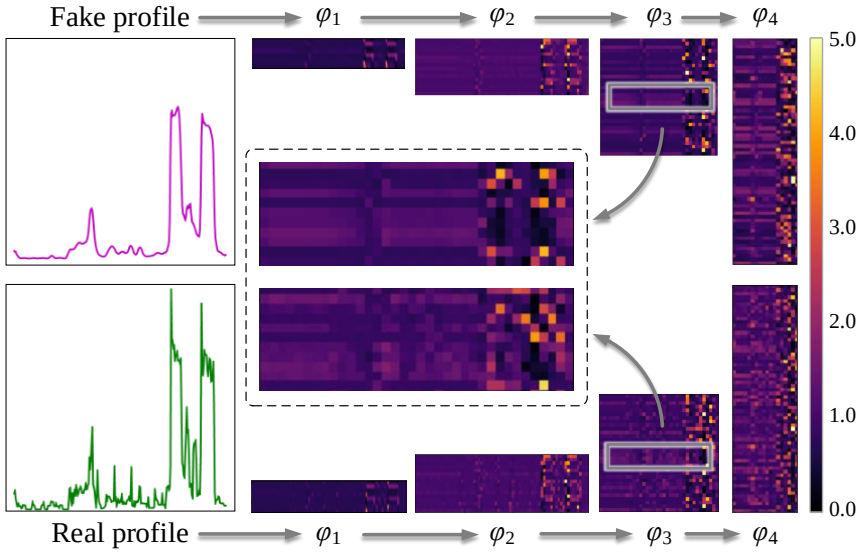


Figure 2.5. Hidden feature maps extracted by the convolutional layers of the discriminator network

Note that the output of the discriminator network is a binary classifier (yes/no) and cannot be directly used to calculate how close the generated load profile resembles the actual. However, one can compare feature values in each hidden convolutional layer when fake and

actual profiles are examined. By minimizing L_{feat}, the generator will favor solutions that share similar features with the actual HR profiles to make the results more realistic.

2.3.4 Loss Function Design For The Polishing Network

Since the GAN model recovers the high-frequency components by reducing the adversarial loss and feature-matching loss, its performance in minimizing MSE loss is compromised. This is because point-to-point matching accuracy is sacrificed in exchange for the flexibility of generating more realistic details.

To resolve this issue, the loss function of the polishing network, L_{pol}, is designed to have two new loss terms: the *outline loss*, L_{outl}, and the *switching loss*, L_{swit}, so we have

$$L_{pol} = L_{outl} + L_{swit} \tag{2.14}$$

$$L_{outl} = \frac{1}{N}\left\|\xi_{\max}(\hat{P}^{\mathrm{HR}}) - \xi_{\max}(P^{\mathrm{HR}})\right\|_2^2 + \frac{1}{N}\left\|\xi_{\max}(-\hat{P}^{\mathrm{HR}}) - \xi_{\max}(-P^{\mathrm{HR}})\right\|_2^2 \tag{2.15}$$

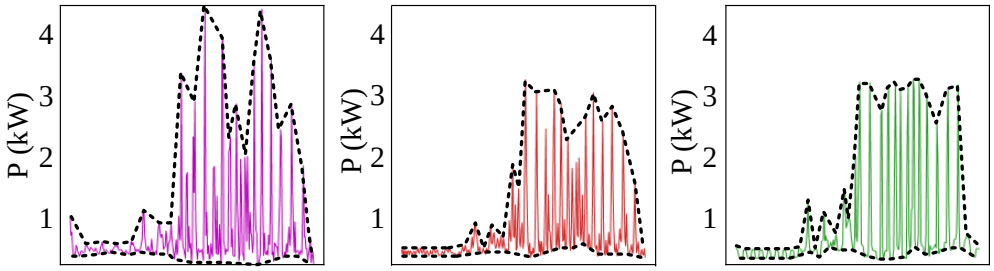
$$L_{swit} = \frac{1}{N}\left\|\xi_{\max}\left|\Delta\hat{P}^{\mathrm{HR}}\right| - \xi_{\max}\left|\Delta P^{\mathrm{HR}}\right|\right\|_2^2 \tag{2.16}$$

$$\Delta\hat{P}^{\mathrm{HR}} = \hat{P}^{\mathrm{HR}}(n+1) - \hat{P}^{\mathrm{HR}}(n), \quad \Delta P^{\mathrm{HR}} = P^{\mathrm{HR}}(n+1) - P^{\mathrm{HR}}(n)$$

where $\xi_{\max}$ is the max pooling operator [43] moving across the entire signal with a kernel size of k_{max} at stride s_{max} , Δ is the first-order difference operator. Note that L_{outl} focuses on comparing the local peaks and valleys of the generated profile with the ground truth profile. This is also a proven effective solution used in solving image segmentation problems [44]. L_{swit} focuses on comparing the change of load between two consecutive sampling intervals so that the load changing rates are similar to the ground truth profile.

Figure 2.6 shows an example of a daily load profile before and after polishing. Note that the on/off of appliances normally lead to flat upper and lower boundaries instead of arbitrary fluctuations. This is because an appliance usually runs at a relatively fixed power level.

Together, L_{outl} and L_{swit} help flatten the unrealistic fluctuations to improve point-to-point accuracy.



(a) Before polishing (b) After polishing (c) Ground truth

Figure 2.6. An illustration of comparing the envelopes of the generated daily HR profiles (before and after polishing) with that of the actual daily load profile

2.4 Metrics Evaluation

To train and test the proposed model, we use the 1-minute smart meter data set collected from 148 residential households in Austin, TX in 2015 by the PECAN Street association [45]. The hourly weather-related data set, including dry ball temperature, visibility, humidity, wind speed, sunrise and sunset time, are downloaded from [46] and then up-sampled to minute-level resolution by linear interpolation to pair with the LR load profiles.

The annual data are split into daily data. After excluding the days with missing data or abnormal data, we finally have 53,000 sets of daily load profiles. Those profiles are divided into two groups: 70% for training, 15% for validation, and 15% for testing. The 1-min data is down-sampled to 5-min and 30-min to obtain P^{HR} and P^{LR}, so the scale factor α is 6. The Gaussian noise, η, has a zero mean with a variance of 0.01.

2.4.1 Training Setup

Adam, an algorithm for first-order gradient-based optimization of stochastic objective functions introduced in [47], is used with momentum terms $\beta_1 = 0.99$ and $\beta_2 = 0.999$. The slope of the LeakyReLU is 0.2. Hyperparameters are tuned on the validation dataset listed in Table 2.1.

Deep neural network models are built in the PyTorch environment and trained on a single GPU of NVIDIA GeForce RTX 3080. The training time is approximately 10 hours.

To demonstrate the impact of introducing GAN-based components (L_{advs} and L_{feat}), we design a CNN model with the same network structure as the ProfileSR-GAN generator (see the upper left part in Figure 2.3) as a controlled experiment. The CNN model is purely trained by the MSE loss defined in equation (2.7). The linear interpolation (LERP) method with a scale-up factor of α = 6 is used as the benchmark case. Other SR approaches, including SRP [10], ASR [9] are also included in performance evaluation.

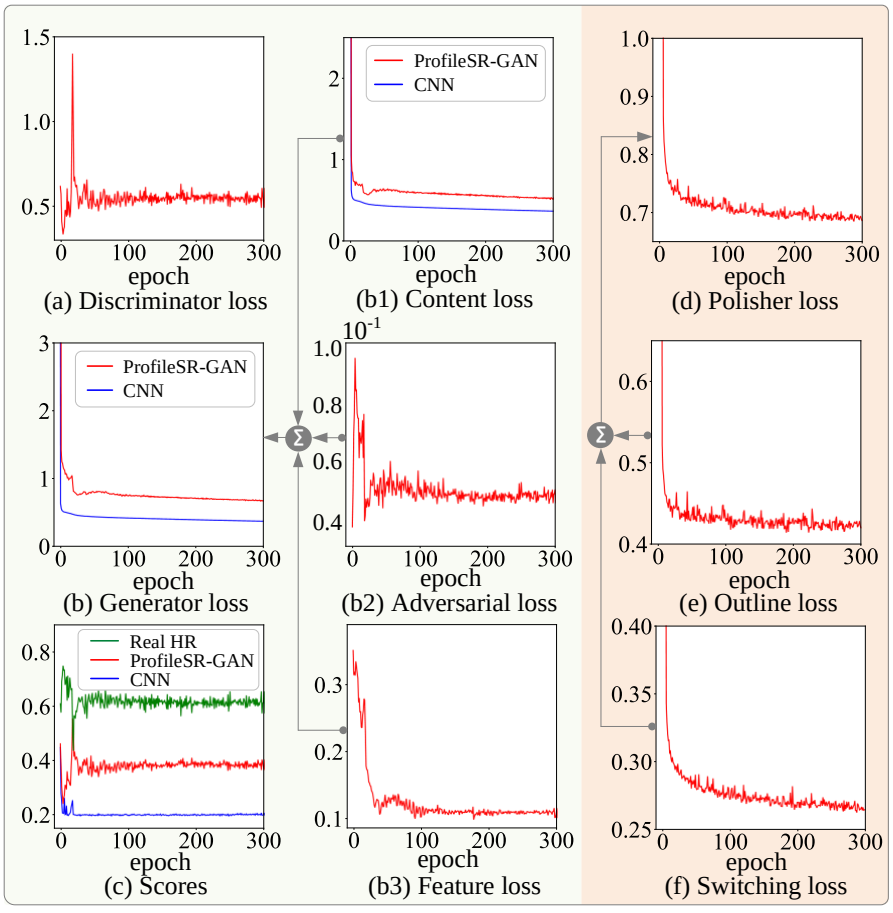


Figure 2.7. Training curves of ProfileSR-GAN (a) discriminator loss of ProfileSR-GAN, (b) generator loss of ProfileSR-GAN and CNN, (b1) content loss of ProfileSR-GAN and CNN, (b2) adversarial loss of ProfileSR-GAN generator, (b3) feature loss of ProfileSR-GAN generator, (c) scores of real, fake HR profiles given by discriminator. (d) polishing loss, (e) outline loss, (f) switching loss.

Table 2.1. Hyperparameter configurations for ProfileSR-GAN

Parameter	Value
Learning rate	1e-4
L_{advs} weight - λ_1	0.05
L_{feat} weight - λ_2	0.5
Batch size	32
Max pooling kernel size - k_{max}	3
Max pooling stride - s_{max}	1
Training epochs	300(GAN) + 300(polishing)

Figure 2.7 shows the loss plots in the two stages and scores given by the discriminator for the real and generated HR profiles during the GAN training process. The following observations are made for each training stage.

Initialization (0 - 10 epoch)

As shown in Figure 2.7(a), initially, there is a sharp decrease of discriminator loss. This means that the discriminator is learning the feature differences between the real and generated HR profiles quickly. Consequently, the discriminator starts to assign a higher score to the real profile and a lower score to the fake one, as shown in Figure 2.7(c). Meanwhile, as shown in Figure 2.7(b), although both CNN and ProfileSR-GAN have decreased generator losses, the generator loss of ProfileSR-GAN starts to bounce back quickly. This is because the decrease in content loss (see Figure 2.7(b1)) is offset by the increase of the adversarial loss (see Figure 2.7(b2)) and the feature-matching loss (see Figure 2.7(b3)).

Evolving (10-50 epoch):

In this stage, for ProfileSR-GAN, the adversarial training allows the generator and discriminator to improve each other. The discriminator loss keeps decreasing, showing that the discriminator becomes more effective in identifying the real HR profiles from the fake ones. Note that despite the increasing generator loss, the performance of the generator continues to improve. This shows that guided by the discriminator, the generator is learning to achieve an optimal trade-off between *Achieving lower MSE error* and *generating more realistic profiles*. In

other words, the content loss is sacrificed to lower the adversarial loss and feature losses. In contrast, the CNN model merely focuses on minimizing the MSE content loss.

Balanced (after 100 epochs):

After about 100 epochs, the generator and the discriminator of ProfileSR-GAN reach a balance in performance. There is no further decrease in discriminator loss, showing that the generator has the ability to generate realistic HR profiles to fool the discriminator. Meanwhile, the scores assigned to profiles by the discriminator are also stabilized: the score of a fake HR profile generated by the CNN model is around 0.2, much lower than those received by ProfileSR-GAN around 0.38). Note that the score of a real profile is approximately 0.62.

For the polishing network, both the outline loss and the switching loss drop sharply during the first 20 epochs and then stabilize at around the 300^{th} epoch.

2.4.2 LPSR Results And Performance Evaluation

In the image SR problem, evaluations of the restored HR images are usually based on a human-judgment-based measurement, called the Mean Opinion Score testing. However, in LPSR, substantial domain knowledge and expertise are required to visually distinguish whether a generated load profile is realistic. Therefore, we need a set of metrics for evaluating the quality of the generated HR load profiles.

MSE, as a point-wise comparison between two waveforms, is insufficient for comparing the shape of waveforms. For example, a minor time shift in waveforms can lead to a large MSE, even though the two waveforms have the same shape. Therefore, in this paper, we introduce three shape-wise load profile evaluation metrics: Peak Load Error (PLE), Frequency Component Error (FCE), and Critical Point Error (CPE). The metrics are calculated as

$$PLE = \left|\max(G_{\theta_G}(P^{\mathrm{LR}})) - \max(P^{\mathrm{HR}})\right| \tag{2.17}$$

$$FCE = \frac{1}{N}\left\| \mathrm{F}\ (G_{\theta_G}(P^{\mathrm{LR}})) - \mathrm{F}\ (P^{\mathrm{HR}}) \right\|_1 \tag{2.18}$$

$$CPE = \frac{1}{N}\left| \sum \mathrm{R}\ (G(P^{\mathrm{LR}})) - \sum \mathrm{R}\ (P^{\mathrm{HR}}) \right| \tag{2.19}$$

where Φ is the Discrete Fourier Transform operator and P is the Ramer Douglas Peucker operator [48].

PLE measures the peak load difference between the ground-truth HR profiles and the restored HR profiles by SR algorithms. PLE has a clear physical meaning and is essential because accurately restoring the intra-interval peak load is critical for the distribution system operation and planning.

FCE measures the frequency-domain similarity between two profiles. Since the critical challenge of LPSR is to restore the intra-interval, high-frequency components, FCE can compare the frequency-domain characteristics to evaluate the effectiveness of the SR algorithms.

Ramer Douglas Peucker algorithm simplifies a waveform in the time domain by eliminating non-critical points and keeping only shape-defining points [48]. Consequently, CPE measures the difference between the number of critical points of two waveforms to compare their similarity.

The LPSR results and metric values are summarized in Table 2.2 and Figures. 2.8 to 2.10. From the results, we have the following observations.

Visual comparison of the generated daily load profiles. As shown in Figures. 2.8 and 2.9, the generated intra 30-min high-frequency components from ProfileSR-GAN are very similar to the ground truth profiles in terms of the magnitude of generated peaks, appliance cycling behaviors, and the envelopes of the load profiles. LERP and ASR fail to generate the intra 30-min power variations. In the daily profiles generated by SRP and CNN, the intra 30-min power

variations are unrealistic with peak loads lower than that of the actual. This shows that ProfileSR-GAN can learn to exclude unrealistic components that are easy-to-be-identified-as-a-fake through the use of a discriminator.

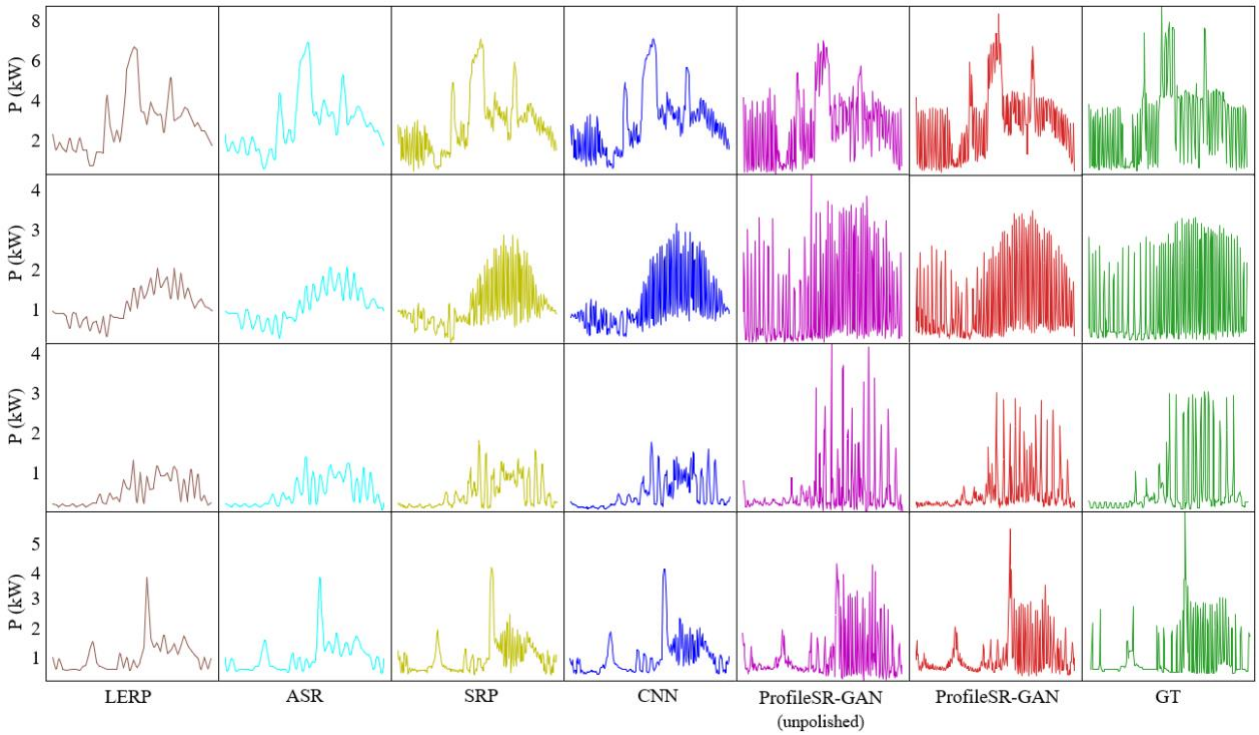


Figure 2.8. LPSR results for generating daily load profiles (upsampling from 30-min to 5-min resolution). Lower resolution (LR), linear interpolation (LERP), convolution neural network (CNN), ProfileSR-GAN (unpolished), ProfileSR-GAN, and ground truth (GT).

Table 2.2 Metric evaluation results

SR method		LERP	ASR	SRP	CNN	ProfileSR GAN- (unpolished)	ProfileSR GAN (polished)
MSE	mean	0.55	0.44	0.42	**0.41**	0.61	0.51
	Gain	/	20%	24%	25%	-11%	7%
PLE	mean	1.38	0.99	0.92	0.91	0.86	**0.73**
	Gain	/	28%	33%	34%	38%	47%
FCE	mean	7.22	5.83	5.36	5.38	4.81	**4.65**
	Gain	/	19%	26%	25%	33%	36%
CPE	mean	0.65	0.41	0.29	0.31	0.26	**0.25**
	Gain	/	37%	55%	52%	60%	62%

* "Gain" represents the improvement w.r.t LERP baseline

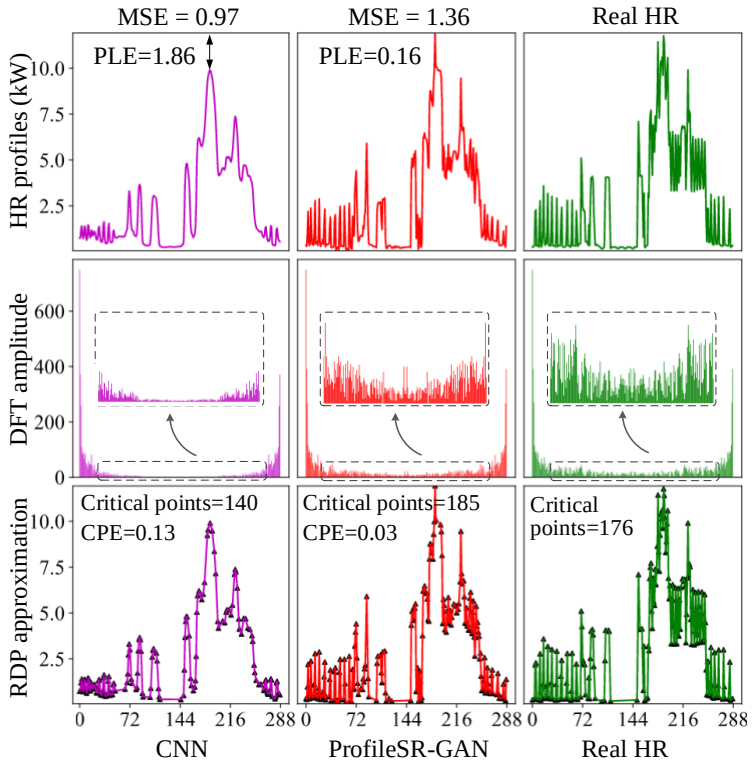


Figure 2.9. Comparison of frequency components and critical point errors

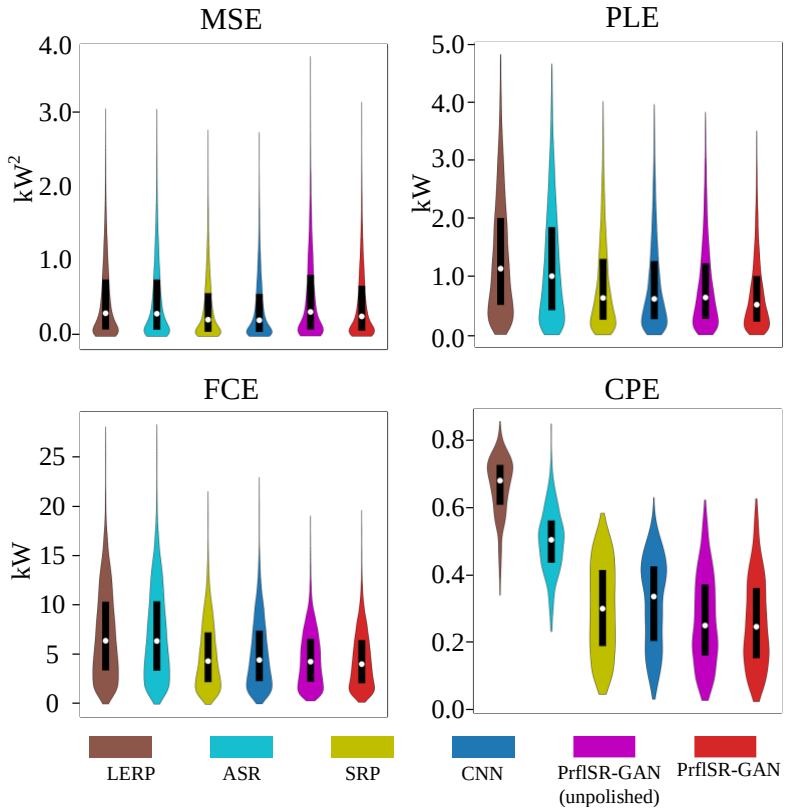


Figure 2.10. Violin plots of the performance metrics. Medians are shown as the white markers

Comparison of MSE: As shown in Table 2.2, MSE-based methods outperform the GAN-based model in achieving the lowest MSE. This is reasonable since MSE is the only optimization objective those models need to take care of. However, only emphasizing point-wise MSE on averaging the point-to-point distance leads to smooth outputs. As shown in Figs. 9 and 10, such processes will filter out high-frequency components because recovering high-frequency detail is risking high point-to-point mismatch. A similar soothing effect has also been observed in the image SR problem and is reported in [11]. The results in Table 2.2 also shows that after adding a polishing network, there is an 18% improvement in MSE, showing the effectiveness of adding a polishing network to the ProfileSR-GAN architecture for improving point-wise accuracy

Comparison of PLE: From Table 2.2 and Figure. 2.9 and 2.9, we can see that the magnitude of the peak in the ProfileSR-GAN generated profile is very similar to that in the ground truth curves. On the contrary, the peak load restored by LERP and MSE-based methods has a relatively large gap with the ground truth. Successfully restoring the load peaks is critical for distribution circuit analysis because load peaks often represent critical operating conditions.

Comparison of FCE: From the spectrum plots in Figure 2.9, we can see that the ProfileSR-GAN model can recover more high-frequency components than LERP and MSE-based methods.

Comparison of CPE: As shown in Table 2.2 and Figure 2.9, ProfileSR-GAN achieved the best performance in critical points matching. We can see that the simplified profile of ProfileSR-GAN is still very similar to the ground truth, indicating that they have similar profile complexity. The profiles generated by MSE-based methods and LERP, by contrast, have much fewer critical points.

Comparison of the performace of the shape-wise evaluation metrics on test set: As shown in Figure 2.10, ProfileSR-GAN consistently outperforms LERP and other MSE-based models in terms of PLE, FCE, CPE while maintaining an acceptable MSE level.

2.4.3 2D Visualization of LPSR results

To intuitively compare the LPSR results from different SR methods, we select the generated daily HR profiles of a single house from April 1st to September 1st in 2015 to make the 2-D visualization of different SR model outputs, as shown in Figure 2.11. First, we apply discrete Fourier transformation to extract the frequency-domain components of the HR profiles generated by different SR methods. Then, we use t-SNE [49] to reduce the dimension to 2-D for better visualization. We can see that the 2-D representations of ProfileSR-GAN results is the closest to the ground truth area. Meanwhile, the distance between the SR result and the ground truth is measured by the Wasserstein distance. As shown in Table 2.3, ProfileSR-GAN achieves the best performance.

Table 2.3. Wasserstein distance of 2D Representation of SR results

Metrics \ SR models		LERP	ASR	SRP	CNN	ProfileSR-GAN
Wasserstein distance	X-axis	4.560	4.518	3.391	3.506	**0.679**
	Y-axis	1.170	1.136	1.089	1.066	**0.256**

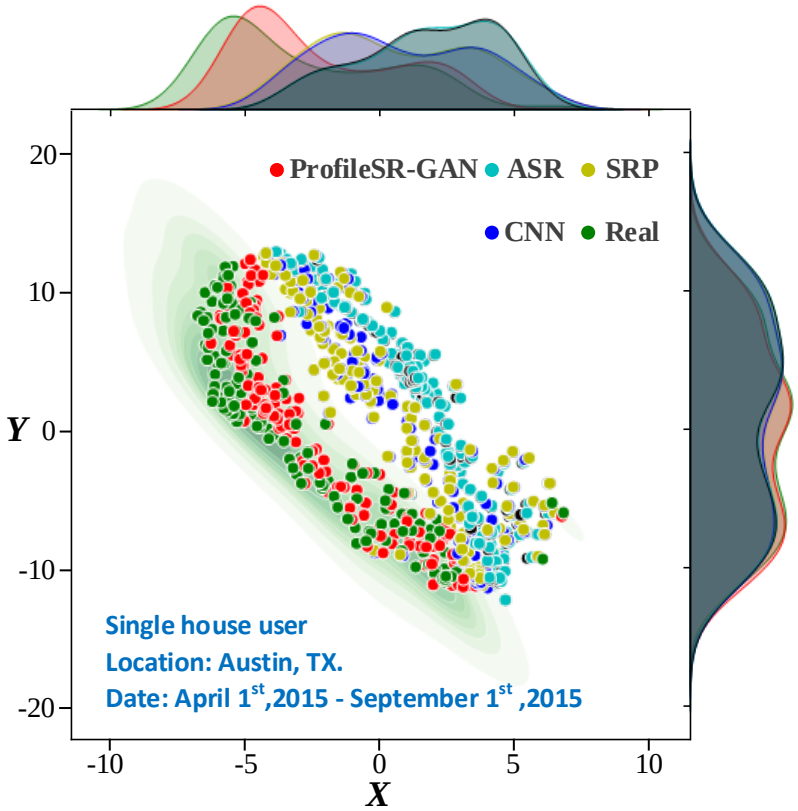


Figure 2.11. 2-D visualization of the frequency components extracted from the HR profiles generated by different SR methods

2.4.4 Performance Comparison under Different Scale-Up Factor

To evaluate the impact of the scale-up factor α, we compare the case of α=6 (i.e. from LR-30min to HR-5min) with two other cases: α=12 (i.e. from LR-60min to HR-5min) and α=3 (i.e. from LR-15min to HR-5min). We keep the same ProfileSR-GAN network structure shown in Figure 2.4 and alter only the stride of the transpose convolution layer of the generator network to cope with different α values.

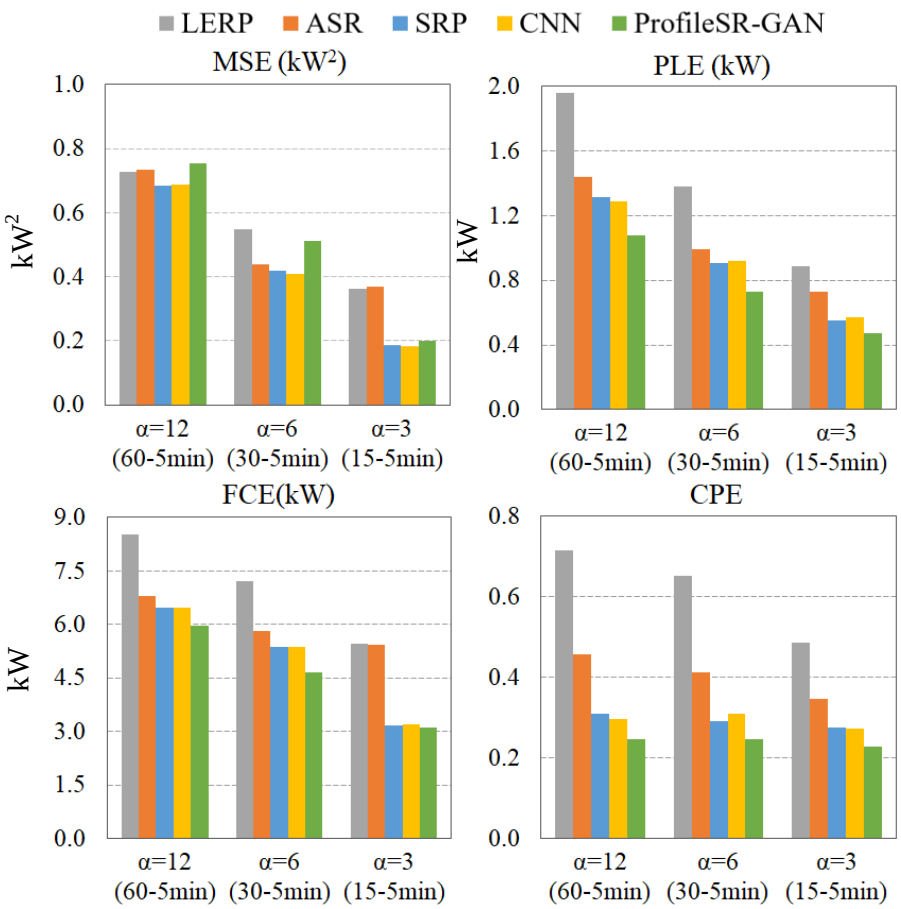


Figure 2.12. Performance evaluation for cases with different scale-up factors

As shown in Figure 2.12, when α decreases, all SR methods perform better (i.e., four error metrics tend to reduce). This is expected because a smaller α represents a less ill-posed LPSR problem, making the problem easier to solve. In most cases, the MSE-based methods still outperform the GAN-based methods in MSE. We also observe that the MSE of ProfileSR-GAN significantly reduces when α decreases, showing that the LPSR problem becomes easier to solve when there are fewer points to recover in an interval.

Moreover, compared with LERP and MES-based learning, ProfileSR-GAN is more effective for larger α (e.g., α = 12 and α = 6). Because for a large α, the LPSR problem is highly ill-posed: restoring the ground truth HR profile from the observed individual LR profile is challenging. However, the ProfileSR-GAN generator can approximate the distribution of the

realistic HR profile dataset under the guidance of the feature-matching loss and adversarial loss. Therefore, it can generate HR samples that have a better chance to be realistic.

When α is small (e.g., α=3), the distribution approximation ability of the GAN-based method becomes less essential because each individual LP profile already contains enough information to recover the new missing points in an interval, making the MSE-based model better choices.

2.4.5 Impact of Weather Data

As mentioned in Section III.A, weather data serves as part of the prior knowledge in this paper to enhance the GAN-based model performance. To assess the impact of the weather data on the performance of ProfileSR-GAN, we conduct a controlled experiment: we train two identical ProfileSR-GAN models, one with and the other without weather data. The performance metrics of the two models are summarized in Table 2.4. We can see that using the weather data as input achieves better performance in all four metrics.

Table 2.4. Performance comparison for quantifying the impact of using weather data as input

Metrics	Weather Data	α=12 (60 to 5min)	α=6 (30 to 5min)	α=3 (15 to 5min)
MSE	with	**0.782**	**0.512**	**0.201**
	without	0.793	0.524	0.213
PLE	with	**1.081**	**0.731**	**0.475**
	without	1.145	0.765	0.491
FCE	with	**5.964**	**4.652**	**3.125**
	without	5.982	4.725	3.163
CPE	with	**0.243**	**0.247**	**0.227**
	without	0.251	0.252	0.259

2.5 Experiments on Non-Intrusive Load Monitoring

NILM methods are used to disaggregate electricity consumption from the smart meter level to the appliance level by capturing the unique signatures of each appliance. Because LPSR aims at restoring the high-frequency components of a down-sampled load profile, NILM is a natural downstream task and can be used to evaluate the effectiveness of LPSR. If the high-frequency load signatures can be recovered by LPSR, the NILM algorithm should be able to achieve better performance.

As shown in Figure 2.13, the experiment includes four steps:

1) LPSR implementation. Five different LPSR methods are used, including the proposed ProfileSR-GAN and four other benchmarking methods, to upsample the 30-min aggregated LR load profiles back to 5-min HR load profiles.

2) NILM model training. The NILM models are trained using the real 5-min aggregated profiles. The outputs of the NILM are 5-min appliance level profiles. The trained NILM models can recognize appliances once provided with an aggregated load profile. In this paper, we will use three different NILM models: Denoising Autoencoder (DAE) [50], sequence to point model (Seq2Point) [51], and sequence to sequence model (Seq2Seq) [51]. These models are provided by the Non-intrusive Load Monitoring Toolkit (NILMTK) [52], which is an open-source NILM algorithm platform.

3) NILM model testing. After the NILM models are trained, they are fed with the HR profiles generated in step 1 to evaluate how well the appliance profiles can be recognized.

4) NILM result evaluation. The recognized appliance profiles in step 3 are compared with the ground truth to evaluate the NILM performance when using up-sampled load profiles by ProfileSR-GAN.

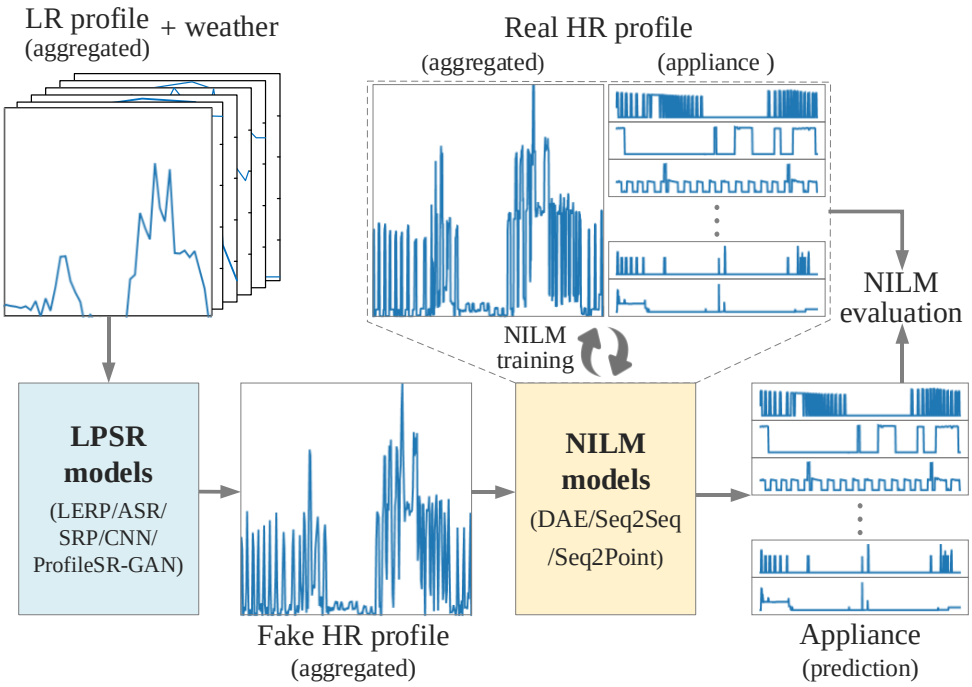


Figure 2.13. Flowchart of the NILM experiments

Pecan Street data set paired with weather data from [46] is used to support the NILM experiments. We randomly selected four residential users from the data set. Each user has five appliances, i.e., air conditioner, electric furnace, fridge, dishwasher, and microwave. The data is one-month length from August 1st to September 1st, 2015, with 1-min granularity. We down sample the original 1-min aggregated profiles and appliance level profiles to 5-min as P^{HR} and 30-min as P^{LR}. The first 20 days are used for training the NILM models, and the last 11 days for evaluation.

Two metrics are adopted to evaluate the NILM performance for each appliance. The first is the root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(y_t - \hat{y}_t\right)^2} \tag{2.20}$$

where T is the profile length, y_t is the actual power consumption of the target appliance at time t, and $\hat{y}_t$ is the corresponding NILM estimation. The second metric is the Overall Error (OE)

[53], which measures the percentage power consumption mismatch between the NILM estimation and the ground truth.

$$OE = \left| \sum_t y_t \ / \sum_t Y_t - \sum_t \hat{y}_t \ / \sum_t \hat{Y}_t \right| \tag{2.21}$$

Where Y_t denote the actual aggregated power consumption of all appliances at time *t*, and $\hat{Y}_t$ is the corresponding NILM estimation. The calculation results are shown in Table 2.5.

Table 2.5. Performance Comparison of different NILM algorithms

Appliance	**Metrics**	**Root mean square error (kW)**					**Overall error** (10^{-1})				
	Houses	**LERP**	**ASR**	**SRP**	**CNN**	**ProfileSR-GAN**	**LERP**	**ASR**	**SRP**	**CNN**	**ProfileSR-GAN**
Air-conditioner	1	1.063	1.049	1.024	1.016	0.969	0.780	0.883	0.403	0.430	0.346
	2	1.216	1.201	1.071	1.103	1.048	2.417	2.303	1.587	1.917	0.758
	3	1.348	1.350	1.134	1.199	0.922	4.949	4.925	2.620	3.004	1.102
	4	0.793	0.809	0.710	0.727	0.679	0.713	0.790	0.404	0.453	0.104
	mean	1.105	1.102	0.985	1.011	**0.905**	2.215	2.225	1.253	1.451	**0.578**
Fridge	1	0.105	0.106	0.105	0.105	0.106	0.145	0.155	0.128	0.144	0.040
	2	0.071	0.074	0.074	0.078	0.067	0.325	0.285	0.233	0.261	0.160
	3	0.102	0.104	0.089	0.088	0.084	2.703	2.712	1.258	1.539	0.482
	4	0.078	0.079	0.078	0.079	0.078	0.278	0.335	0.138	0.153	0.236
	mean	0.089	0.091	0.087	0.087	**0.084**	0.863	0.872	0.439	0.524	**0.230**
Electric furnace	1	0.118	0.116	0.106	0.106	0.090	0.329	0.362	0.162	0.138	0.130
	2	0.063	0.064	0.057	0.058	0.056	0.687	0.634	0.501	0.644	0.474
	3	0.299	0.299	0.220	0.243	0.201	0.571	0.569	0.689	0.654	0.349
	4	0.071	0.073	0.066	0.067	0.064	0.055	0.057	0.049	0.064	0.021
	mean	0.138	0.138	0.112	0.119	**0.103**	0.410	0.405	0.350	0.375	**0.244**
Dish washer	1	0.127	0.135	0.100	0.114	0.075	0.453	0.495	0.293	0.329	0.051
	2	0.364	0.365	0.321	0.333	0.224	1.378	1.362	0.832	0.990	0.151
	3	0.128	0.123	0.102	0.106	0.073	1.434	1.351	0.636	0.755	0.248
	4	0.089	0.086	0.105	0.095	0.087	0.065	0.049	0.163	0.123	0.025
	mean	0.177	0.177	0.157	0.162	**0.115**	0.832	0.814	0.481	0.549	**0.119**
Microwave	1	0.085	0.084	0.083	0.084	0.083	0.114	0.124	0.130	0.138	0.156
	2	0.023	0.022	0.022	0.022	0.021	0.023	0.018	0.018	0.020	0.010
	3	0.028	0.028	0.013	0.014	0.013	0.468	0.453	0.037	0.055	0.023
	4	0.123	0.122	0.121	0.122	0.120	0.489	0.492	0.433	0.458	0.160
	mean	0.065	0.064	0.060	0.060	**0.059**	0.273	0.272	0.154	0.168	**0.087**

As shown in Table 2.5, using ProfileSR-GAN for upsampling achieves the best performance in most cases. This is because NILM algorithms rely heavily on capturing the load switching signature in the aggregated load profile (e.g., the rising and falling edges, the spikes), which are usually caused by the appliance ON/OFF or cycling activities. Thus, by restoring the high-frequency waveforms, ProfileSR-GAN makes it easier for NILM to identify appliance-level load profiles and energy consumption patterns. The MSE-based SR methods produce over-

smoothed HR profiles, which provides fewer waveform signatures for NILM to capture. These results further demonstrate the value of the proposed ProfileSR-GAN model. Figure 2.14 shows the NILM results for the air conditioner identification as an illustration of the results.

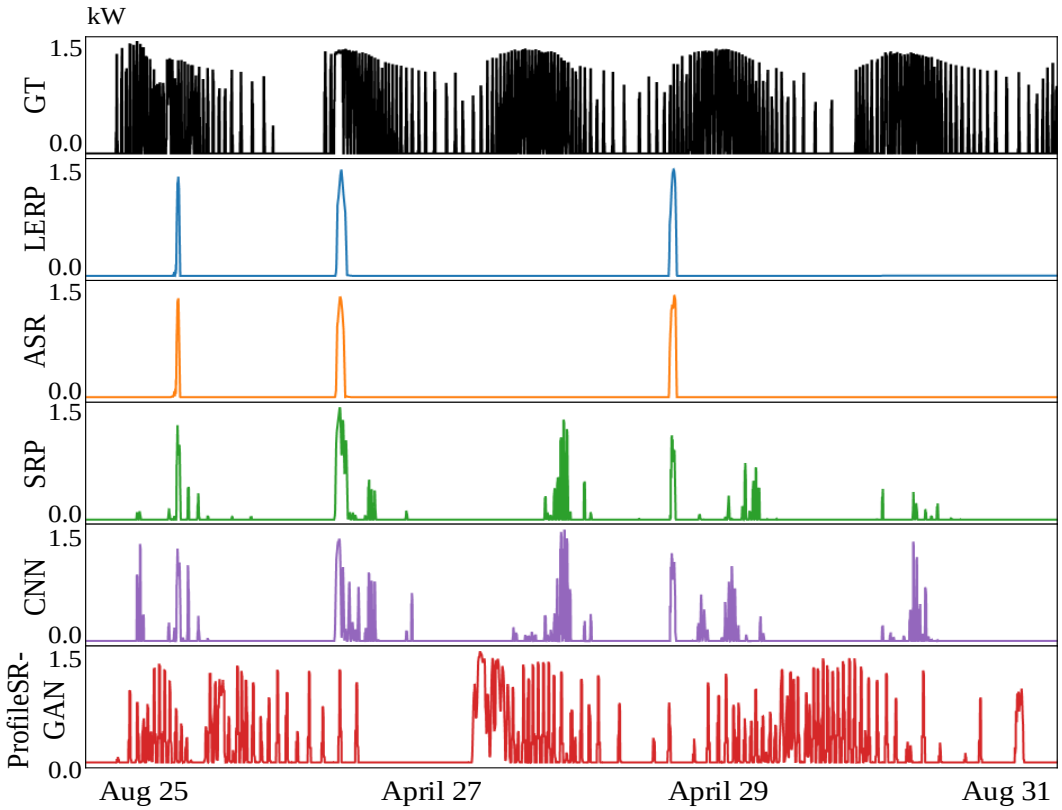


Figure 2.14. NILM results comparison for air conditioner based on Seq2Seq algorithm

2.6 Conclusion

In this chapter, we propose ProfileSR-GAN, a 2-stage GAN-based method for solving LPSR problems. In the first stage, a GAN-based model is trained to restore the high-frequency components from the low-resolution data. In the second stage, a polishing network is developed to remove unrealistic power fluctuations in the GAN generated high-resolution load profiles. Compared with conventional up-sampling methods, such as interpolation and CNN-based methods, the proposed ProfileSR-GAN achieves superior performance in restoring high-frequency components inside sampling intervals. The overall performance improvements

attribute to three aspects: the adversarial training of the GAN-based model, the inclusion of weather data, and the fine-tuning of the polishing network.

The simulation results demonstrate that ProfileSR-GAN achieved 36%-62% improvements in shape-related evaluation metrics compared with the baseline method (i.e., the linear interpolation method). An application of ProfileSR-GAN is presented as a case study to demonstrate that applying ProfileSR-GAN on upsampling can benefit downstream tasks that require the use of high-resolution load profiles. Simulation results show that when using ProfileSR-GAN to up sample the low-resolution profiles before conducting NILM, appliance-level activities can be better recognized by the NILM algorithms.

CHAPTER 3 LOAD PROFILE INPAINTING FOR BASELINE ESTIMATION

3.1 Background

Missing data is a common issue in distribution system load profile processing. Oftentimes, missing data are caused by temporarily lost communication with the equipment. Statistics show that approximately 70% of missing data segments in power distribution systems are less than 4 hours. Thus, restoring short missing data segments is critical for improving the data utilization rate and providing high-quality data sets to down-stream data processing tasks.

Moreover, algorithms used for restoring missing data can also be used to identify the operational baseline of demand response (DR) programs. For example, Conservation Voltage Reduction (CVR) is widely used by utilities for peak load reduction [54]. During a CVR event, system voltage at the substation bus will be reduced by 2-4% to achieve load reduction. To quantify the CVR caused load reduction, it is very important for utility engineers to accurately estimate what the original load profile (i.e., the baseline) during a CVR event would have been had the bus voltage not been reduced. Because the pre- and post- CVR load profiles representing customer consumption under normal system voltage, uncovering the CVR baseline is equivalent to restoring missing data in the CVR period. Because baseline estimation is essential to DR performance evaluation, inpainting the would-have-been load profile during a DR event is highly valuable to load service providers.

Existing missing data restoration methods for load profile inpainting are categorized into model-based and data-driven. Model-based methods use physical system models to simulation responses to external disturbances in hope of restoring missing data segments. For example, to estimate the CVR baseline, researchers use the distribution system topology and load models to predict load changes when system voltage changes [55-59]. In general, model-based methods

require accurate distribution system models. However, in practice, distribution system models are incomplete and inaccurate due to topology changes and lack of measurements. Moreover, load composition varies with respect to the time-of-the-day while customer consumption patterns shift constantly due to occupancy and weather conditions. Therefore, it is often times infeasible for utilities to apply model-based methods for restoring missing data segments.

Thus, the data-driven approach is the dominant approach for missing data restoration. Data-driven methods can be further categorized into three approaches: similarity-based, regression-based, and generative-based. The similarity-based approach groups load profiles by day type, weather conditions, and similarity among load profiles. The missing data segments are restored by referencing to the data on the load profiles having the best similarity match. Similarity-based methods are straightforward, easy to implement and explainable, therefore are widely used in field implementation[59-61]. However, in many cases, similarity metrics are normally defined by human analysts and can be based on the weighted average of many factors (e.g., weather, time, geographical conditions, and load types). This makes the accuracy of the method a dependent on subjective selections of similarity metrics and weights.

Regression-based methods include linear regression [61], Long Short Term Memory (LSTM) [31], Stacked Autoencoder (SAE) [32] , Gaussian Regression [23], and Support Vector Regression (SVR) [20, 62]. Regression-based methods usually achieve higher estimation accuracy compared to the similar day approach because of their nonlinear learning capabilities, especially when using deep-learning models. However, compared with similarity-based methods, the deep-learning based methods are less explainable and having higher computing costs. Thus, in recent years, hybrid solutions combining multiple regression models [24, 26, 27] are proposed for baseline estimation or missing data restoration.

The main drawbacks of the existing regression-based approaches is that the restored data segments need to have the same fixed length. In practice, the duration of missing data varies from minutes to several hours. To cope with variable missing data length, many existing methods either increase the output window to cover the longest event or train separate models for different missing data length. This inevitably increases model complexity with added computing and deployment cost.

In this paper, we propose a third approach for missing data restoration: the Generative Adversarial Nets (GAN) based approach. Studies of using GAN to solve the missing data restoration problem is still in infancy. In [63, 64], the authors discuss the basic theory of GAN in restoring missing data, while in [65, 66], the authors implement a GAN-based method in power domain to restore the grid measurement data and the PV profiles. Inspired by image inpainting, we develop a Load Profile Inpainting Network (Load-PIN) using GAN [12] as the basic structure of a highly accurate and flexible missing data restoration framework for recovering missing data on load profiles. The framework, with little fine-tuning, can be readily applied for DR baseline estimation. The generator consists of a coarse network and a fine-tuning network. Initially, the bidirectional time series load data before and after the missing data segment together with the explanatory variables are fed into the coarse network to obtain an initial estimation for the missing part. Next, initial estimations are sent to the fine-tuning network consisting of Gated Convolution layers [33] and Multi-head self-attention blocks [34] to improve accuracy. The generator network is trained under the guidance of the discriminator with specially designed loss functions.

The rest of the chapter are organized as follows: Section 3.2 formulates the CVR baseload estimation problem, Section 3.3 introduces the proposed Load-PIN model, Section 3.4 demonstrates the case study results, and Section 3.5 concludes this paper.

3.2 Customer Baseline Load Estimation Problem Formulation

In this section, we first illustrate the problem formulation of load profile inpainting and the background of the GAN based approach. Then, we present the proposed 2-stage GAN generator structure and the loss function design of the Load-Load-PIN framework.

Denote a historical time series matrix of load as $\mathbf{Y} = [y_1, y_2, \dots, y_L]$, where L is the length of the time series. Denote the explanatory variables $\mathbf{X}$ as

$$X = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_L^1 \\ x_1^2 & x_2^2 & \cdots & x_L^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^E & x_2^E & \cdots & x_L^E \end{bmatrix} \tag{3.1}$$

where E is the number of explanatory variables.

Define an event to be a missing data segment (see Fig. 3.1(a)) or an unknown DR baseline (see Fig. 3.1(b)). Assume there are N events $\mathbf{Y}$ and the duration of the i^{th} event (i.e. the i^{th} inpainting period) is T_{event}^{i}. The inpainting duration vector, $\mathbf{T}_{event}$, is

$$\mathbf{T}_{event} = [T_{event}^{1}, T_{event}^{2}, \dots, T_{event}^{N}] \tag{3.2}$$

while the i^{th} restored data segment, $\widehat{\mathbf{Y}}_{event}^{i}$, is

$$\widehat{\mathbf{Y}}_{event}^{i} = \left[\hat{y}_1^i, \hat{y}_2^i, \dots, \hat{y}_{T_{event}^{i}}^{i}\right] \tag{3.3}$$

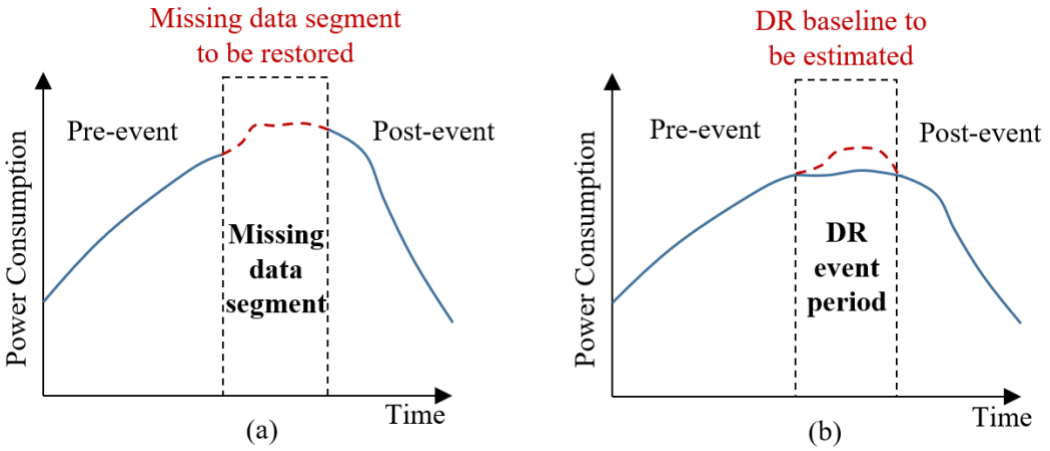


Figure 3.1. Two basic event types: (a) missing data restoration, and (b) DR baseline identification. Blue solid lines are field measurements and red dotted lines are uncovered data segments

As shown in Fig. 3.2, we divide the load profile containing the ith event into three periods: $[\mathbf{X}^{i}_{event}, \mathbf{Y}^{i}_{event}]$ as the inpainting data period, $[\mathbf{X}^{i}_{pre}, \mathbf{Y}^{i}_{pre}]$ as the pre-event period, and $[\mathbf{X}^{i}_{post}, \mathbf{Y}^{i}_{post}]$ as the post-event period.

Thus, a load profile inpainting problem can be described as

$$\widehat{\mathbf{Y}}^{i}_{event} = f_{\theta}\left(\mathbf{X}^{i}_{pre}, \mathbf{Y}^{i}_{pre}, \mathbf{X}^{i}_{event}, \mathbf{X}^{i}_{post}, \mathbf{Y}^{i}_{post}\right) \tag{3.4}$$

where f_{θ} is the mapping function.

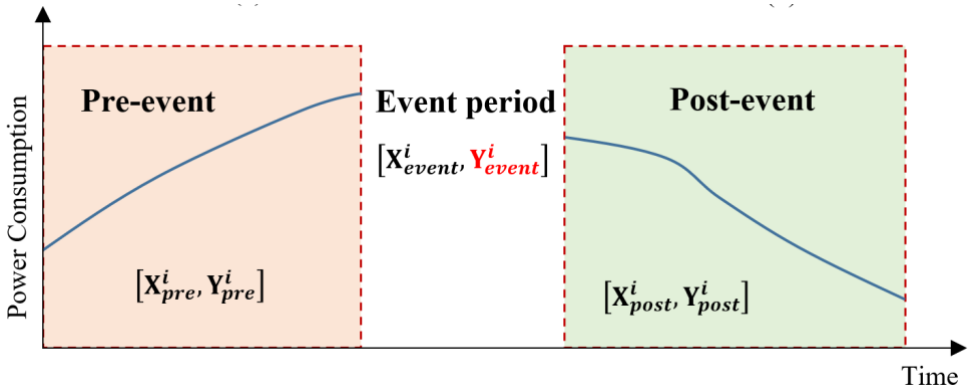


Figure 3.2. An illustration of the division of the load profile containing an event

In the similarity-based approach, pre- and post- event data are mainly used for identifying similar days. Traditional forecasting-based methods use only pre-event data, $\mathbf{X}^{i}_{pre}$ and $\mathbf{Y}^{i}_{pre}$, to model inputs to forecast $\widehat{\mathbf{Y}}^{i}_{event}$. Most correlation-based methods take $\mathbf{X}^{i}_{event}$ as inputs to build

the mapping function between **Y** and **X** (e.g., temperature and voltage) for estimating $\widehat{\mathbf{Y}}_{event}^{i}$. Thus, one of the main drawbacks of the state-of-the-art methods is that the information contained in all five available data sets (i.e., $[\mathbf{X}_{pre}^{i}, \mathbf{Y}_{pre}^{i}, \mathbf{X}_{event}^{i}, \mathbf{X}_{post}^{i}, \mathbf{Y}_{post}^{i}]$) have not yet been fully utilized. To resolve this deficiency, we take the GAN based approach to use all 5 available data sets as inputs to f_θ for predicting $\widehat{\mathbf{Y}}_{event}^{i}$.

3.3 GAN-Based Approach Implementation

As shown in Fig. 2.2, a GAN model consists of a generator network (*G*) and a discriminator network (*D*). The input of the generator is a latent vector **z**, often a Gaussian noise. The generated data, $G(\mathbf{z})$, along with the actual data, **x**, are then passed to discriminator *D*. The goal of *D* is to distinguish real data sets from the fake ones. The training of a GAN model is an iterative, adversarial process: *G* tries to generate samples $G(\mathbf{z})$ to fool *D*; *D* learns to identify $G(\mathbf{z})$ from **x** by assigning greater probabilities to **x** and smaller ones to $G(\mathbf{z})$. As introduced in [12], this process is formulated as a minimax game

$$\min_{G}\max_{D} R(D,G) = \mathrm{E}_{\mathbf{x}\sim p_{(\mathbf{x})}}[\log D(\mathbf{x})] + \mathrm{E}_{\mathbf{z}\sim p_{(\mathbf{z})}}[\log(1-D(G(\mathbf{z})))] \tag{3.5}$$

where $R(D,G)$ is the reward function, $p(\mathbf{x})$ and $p(\mathbf{z})$ are the probability distributions of training data and latent vector, E is the expectation operator.

The Load-PIN framework is illustrated in Fig. 3.4. The model input **z** has three parts: 24-hour load and temperature profiles, and a Boolean mask indicating the event period as one and the normal period as zero. The load data resolution varies from 1-minute to 15-minute and the missing data duration, T_{event}, is less than 4 hours. The generator contains two stages: a coarse network for initial estimation and a fine-tuning network for polishing. The discriminator is a deep convolutional network with specially designed loss functions.

3.3.1 Training Sample Generation

The goal of the sample generation process is to generate samples evenly distributed in yearly load profiles so that the trained model will not be biased by factors such as the time-of-the day and season-of-the-year. To achieve this goal, we slide a 24-hour moving window over yearly historical load profiles, as shown in Fig. 3.3. In this paper, the time shift ($\triangle t$) of the moving window is one hour.

There are two considerations. *First*, to train and test the Load-PIN model, the training and testing samples should be generated from load profiles containing no CVR events. This is because, during a CVR event, system voltages are reduced by 2-4% for load reduction, making the load profile under the normal system voltage (the ground truth) unknown to the analyst. *Second*, the missing data segment will be positioned at the center of the 24-hour window so that the pre- and post- event data are equal in length. *Third*

After the Load-PIN model is trained, we apply it to CVR samples for CVR baseline identification. A CVR sample is a 24-hour load profile with the CVR event being placed in the middle of the 24-hour window (See Fig. 3.3).

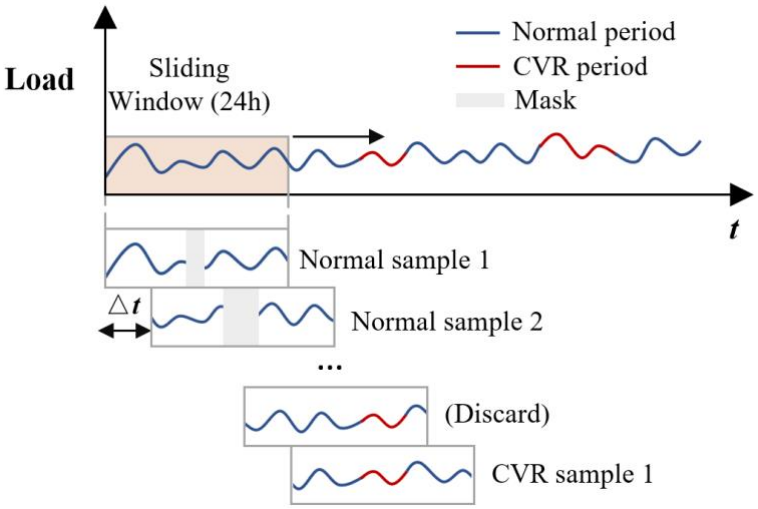


Figure 3.3. Training sample generation process

Denote N_{train} , N_{test}, and N_{CVR} as the total number of training samples, testing samples, and CVR samples, respectively. For the i^{th} normal sample, a mask of variable length (but less than 4-hour) is placed in the middle of the 24-hour window. The data being masked is the ground truth ($\mathbf{Y}_{event}^{i}$) of the forecasted missing data segment ($\widehat{\mathbf{Y}}_{event}^{i}$). Before and after the mask are the pre-event and post-event load segments (i.e., $\mathbf{Y}_{\boldsymbol{pre}}^{\boldsymbol{i}}, \mathbf{Y}_{\boldsymbol{post}}^{\boldsymbol{i}}$). Meanwhile, load segments are paired with their corresponding temperature profiles as the explanatory variable (i.e.,$\mathbf{X}_{\boldsymbol{pre}}^{\boldsymbol{i}}, \mathbf{X}_{\boldsymbol{event}}^{\boldsymbol{i}}, \mathbf{X}_{\boldsymbol{post}}^{\boldsymbol{i}}$). After $\widehat{\mathbf{Y}}_{event}^{i}$ is estimated using (3.4), we can optimize the model parameters by

$$\theta^{*} = \arg\min_{\theta} \sum_{i=1}^{N_{train}} \left\| \mathbf{Y}_{event}^{i} - \widehat{\mathbf{Y}}_{event}^{i} \right\|_{2}^{2} \tag{3.6}$$

3.3.2 Generator Network: 1st-Stage Coarse Network

In the first stage, we employ the Gated Convolution Network (GCN) [33] to formulate an encoder-decoder structure to restore the masked segments. Compared with conventional convolution neural network (CNN), GCN adds an additional feature-wise gating control mechanism, which learns soft masks automatically to compute the hidden layer features as

$$h_l(\mathbf{X}) = \varphi(\mathbf{X}\cdot\mathbf{W} + \mathbf{b}) \otimes \sigma(\mathbf{X}\cdot\mathbf{U} + \mathbf{c}) \tag{3.7}$$

where $\mathbf{X}$ is the input of layer h_l, $\mathbf{W}$, $\mathbf{b}$, $\mathbf{U}$, $\mathbf{c}$ are trainable parameters. σ is the sigmoid function, φ can be any activation function (e.g., ReLU and LeakyReLU), and $\otimes$ is the element-wise product between matrices. This equation demonstrates that different weights of filters are applied at different temporal points to produce the output, resulting in a dynamic feature selection mechanism for each channel and each time point. Besides, Gated Transpose Convolution Network (GTCN) [67] layers are introduced to recover the estimated daily profiles from the GCN output.

The 1st-stage coarse network is trained purely based on the point-to-point content loss function, L_{coarse}.

$$L_{coarse} = \frac{1}{H} \left\| G_{\theta_{G1}}(\mathbf{z}) - \mathbf{P} \right\|_2^2 \tag{3.8}$$

where θ_{G1} is the parameter of the coarse generator network, **P** is the ground truth load profile, and *H* is the dimension of **P**.

Note that instead of computing the point-to-point content loss (L_{coarse}) for the entire 24-hour period, we calculate only the content loss for the masked segment plus a few points before and after the masked segment, the length of which is *H*. For example, in this paper, *H* is set as 5 hours to cover the masked segment with a minimum margin of 0.5 hour. This ensures that the coarse network focuses on the masked period to enhance accuracy and achieve smoother transition between non-event periods and the event period (i.e., no spikes during the transitions).

3.3.3 Generator Network: 2nd-Stage Fine-Tuning Network

In the second stage, we make two modifications to the conventional GAN framework: *adding Multi-head self-attention* and *considering the tradeoff between content loss, feature matching loss and adversarial loss in the loss function* to polish the first-stage results and recover realistic details (high-frequency components).

Self-attention, also known as intra-attention, is a technique for focusing attention on various points in a single sequence when creating a representation of the sequence. An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the value as

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = softmax(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\alpha})\mathbf{V} \tag{3.9}$$

where **Q**, **K**, and **V** include learnable parameters representing query-key-value pairs, respectively, α is a scaling factor. Instead of performing a single attention function, [34] founds it

beneficial to linearly project the queries, keys, and values multiple times with different learned linear projections in parallel, which is multi-head attention formulated as

$$\begin{aligned} MultiHead(\mathbf{Q},\mathbf{K},\mathbf{V}) &= concate(head_1, head_2, ..., head_n)\mathbf{W}^{\mathbf{O}} \\ head_i &= Attention(\mathbf{Q}\mathbf{W}_i^{\mathbf{Q}}, \mathbf{K}\mathbf{W}_i^{\mathbf{K}}, \mathbf{V}\mathbf{W}_i^{\mathbf{V}}) \end{aligned} \tag{3.10}$$

where $\mathbf{W}_i^{\mathbf{Q}}$, $\mathbf{W}_i^{\mathbf{K}}$, $\mathbf{W}_i^{\mathbf{V}}$ and $\mathbf{W}^{\mathbf{O}}$ are the learnable parameter matrices of the projection. This allows the temporal dependencies to be modeled without considering their distances in the input or output sequences [68].

The loss function of the fine-tuning network (L_{refine}) includes 3 terms: the content loss, the adversarial loss (L_{adv}) and the feature-matching loss (L_{feat}), as shown in (3.11)-(3.13). λ_1 and λ_2 are the weights. Same as in L_{coarse}, the content loss minimizes point-to-point errors. L_{adv} improves the realisticness of the estimation results by maximizing the scores of the discriminator, where M is the dimension of the discriminator output (shown as the 3D blue matrix in Fig. 3.4). L_{feat} is defined in (3.13) as the distance between high-level feature maps extracted from the hidden layers of the discriminator network, where $\varphi_j(\cdot)$ represents the output of the j^{th} intermediate convolution layer of the discriminator network. J is the number of intermediate layers in the discriminator network. Because high-level features of real load profiles are embedded in the hidden layer outputs, L_{feat} guides the fine-tuning network to generate more realistic results by matching those high-level features extracted from real profiles. Similar to L_{coarse}, all 3 loss terms in L_{refine} are calculated using the 5-hour segment instead of the 24-hour load profile.

$$L_{refine} = \frac{1}{H}\left\| G_{\theta_{G_2}}(\boldsymbol{z}) - \boldsymbol{P} \right\|_2^2 + \lambda_1 L_{adv} + \lambda_2 L_{feat} \tag{3.11}$$

$$L_{adv} = -\frac{1}{M} D\left(G_{\theta_{G_2}}(\boldsymbol{z}) \right) \tag{3.12}$$

$$L_{feat} = \sum_{j=1}^{J} \left\| \varphi_j \left(G_{\theta_{G_2}}(\boldsymbol{z}) \right) - \varphi_j(\boldsymbol{P}) \right\|_2^2 \tag{3.13}$$

3.3.4 Discriminator Network

The discriminator network is trained to solve the maximization problem defined by (3.5). In practice, an event can happen at any time of the day with varying lengths. To help the discriminator focus on the event duration, we also sent the corresponding Boolean mask together with the load profile. The discriminator contains five convolutional layers with an increasing number of kernels. This allows us to compress the input profiles into high-level feature matrix, in which each element can cover the entire input load profile. Finally, the adversarial loss is applied to each neural to identify the fake and real inputs.

Inspired by [67], we adopt spectral normalization and hinge loss to stabilize the training process of the discriminator by minimizing the loss function L_D calculated as

$$L_D = \frac{1}{M} ReLU\left(1 - D_{\theta_D}(\boldsymbol{P})\right) + \frac{1}{M} ReLU\left(1 + D_{\theta_D}\left(G(\boldsymbol{z})\right)\right) \tag{3.14}$$

where θ_D is the parameters of the discriminator networks.

3.3.5 Model Performance Evaluation

Three performance metrics are calculated: normalized Root Mean Squared Error (nRMSE), Energy Error (EE), and bia.

$$nRMSE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \sqrt{T_{event}^i \left\| \mathbf{Y}_{event}^i - \mathbf{Y}_{event}^i \right\|_2^2} \Big/ \left\| \mathbf{Y}_{event}^i \right\|_1 \tag{3.15}$$

$$EE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left| \sum_{t=1}^{T_{event}^i} (y_t^i - \hat{y}_t^i) \right| \Big/ \left\| \mathbf{Y}_{event}^i \right\|_1 \tag{3.16}$$

$$bias = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \left(\frac{1}{T_{event}^i} \sum_{t=1}^{T_{event}^i} \frac{y_t^i - \hat{y}_t^i}{y_t^i} \right) \times 100\% \tag{3.17}$$

where nRMSE evaluates the expected point-to-point error; EE evaluates the expected accumulated energy error; $bias$ reflects whether the model has a consistent difference between the actual and the missing data.

3.3.6 CVR Efficacy Estimation

To evaluate the performance of a CVR program containing N_{CVR} CVR events, the trained model is used to estimate the CVR baseline for each CVR sample so the expected load reduction when executing CVR can be calculated.

The raw average, normalized load reduction of the i^{th} CVR event is

$$CVR_{raw}^{i} = \frac{1}{T_{event}^{i}} \sum_{t=1}^{T_{event}^{i}} \frac{y_t^i - \hat{y}_t^i}{y_t^i} \times 100\% \tag{3.18}$$

The net average, normalized load reduction of the i^{th} CVR event considering the forecasting bias is

$$CVR_{net}^{i} = CVR_{raw}^{i} - bias \tag{3.19}$$

When calculating the forecasting bias in CVR baseline estimation, only the testing samples in the same season and having missing data during the CVR event periods are selected, i.e., the bias reflects the consistent difference between the actual and the missing data in CVR periods only.

If CVR_{net} is negative, load is reduced during the CVR period. If CVR_{net} is positive, load is increased during the CVR periods, making the feeder unfit for CVR.

The average CVR factor for all CVR events is

$$CVR_f = \frac{1}{N_{CVR}} \sum_{i=1}^{N_{CVR}} CVR_{net}^{i} \Big/ \Delta V^{i} \tag{3.20}$$

where ΔV^{i} is the voltage reduction ratio during the i^{th} CVR event. Note that the CVR factor can be used by utilities to identify suitable feeders for CVR.

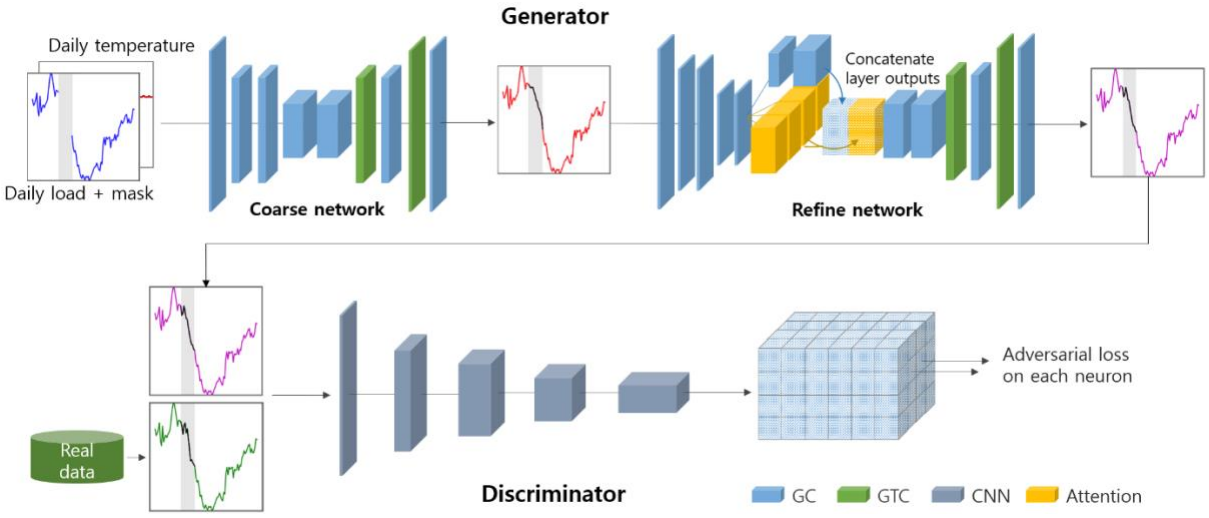


Figure 3.4. The proposed Load-PIN framework. "GC" refers to a gated convolution block, "GTC" refers to a gated transpose convolution block, "CNN" refers to a convolutional block, and "Attention" refers to a self-attention block. "ks" means kernel size, "kn" means number of kernels, and "st" means stride

Table 3.1. Network parameter of proposed Load-PIN model

Generator	Coarse	Layer	gc	gc	gc	gc	gc	gtc	gc	gtc	gc				
		ks	5	4	3	4	3	3	3	3	3				
		kn	64	128	128	256	256	128	128	64	1				
		st	1	2	1	2	1	2	1	2	1				
	Refine	Layer	gc	gc	gc	gc	gc	gc	gcn	attn*4	gcn*2	gtcn	gcn	gtcn	gcn
		ks	5	4	3	4	3	3	3	3	3	3	3	3	3
		kn	64	64	64	64	64	128	256	64	1	2	128	2	128
		st	1	2	1	2	1	1	1	1	256	128	1	128	1
Discriminator		Layer	cnn	cnn	cnn	cnn	cnn								
		ks	4	4	4	4	4								
		kn	16	32	64	128	256								
		st	2	2	2	2	2								

3.4 Case Study

In this paper, as shown in Table 3.2, three test cases are set up to demonstrate the efficacy of the proposed method: Base case for performance benchmarking, fixed-duration CVR case, and variable-duration CVR case.

Table 3.2. Test Case Descriptions

Case	Description	Source	Resolution	Data Length	Data Size	CVR Events
1	Base	PECAN Street	1-minute	1-year	318 residential users	None
2	Fixed duration	Utility A	15-minute	2-year	3 feeders	24
3	Variable duration	Utility B	5-minue	1-year	1 substation	33

To benchmark LoadPIN, we compare its performance with five other deep learning models: Multi-layer Perceptron (MLP) [69], Long-short Term Memory (LSTM) [70], Temporal Convolutional Net (TCN) [71], Bi-LSTM (Bidirectional LSTM) [72], and SAE (Stacked Auto-Encoder) [73].

Using only the pre-event data (i.e., X_pre and Y_pre) as inputs, MLP, TCN, and LSTM formulate missing data restoration and CVR baseline identification as a load forecasting problem. Bi-LSTM and SAE are bi-direction models. Thus, the inputs for the Bi-LSTM, SAE LoadPIN models are the same. Model hyper-parameters are selected based on the trial-and-error method.

In each case, all models are trained (70%), validated (15%) and tested (15%) on non-CVR samples. For the two CVR cases, the trained model is applied on the CVR samples for CVR baseline identification. The baselines are then used for computing CVR factors, which will be used to assess the efficacy of the CVR program.

3.4.1 Performance Evaluation for Missing Data Restoration

As shown in Table 3.3, in the base case, PECAN Street data [45] is used for performance benchmarking on missing data restoration. To test the impact of data resolution on estimation accuracy, 1-minute data are downsampled to 5-min, 15-min, 30-min and 1-h. To benchmark the model performance at different load aggregation levels, we test on aggregated load profiles of 10 users, 50 users, 100 users, 200 users and 300 users. After combining 5 data resolutions and 5 load levels, we obtain 25 test scenarios in total. The mask length is set to be 3-hour.

From the results, the following observations are made:

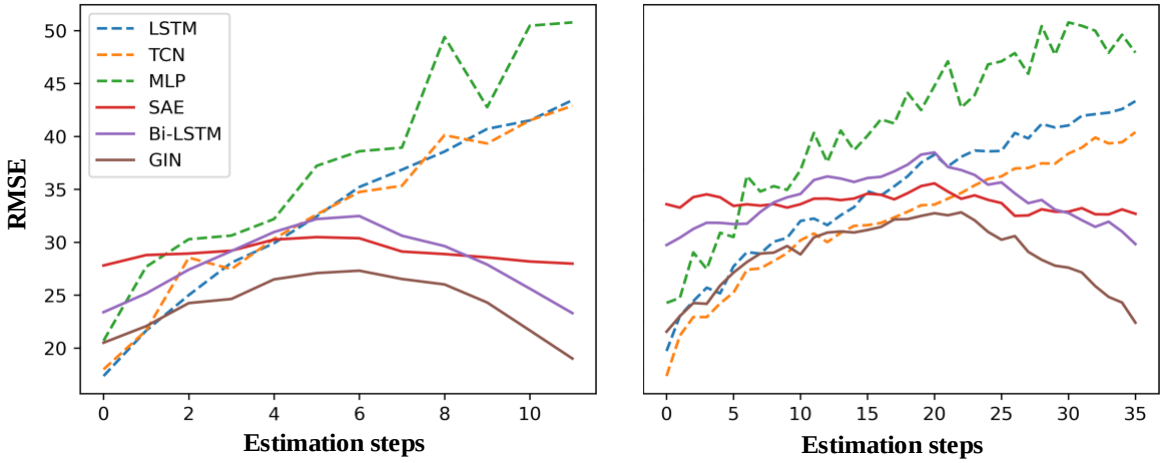
- As shown in Table 3.3, the three bidirectional models (i.e., Bi-LSTM, SAE, and Load-PIN) outperform the one-directional forecasting models in most cases. As

shown in Fig. 3.5, the bidirectional models (solid lines) have lower RMSEs than the one-directional models (dashed lines), especially in the second half of the 3-hour estimation window. The results clearly demonstrate that, by adding post-event data into the input, the accuracy will improve significantly. The step-by-step RMSEs distributions shown in Fig. 3.6 further confirm the above observation.

- As shown in Table 3.3, when data granularity is 5-min and 15-min, Load-PIN outperforms all other models and shows 15-30% improvement compared with the second-best model. This shows that LoadPIN can extract information hidden inside the high-resolution data for forecasting the missing data segments. However, if the data resolution is too low, the LoadPIN does not show significant performance improvements. This is because in those cases, forecasting average values outweigh uncovering load shape details.
- As shown in Fig. 3.7, when the load aggregation level increases from 10 to 300 users, both $nRMSE$ and EE decrease. As shown in Fig. 3.7(a), to achieve a smaller $nRMSE$, the best data granularity (highlighted by yellow rectangles) for a group with less than 100 users is 1-h and for 300 users is 15-min. This is because when load aggregation level increases, the load profile becomes smoother. Thus, it is possible to use higher data granularity for restoring more detailed load shapes, which, in turn, improves the point-to-point accuracy. According to Fig. 3.7(b), to achieve a smaller EE, the best data granularity (highlighted by yellow rectangles) is 15-minute.

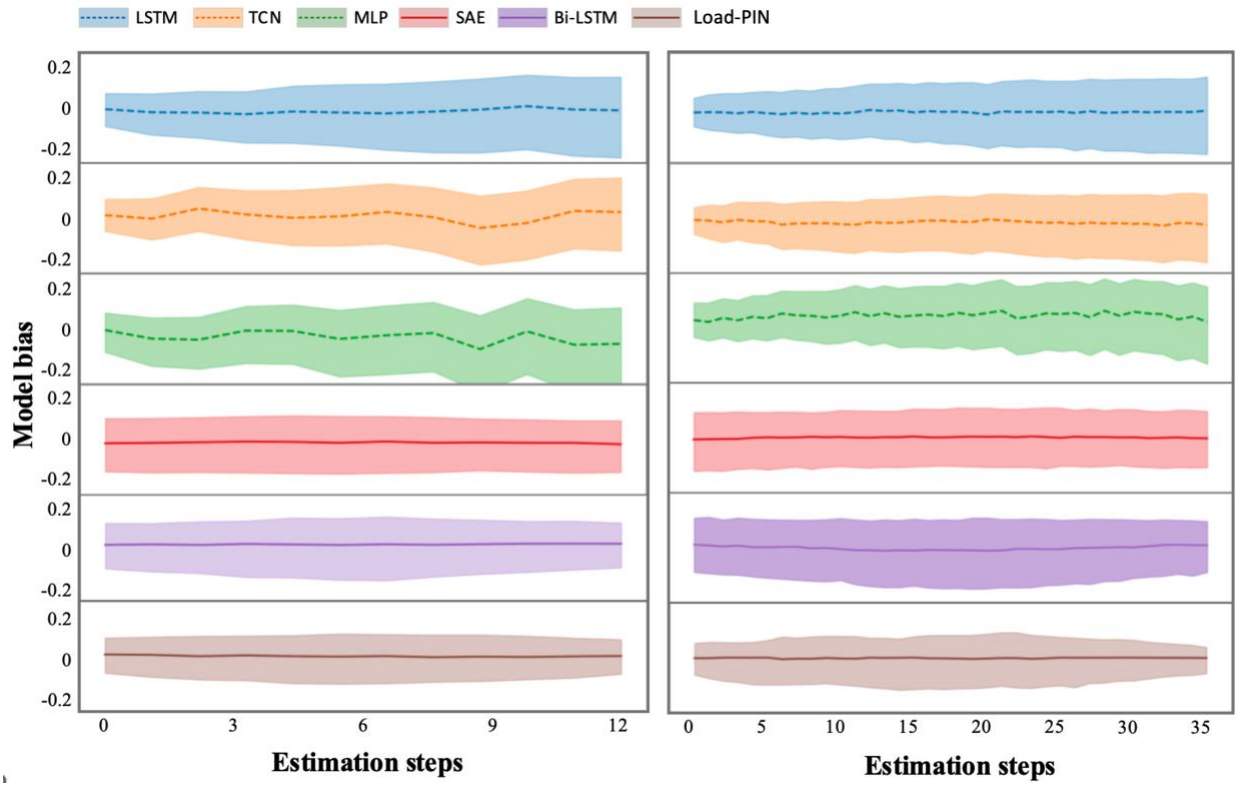
- As shown in Fig. 3.8, Load-PIN performance is consistent across different hours of the day and in different seasons. Note that the error is substantially smaller for summer peak load periods (i.e. between 15:00-18:00 in summer months), which is highly desirable for CVR baseline identification.

To summarize, benchmark tests show that Load-PIN outperforms all other algorithms in missing data restoration. Higher resolution data is preferred when there are more users in the group. In most cases, 5 or 15 minutes data set are sufficient for restoring a missing data segment with a duration of 4-hour or less.



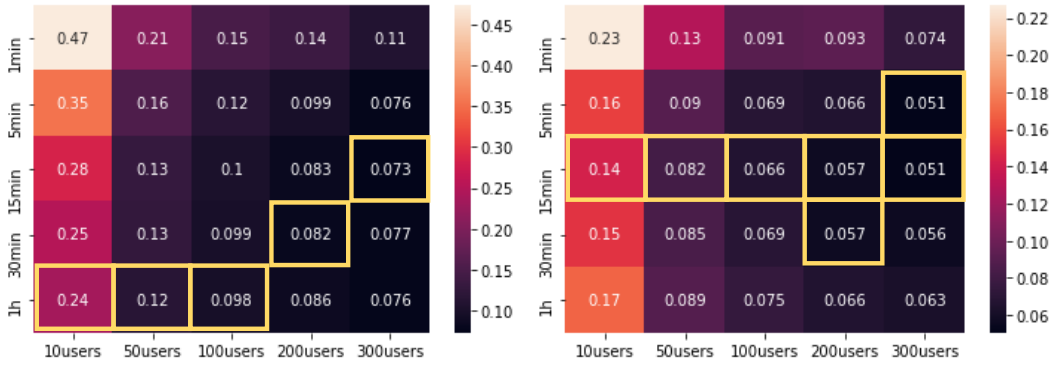
(a) (b)

Figure 3.5. LoadPIN performance evaluation: averaged step-by-step RMSE of the 6 models during 3-hour estimation interval. Test on the aggregated load of 300 users. (a) 15-min data granularity, (b) 5-min data granularity



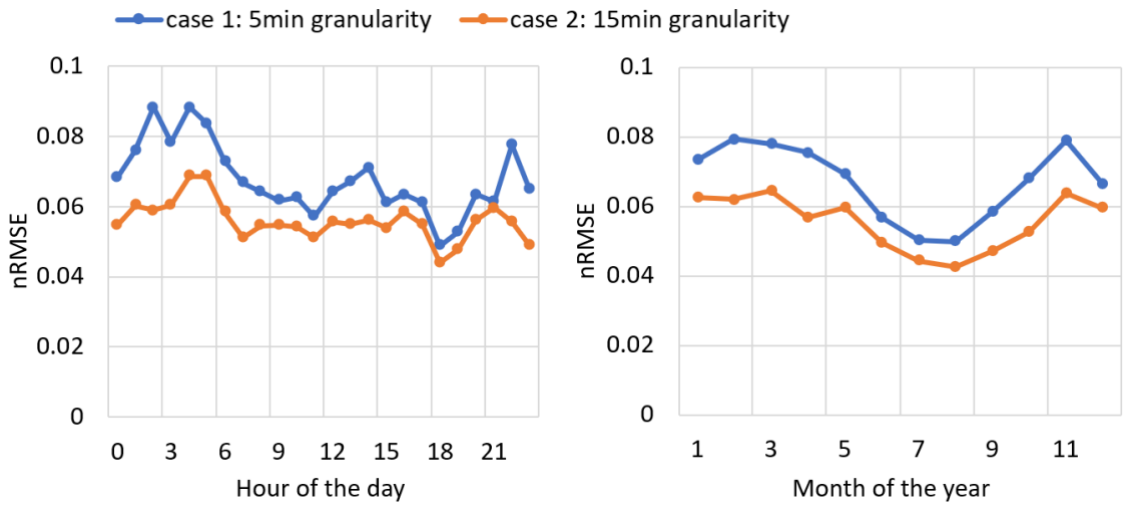
(a) (b)

Figure 3.6. Averaged step-by-step RMSE and the 90% confidence intervals of the 6 models during 3-hour estimation period. Test on the aggregated load of 300 users. (a) 15-min data granularity, (b) 5-min data granularity



(a) (b)

Figure 3.7. Impact of data resolution on missing data restoration. (a) nRMSE, (b) EE. Note the value is an averaged of all 6 deep learning models. The yellow rectangles highlight the smallest error under each aggregation level



(a) (b)

Figure 3.8. Performance of Load-PIN on (a) different hours of the day, (b) months of the year

Table 3.3. Model performances on the Pecan Street Test Case

data granularity	aggregation level	nRMSE							EE						
		LSTM	TCN	MLP	SAE	Bi-LSTM	Load-PIN	improvement	LSTM	TCN	MLP	SAE	LSTM	Load-PIN	improvement
1-min	10users	0.46	0.43	0.39	0.35	0.57	0.35	-0.85%	0.22	0.21	0.16	0.15	0.26	0.15	-1.99%
	50users	0.25	0.21	0.19	0.15	0.22	0.12	18.90%	0.17	0.13	0.11	0.08	0.13	0.06	23.35%
	100users	0.19	0.17	0.15	0.11	0.16	0.10	7.79%	0.13	0.10	0.09	0.06	0.10	0.06	4.48%
	200users	0.16	0.15	0.12	0.08	0.12	0.07	15.74%	0.12	0.10	0.06	0.05	0.09	0.05	4.14%
	300users	0.15	0.13	0.10	0.07	0.11	0.09	-25.30%	0.12	0.09	0.06	0.05	0.08	0.05	-12.78%
	mean	0.24	0.22	0.19	0.15	0.24	0.15	3.26%	0.15	0.13	0.10	0.08	0.13	0.08	3.44%
5-min	10users	0.39	0.34	0.35	0.32	0.38	0.19	39.76%	0.20	0.13	0.15	0.15	0.17	0.08	38.46%
	50users	0.20	0.15	0.15	0.14	0.15	0.10	26.63%	0.14	0.08	0.08	0.08	0.08	0.07	9.76%
	100users	0.15	0.11	0.12	0.10	0.11	0.11	-7.36%	0.10	0.06	0.07	0.06	0.06	0.05	11.79%
	200users	0.15	0.09	0.12	0.08	0.08	0.07	17.36%	0.11	0.05	0.09	0.05	0.05	0.05	3.40%
	300users	0.08	0.08	0.10	0.07	0.07	0.06	13.73%	0.05	0.05	0.07	0.05	0.05	0.04	24.77%
	mean	0.19	0.15	0.17	0.14	0.16	0.11	18.02%	0.12	0.07	0.09	0.08	0.08	0.06	17.64%
15-min	10users	0.30	0.30	0.30	0.27	0.28	0.21	21.20%	0.15	0.15	0.15	0.14	0.15	0.09	37.89%
	50users	0.14	0.14	0.15	0.12	0.12	0.09	24.88%	0.09	0.10	0.09	0.08	0.07	0.05	28.46%
	100users	0.11	0.11	0.12	0.09	0.09	0.08	8.68%	0.07	0.07	0.09	0.06	0.06	0.05	17.67%
	200users	0.10	0.10	0.09	0.07	0.07	0.05	31.02%	0.07	0.07	0.06	0.05	0.05	0.04	20.47%
	300users	0.08	0.08	0.09	0.06	0.06	0.05	15.70%	0.06	0.05	0.07	0.05	0.05	0.03	35.10%
	mean	0.15	0.15	0.15	0.12	0.13	0.10	20.30%	0.09	0.09	0.09	0.08	0.07	0.05	27.92%
30-min	10users	0.27	0.27	0.28	0.23	0.23	0.22	4.74%	0.15	0.15	0.19	0.15	0.14	0.11	23.15%
	50users	0.15	0.14	0.15	0.11	0.11	0.09	15.70%	0.12	0.09	0.09	0.08	0.07	0.07	2.81%
	100users	0.11	0.12	0.12	0.08	0.08	0.08	5.02%	0.08	0.09	0.08	0.06	0.06	0.05	8.16%
	200users	0.10	0.10	0.10	0.07	0.07	0.06	11.97%	0.07	0.06	0.07	0.05	0.05	0.04	14.69%
	300users	0.09	0.10	0.10	0.06	0.06	0.05	16.35%	0.06	0.06	0.07	0.05	0.05	0.04	12.78%
	mean	0.14	0.15	0.15	0.11	0.11	0.10	10.75%	0.10	0.09	0.10	0.08	0.07	0.06	12.32%
1-h	10users	0.29	0.28	0.27	0.19	0.19	0.19	-0.84%	0.19	0.23	0.18	0.15	0.15	0.12	17.53%
	50users	0.14	0.14	0.15	0.09	0.09	0.10	-11.87%	0.11	0.10	0.11	0.07	0.07	0.08	-8.50%
	100users	0.13	0.13	0.11	0.07	0.07	0.07	1.72%	0.10	0.09	0.08	0.06	0.06	0.05	14.47%
	200users	0.12	0.11	0.11	0.06	0.06	0.06	-4.37%	0.10	0.07	0.08	0.05	0.05	0.05	-3.68%
	300users	0.11	0.09	0.09	0.05	0.06	0.04	21.79%	0.09	0.06	0.07	0.04	0.05	0.04	6.43%
	mean	0.16	0.15	0.15	0.09	0.09	0.09	1.28%	0.12	0.11	0.10	0.07	0.08	0.07	5.25%

3.4.2 Performance Evaluation on CVR Baseline Estimation

In this section, we apply the proposed Load-PIN model to estimate the CVR baseline for actual feeders in North Carolina, USA. For each feeder, we first train and test the Load-PIN model using data collected in non-CVR days (same as in Section III.A). Next, the trained model is used for baseline estimation in CVR days. The estimated baselines are used to calculate the CVR factor for CVR efficacy assessment.

(1) Fixed CVR Duration

As shown in Table 3.2, the fixed CVR duration case is conducted using data collected from three residential distribution feeders in utility A, namely BR, DF and SL. All 24 labeled CVR events are in summer months with a fixed duration of 3 hours between 14:00 and 19:00 p.m. The CVR voltage reduction is 4%. The model performance is evaluated on non-CVR days. Note that forecasting bias is removed using methods introduced in Section II.C.6 when calculating CVR load reduction.

Table 3.4. Model performances on the 3 test feeders (in percentage)

(%)	Feeder	TCN	LSTM	MLP	SAE	BLSTM	Load-PIN
nRMSE	**BR**	3.12	3.13	3.8	4.05	3.67	**2.50**
	DF	3.87	3.78	5.56	6.35	5.8	**2.86**
	SL	3.66	3.72	3.91	4.03	3.67	**3.02**
EE	**BR**	2.02	2.03	2.63	2.91	2.54	**1.13**
	DF	2.86	2.75	4.03	4.91	3.78	**1.98**
	SL	2.54	2.65	2.77	2.47	2.18	**1.40**
Absolute Bias	**BR**	0.33	0.37	1.52	1.79	0.39	**0.03**
	DF	1.63	**0.18**	2.34	3.32	1.13	**0.22**
	SL	1.03	1.21	1.1	1.31	0.11	**0.05**

As shown in Table 3.4, Load-PIN outperforms the other methods (lowest $nRMSE$ and EE) by a large margin. From the results, we made the following observations

- As shown in Fig. 3.9, the results obtained by Load-PIN (the solid brown line) show similar trends as the average of all six models (the solid pink line). This shows that LoadPIN captures the trending information well.

- In Fig. 3.10, we randomly plot four CVR baselines for each feeder (out of 24 CVR events). The forecasted baseline shows a smooth transition from the pre-CVR to CVR periods and from the CVR to post-CVR periods.
- As shown in Figs. 3.9 and 3.10, for all three feeders, we observe clear load reduction (i.e. the value of CVR effect is negative in Fig. 3.9) in the first 1.5-hour. However, for feeders BR and DF, a pay-back period is observed in the second 1.5-hour. In fact, the amount of load reduction starts to diminish after the first hour. In some cases, feeder loads start to rise after the initial drop until exceeding the baseline.
- This CVR diminishing and subsequent pay-back effect is very likely caused by the increasing penetration of thermostatically controlled appliances (e.g., refrigerators, ovens, air conditioners) and LED lighting loads, which are rapidly replacing the incandescent lighting loads. Note that when voltage is low, appliances will turn on longer if a fixed amount of energy is required in each duty cycle.
- For feeder SL, even though the pay-back effect is less visible compared to feeders BR and DF, the CVR effect also diminishes after one hour.
- Overall, the results show that CVR efficacy will diminish in a prolonged CVR event, which indicates that the utilities may need to execute CVR for a period less than 2-hour.

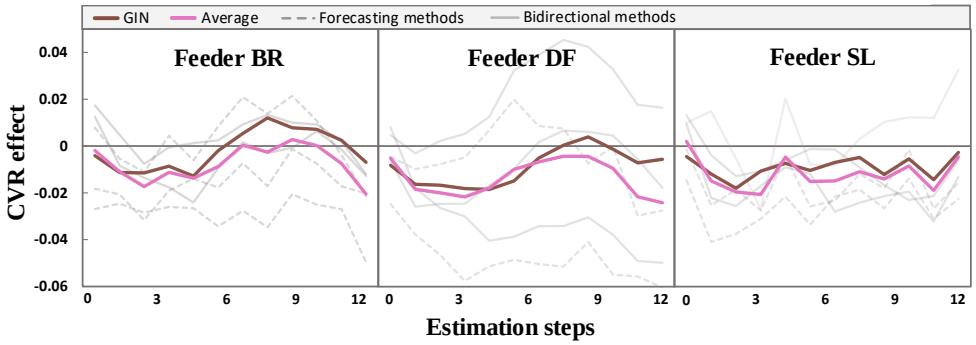


Figure 3.9. Averaged step-by-step CVR effects of the 3 test feeders in NewRiver

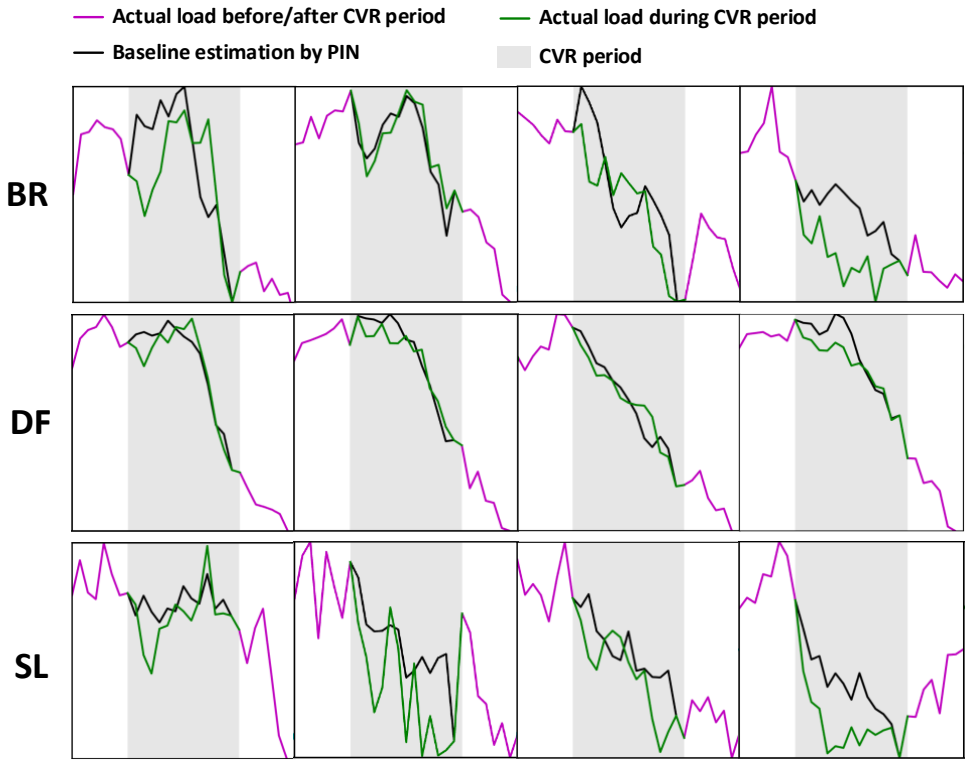


Figure 3.10. Examples of the Load-PIN generated CVR baseline

(2) *Variable-length CVR Cases*

As shown in Table 3.2, in this case, we use 5-minute data collected by a utility at a distribution substation bus to identify the CVR baseline for CVR events with variable durations. There are 33 CVR events with duration range from 75 minutes up to 190 minutes. Fig. 3.11 shows the daily voltage profiles at the feeder head of 3 identified CVR days as an example.

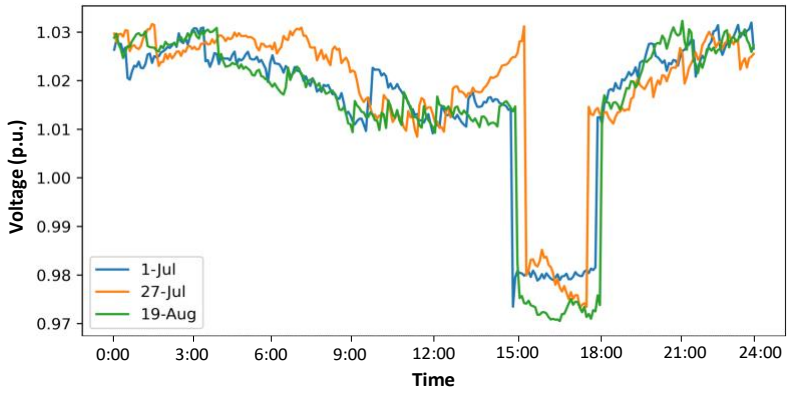


Figure 3.11. Examples of daily voltage profiles of three CVR event days measured at the feeder head

To make the bidirectional models (including Load-PIN) adapt to the varying-length CVR events, we randomly put a mask to each training sample with varying lengths between 1 - 4 hours. For the 3 forecasting models, we fix the output window to 4 hours so all possible CVR durations are covered in their forecasting range.

As shown in Table 3.5, Load-PIN outperforms all other models by achiving the smallest nRMSE and EE. The model bias is 0.43 only slightly higher than that of LSTM. The results demonstrate that Load-PIN can handle varying-length estimation tasks by leveraging the closest bidirectional data around each CVR events.

Table 3.5. Model performance on Fayetteville feeder case (in percentage)

(%)	TCN	LSTM	MLP	SAE	BLSTM	Load-PIN
nRMSE	3.23	4.13	4.93	4.31	3.18	**2.15**
Energy	1.91	2.69	3.37	3.03	1.95	**1.07**
Absolute Bias	0.82	**0.27**	1.17	1.74	1.57	0.43

We implement the six trained models to estimate the CVR baseline for all the 33 events, and then calculate the averaged CVR effect, as shown in Fig. 3.12. Note that due to the uneven durations of the CVR events, results are less stable after 2h because the durations of many CVR events are less than 2 hours. However, we can still see similar trends with the fixed-duration case (using data from utility B): the CVR effect is more significant in the first hour and the initial load drop period will be followed by a pay-back period where load will rise.

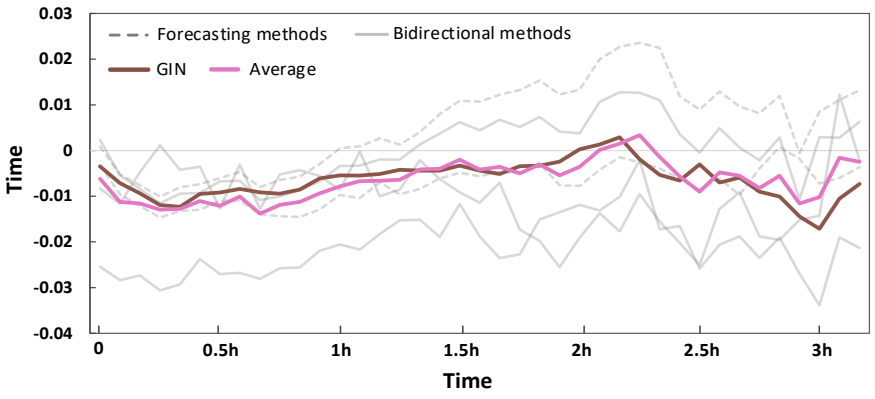


Figure 3.12. Averaged step-by-step CVR effects at a substation bus

3.5 Conclusion

In this paper, we propose a novel deep-learning model Load-PIN to solve the CVR baseload estimation problem. Load-PIN merges Gated Convolution and Multi-head self-attention mechanisms into the GAN based framework to enhance the estimation accuracy. Load-PIN is trained using the dynamic-masking strategy so that it can handle CVR events with varying duration. We first demonstrate that at higher load aggregation levels, higher data resolution can achieve better estimation accuracy. In general, 5-min and 15-min resolutions are sufficient for feeder level studies. Next, we demonstrate that the Load-PIN model can achieve 15-30% accuracy improvement under the suggested data granularity, compared with 5 benchmarking methods. Using the trained LoadPIN model for CVR baseline identification, we computed the CVR factors for CVR programs with fixed- and variable- CVR durations. We show that CVR can achieve load reduction in the first 1 hour. However, after 1.5 hours, the CVR effect starts to diminish and a pay-pack period can be observed. This may cause unexpected load peaks in post-CVR periods. From the results, we want to make two recommendations. First, the CVR execution duration should be less than 2-hours. Second, feeders with high penetration of thermostatically controlled loads may not be good candidates for prolonged CVR programs.

CHAPTER 4 ANOMALY DETECTION IN SMART DISTRIBUTION GRID WITH BATTERY ENERGY STORAGE SYSTEM

4.1 Background

Cyber-physical attacks against the smart grid can result in catastrophic financial losses and, more importantly, life-threaten issues. The three most common classifications based on the mechanism of attack delivery are cyber-based attacks, network-based attacks, and communication-based attacks. Cyber-based attacks are exclusively delivered through the system's internet layer. Code manipulation, command manipulation, FDIA, and sleep deprivation are examples of cyber-based attacks. Through virtual network access, network-based attacks are constructed without impacting the source code or firmware of the system or the physical communications system. The main attack in this category is Denial of Service (DoS), in which the network will become unreachable due to a high volume of useless packets. Even at the network layer, FDIA is feasible. Man-in-the-middle, packet sniffing, rogue node, and fuzzing are examples of additional network-based attacks. Communication-based attacks rely on the actual physical communications network to deliver the attacks. These attacks can be carried out by disrupting the communication channel or transmitting forged messages (e.g., FDIA).

When examining the categorization of cyber-physical attacks in depth, it is evident that FDIA could be implemented at all communication layers. Even worse, FDIA is more dangerous due to its difficulty to detect (e.g., stealth FDIA). In contrast to other attack types, the system may appear to operate normally without realizing the existence of FDIA until severe damage happens. Over the past decade, extensive research has been conducted on FDIA's effects, revealing a vast array of effects. [74] provides comprehensive statistics of FDIA research work

in recent years, as illustrated in Figure 4.1. Clearly, the current research on FDIA focuses primarily on the steady-state analysis of basic attacks on transmission systems. However, as stated previously, with the development of the smart distribution grid and the widespread deployment of energy storage systems, the risk of FDIA attacks on distribution networks is tremendous, but minimal research has been conducted in this area. This paper will therefore investigate stealth FDIA for the distribution system, particularly integrated with BESS. A real-time stealth FDIA generation method based on reinforcement learning is proposed, which can attack SoC, the most critical measurement quantity of BESS, without being detected by a state estimation-based bad data detection system, resulting in the miscalculation of the storage capacity by the control center, leading to erroneous energy management instructions, and causing in economic losses or power supply interruption accidents. Then, an anomaly detection algorithm based on the graph convolutional network is proposed combined with the essential characteristics of the power grid as a graph, which can effectively identify stealth FDIA for BESS SoC.

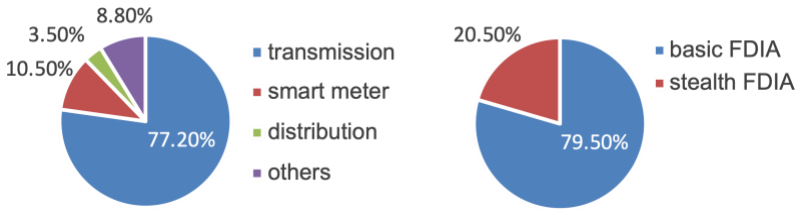


Figure 4.1. Statistics of FDIA research

The remaining of this chapter is structured as follows. The chapeter begins by introducing the model of the integrated power distribution system and BESS, including the state estimation technique of the distribution system with the BESS unit and the estimation method for battery SoC. Then, an attacker algorithm based on the reinforcement learning framework is introduced, including training environment development, model learning method, simulation experiment, and evaluation of results. Then a stealth FDIA detection technique is proposed using the graph convolution network (GCN) and gated recurrent network (GRU). Compared to current state-of-

the-art algorithms, experimental results indicate that this method can achieve high precision while maintaining a light architecture, which provides a practical solution for large-scale power systems.

4.2 System Model

4.2.1 Distribution System Monitoring and State Estimation with BESS Integration

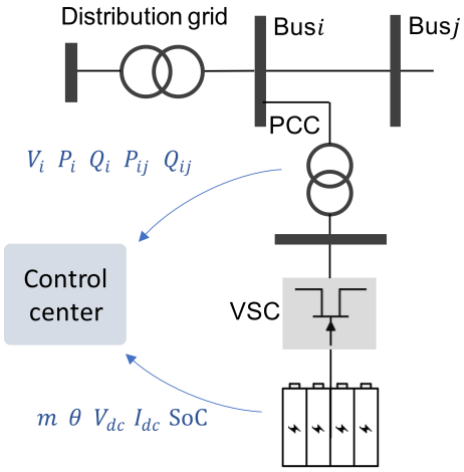


Figure 4.2. A single-line diagram of a typical distribution system with BESS integration

As depicted in Figure 4.2, for a distribution system with integrated BESSs, the measurement devices communicate with the control center via various communication links, such as wide area networks, public cellular networks, etc. At the point of common coupling (PCC), the BESS connects via the three-phase AC-DC voltage source converter (VSC), coupling transformer, and LCL filter with bidirectional power flows. Conventional distribution system monitoring is mainly dependent on the distribution supervisory control and data acquisition (SCADA) system, which provides most of the real-time measurements, including the voltage amplitude (V_i), active and reactive power injection (P_i, Q_i) of each bus, and active and reactive power flow (P_{ij}, Q_{ij}) between buses. When introducing BESS, The BESS controller usually determines the magnitude modulations (m) and phase-displacement angles (θ) based on the active and reactive power setpoints. For battery monitoring and control, a battery management

system collects battery bank measurements, such as terminal voltage (V_{dc}) and current (I_{dc}), and estimates battery pack statuses, such as SoC.

SE is used to analyze meter measurement data and power system models to estimate the state of the power grid as precisely as possible. SE is the estimation of unknown state variables in a power grid based on meter readings. The control center uses the output of SE to perform contingency analysis, which involves reasoning about potential operational problems in the grid, actions they may take to avoid these problems, and the potential side effects of these actions. A power flow model is a set of equations that depicts the flow of energy on each transmission line of a power grid. AC power flow is power flow models that incorporate both real and reactive power and are formulated using nonlinear equations. Generally, the SE can be formalized by

$$\boldsymbol{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{bmatrix} = \begin{bmatrix} h_1(x_1, x_2, \dots x_n) \\ h_2(x_1, x_2, \dots x_n) \\ \vdots \\ h_m(x_1, x_2, \dots x_n) \end{bmatrix} = h(\boldsymbol{x}) + e \tag{4.1}$$

Where $\boldsymbol{z} = [z_1, z_2, \dots z_m]^T$ represents the measurement vector with m measurements. $\boldsymbol{x} = [x_1, x_2, \dots x_n]^T$ represents the state vector with n variables. $\boldsymbol{e} = [e_1, e_2, \dots e_m]^T$ represents the measurement error vector, in which each error variable is assumed to follow Gaussian distribution with zero mean and variance of σ_i. $h(\cdot)$ is a nonlinear vector function derived from the network topology based on AC power flow models, which can be derived by

$$P_{ij} = V_i V_j \left(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}\right) - G_{ij} V_{ij}^2 \tag{4.2}$$

$$Q_{ij} = V_i V_j \left(G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}\right) + V_i^2 \left(B_{ij} - b_{s,ij}\right) \tag{4.3}$$

$$P_i = V_i \sum_{j=1}^{n} V_j \left(G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}\right) \tag{4.4}$$

$$Q_i = V_i \sum_{j=1}^{n} V_j \left(G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}\right) \tag{4.5}$$

where G_{ij}, B_{ij} represents the conductance and susceptance of bus parameter, $b_{s,ij}$ denotes the shunt susceptance associated to the respective π model.

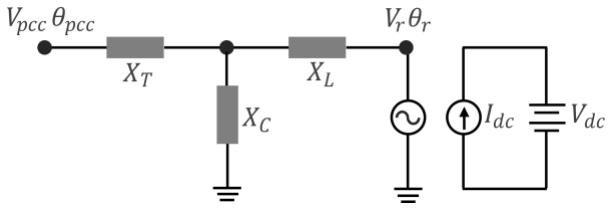


Figure 4.3. Equivalent circuit model of VSC for power system state estimation

The DC grid of the battery storage usually connects with the AC grid using the VSC. The connection of the VSC to the grid is facilitated by the presence of a coupling transformer and an LCL filter at the output side of the VSC. [75] proposed an enhanced VSC converter model for power system state estimation, as shown in Fig. 4.3, where X_T, X_C, and X_L represent the equivalent impedance of the coupling transformer and LCL filter. The DC grid is coupled with the AC side follows the control signal of PWM modulation index *m*, and power conservation by the following equation

$$V_r = mV_{dc}/\sqrt{2} \tag{4.6}$$

$$P_{ri} + P_{dc} + P_{loss} = 0 \tag{4.7}$$

$$P_{dc} = V_{dc}I_{dc} \tag{4.8}$$

$$P_{loss} = I_{ri}^2 R_{ac} + V_{dc}^2/R_{dc} \tag{4.9}$$

where R_{ac} and R_{dc} represents the series resistance of AC and DC bus. The following measurement functions are added to measurement function $h(\cdot)$ based on eq. (4.6) - (4.9)

$$\hat{V}_{dc} = V_{dc} + e_{V_{dc}} \tag{4.10}$$

$$\hat{I}_{dc} = I_{dc} + e_{I_{dc}} \tag{4.11}$$

$$P_{ri} = -V_{dc}I_{dc} - I_{ri}^2 R_{ac} - V_{dc}^2/R_{dc} \tag{4.12}$$

Based on formulas (4.2)-(4.5) and (4.10)-(4.12), the measurement vector of the smart distribution system with BESS integration is defined as $\boldsymbol{z} = \left[|V_i|, P_i, Q_i, P_{ij}, Q_{ij}, V_{dc}, I_{dc}, m\right]$, the state vector is defined as $= [|V_i|, \theta_i, V_{dc}, I_{dc}]$.

The purpose of SE is to obtain the estimated system states x that best fit the measurements z using a mathematical model (4.1). The most prevalent method for estimating the state of a power system is the weighted least squares (WLS) estimator [76], which minimizes the weighted measurement residuals by solving the optimization

$$\hat{\boldsymbol{x}} = arg\min_{\boldsymbol{x}} \big(\boldsymbol{z} - h(\boldsymbol{x})\big)^T \boldsymbol{W} \big(\boldsymbol{z} - h(\boldsymbol{x})\big) \tag{4.13}$$

Where $\boldsymbol{W} = diag(\sigma_1^{-2}, \sigma_1^{-2}, \cdots, \sigma_m^{-2})$ is the weights matrix for measurements, which is usually determined by the variance of measurement noise. The optimization problem (4.13) can be solved iteratively using Gauss-Newton algorithm [76].

However, SE using an AC power flow model is computationally costly and does not always converge to a solution. As a result, sometimes, power system engineers approximate the AC power flow model with a linearized power flow model, the DC power flow model [77]. It is a simplification of a complete AC power flow that focuses just on active power flows while disregarding voltage support, reactive power management, and transmission losses by assuming the bus voltage magnitudes are already known and equal to 1.0 per unit and ignoring all shunt elements, bus and branch, and reactive power flow. The DC power flow can be formulated as

$$P_{ij} = \frac{\theta_i - \theta_j}{X_{ij}} \tag{4.14}$$

Accordingly, the measurement function can be modified if DC power flow is applied as

$$\boldsymbol{z} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{e}$$

where $\boldsymbol{H}$ is Jacobian matrix determined by grid topology and line parameters. And the WLS state estimation results result can be obtained by solving

$$\hat{\boldsymbol{x}} = (\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{H})^{-1}\boldsymbol{H}^T\boldsymbol{W}\boldsymbol{z} \tag{4.15}$$

In the smart power system, bad data measurements can originate from a variety of sources, including meter reading errors and cyber-attacks (e.g., FDIA). A bad data detection system takes into account the geographical correlation and statistics of measurement noises for BDD based on measurement residuals [78]

$$\|\boldsymbol{z} - h(\hat{\boldsymbol{x}})\|_2^2 \tag{4.16}$$

where $\hat{\boldsymbol{x}}$ is the estimated state variables using the WLS method. In the case of measurements with noises that are normally distributed, the squared measurement residual stays below a threshold τ when using a hypothesis test with a significance level λ. As a result, $\|\boldsymbol{z} - h(\hat{\boldsymbol{x}})\|_2^2 > \tau$ signals bad data with a chance of false alarm probability of λ [75].

4.2.2 Soc Estimation of BESS

SoC calculations must be precise for any battery-powered equipment intended to assist the energy management system (EMS) at the control center. In order to fulfill the needs of grid support functions such as voltage regulation and energy management, the battery bank of a BESS is often made up of a large number of battery cells. If the estimation SoC of each individual battery cell is considered, the result will be a high computational complexity, making it challenging to satisfy the needs of real-time applications. In addition, the balancing technology will be utilized in practice by the battery management system in order to keep the power of each battery under the balance [79]. As a result, when attempting to estimate the SoC of the BESS, the entire battery pack is typically considered to be a unit model as shown in Fig. 4.4 [80]. The terminal voltage V_{dc} can be determined as

$$V_{dc} = E_m + V_{RC} + R_{in}I_{dc}^t \tag{4.17}$$

where R_{in} is the battery's internal resistance, E_m is the internal open circuit voltage, which can be modeled as a non-linear function of SoC [81]

$$E_m = g(\varphi) \tag{4.18}$$

which can be approximated by a polynomial equation of the seventh-order [81].

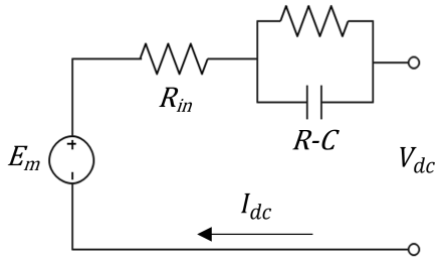


Figure 4.4. Equivalent circuit model for battery bank

Theoretically, the battery SOC can be determined using the Coulomb counting approach by integrating the measured battery current as

$$\varphi_t = \varphi_{t-1} - \frac{\Delta t}{3600\mathcal{C}} I_{dc}^t \tag{4.19}$$

where φ_t is the battery SoC at time step t, φ_t is the previous SoC, $\mathcal{C}$ represents the battery bank capacity in Ah, Δt is the sample time interval in seconds. However, due to measurement errors and noise, battery age, and unknown initial SoC, it might be difficult to calculate SoC in practice purely relying on the Coulomb counting method. Therefore, the extended Kalman filter (EKF) is adopted in [81] to determine the battery's SoC by approximating the nonlinearity of the system's dynamics using a linearized version of the nonlinear system model. The EKF is insensitive to the initial value and can adjust the output based on the measurement and the model prediction as follow

$$\begin{cases} \hat{x}_k^- = f(\hat{x}_{k-1}^-, u_{k-1}) & \text{State estimate time update} \\ P_k^- = A_{k-1} P_{k-1} A_{k-1}^T + Q(t) & \text{Error covariance time update} \\ G_k = \hat{P}_k^- C_k^T [C_k P_k^- C_k^T + R(t)]^{-1} & \text{Kalman gain matrix} \\ x_k^- = \hat{x}_k^- + G_k [y_k - h(\hat{x}_k^-, u_k)] & \text{State estimate measurement update} \\ P_k = (I - G_k C_k) P_k^- & \text{Error covariance measurement update} \end{cases} \tag{4.20}$$

where x, u are the state and control variables, which refer to battery SoC, and I_{dc} in battery SoC estimation problem. $f(x,u)$ and $h(x,u)$ are the system function determined by Eq. (4.15 - 4.17). A is the Jacobian matrix of partial derivatives of $f(x,u)$ with respect to x_{k-1} and u_{k-1}, C is the Jacobian matrix of partial derivatives of h with respect to x_k and u_k. P_k^- and P_k is the prior and posteriori estimate error covariance. $Q(t)$ is the process noise, which is calculated empirically with reference to the measurement noise variance [82]. $R(t)$ is the measurement noise covariance, which is assumed to be constant.

4.3 Stealth FDIA Formulation Against Soc Estimation

4.3.1 State Estimation-Based Bad Data Detection in Power System

The existence of bad data points has the potential to drastically reduce the performance of any of the static state estimators for the power system. [78, 83, 84] proposed bad data suppression algorithms based on real-time state estimation and its variants.

Recent research demonstrates that an adversary can bypass existing bad data detection algorithms, posing grave operational risks to power grid systems, which is stealth FDIA. To compromise the FDIA, [85] proposed a stealth FDIA algorithm against the DC state estimation model by adding a nonzero attack vector $\boldsymbol{a} = [a_1, a_2, \dots, a_m]$ to the original sensor measurements vector z. The attacked measurement vector $\boldsymbol{z}_a = \boldsymbol{z} + \boldsymbol{a}$. This altered vector $\boldsymbol{z}_a$ is transmitted to the control center, which uses it to produce false estimates $\boldsymbol{x}_a = \boldsymbol{x} + \boldsymbol{c}$, where $\boldsymbol{x}$ represent the original estimations and $\boldsymbol{c}$ represent the malicious errors introduced to $\boldsymbol{x}$. [85] researched an attack approach that can bypass the present BDD test using the principle

$$\boldsymbol{a} = \boldsymbol{H}\boldsymbol{c} \tag{4.21}$$

4.3.2 Construction of Stealth FDIA Against SoC Estimation

Figure 4.5 depicts the workflow of the bad data detection system in a distribution system. First, the control center obtains system measurement data via the communication links, including the voltage amplitude of each bus, active and reactive power injection, and the power flow between the buses. Since the system considered in this study contains BESS, additional measurements must be added compared to the conventional grid, including the DC side voltage V_{dc} and current I_{dc}, SoC (φ) measured by battery management system (BMS), as well as the control signal of PWM modulation index m in the VSC. Initially, residual-based BDD, as depicted in Eq. (4.14), will be used to check these field measurements. If the measurements can pass through the BDD, the obtained estimated state variables (V_{dc}, I_{dc}) will be input to an EKF-based SOC estimator, which is the same as the field BMS, and calculate an estimated SoC ($\tilde{\varphi}$). This estimated SoC ($\tilde{\varphi}$) will be cross-validated with the measured SoC to ensure that the remaining battery charge estimate is accurate.

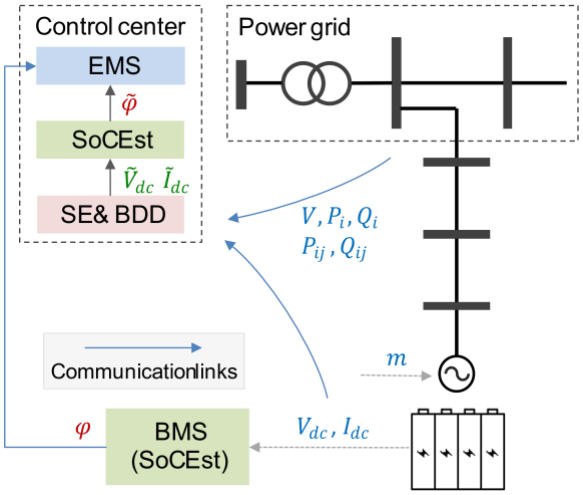


Figure 4.5. Diagram of a typical bad data detection in smart distribution with BESS integration

As discussed in Section 4.1.1, the relationship between measurements and state variables can be expressed using the nonlinear function h. In general, the system function h of a power system is determined by the system's topology and line impedances, along with battery parameters such as capacity, internal resistance, open circuit function, etc. We assume the

attacker has access to the topology, line, and battery parameters of the target power system and can inject malicious measurements into compromised meters to undermine the state estimation process. It is also assumed that, the EKF is already in a steady state before the attack.

To successfully launch the attack against the battery SoC, the attacker must first modify the original measured SoC by attacking the communication link. However, this is insufficient because of the SoC cross-validation mechanism. In order to successfully trick the control center into accepting false SoC data, the attacker must inject the offset into the original V_{dc} and I_{dc} to mislead the EKF into obtaining a value close to the attacked SoC.

Analytical investigation of the construction concept of stealth FDIA against SoC estimation of BESSs is conducted in [80] through the development of static FDIAs targeting a single snapshot. Experiments demonstrate that the formulated static FDIAs can have a substantial effect on the precision of SoC estimation and circumvent the typical residual-based BDD in smart distribution systems with BESS integration as an optimization problem formulated as

$$\max_{a_t} |\varphi_t^a - \varphi_t| \tag{4.22}$$

subject to device operation constraints (e.g., battery voltage and current limitation), and stealth attack constraint in Eq. (4.19), where φ_t^a and is the battery SoC estimation using EKF follows Eq. (4.18) using attacked and original measurement vector $\boldsymbol{z}_a$ and $\boldsymbol{z}$ respectively. The attack vector $\boldsymbol{a} = [\Delta V_{dc}, \Delta I_{dc}]$. Since the DC power flow model is adopted, (4.21) can be solved using linear programming (LP) in real-time application. However, the simplification assumption of DC power flow is limited to those MW-oriented applications where the effects of network voltage and Var conditions are minimal [86]. So DC power flow is not suitable for distribution feeder

analysis. However, if the non-linear power flow equations in Eq. (4.2-4.5) is adopted, the stealth attack constraint becomes

$$\boldsymbol{a} = h(\boldsymbol{x} + \boldsymbol{c}) - h(\boldsymbol{x}) \tag{4.23}$$

This nonlinearity makes the stealth attack problem unsolvable with LP and introduces a high computational cost. Based on our experiments, nonlinear constraints can not meet the requirements of real-time applications. Considering the above problems, a reinforcement learning-based stealth attack generation method is proposed. It transforms the traditional online optimization problem into offline deep learning training, making it possible to implement stealth attacks under the premise of satisfying nonlinear constraints.

4.3.3 Stealth FDIA using Deep Reinforcement Learning

Deep reinforcement learning (DRL) is a subfield of machine learning that combines deep learning (DL) and reinforcement learning (RL). RL examines the problem of a computational agent learning through trial and error to make decisions. Deep learning is incorporated into the solution, enabling agents to make decisions based on unstructured input data without the need for manual engineering of the state space. A typical framework of DRL is shown in Fig. 4.6. A reinforcement learning framework consists of an agent operating in an environment modeled by the current state. Based on the current state, the agent is capable of performing specific actions. After selecting an action at time t, the agent receives a reward and switches to a new state that is dependent on its current state and the chosen action. The process can be formulated as Markov Decision Process (MDP), in which the future of the process only depends on the current state [87]

$$P\left(S_{t+1}\middle|(s_0, a_0), \dots (s_t, a_t)\right) = P\left(S_{t+1}\middle|(s_t, a_t)\right) \tag{4.24}$$

where s_t, a_t are the state and reward of the current time step, respectively, and P is the state transition probability.

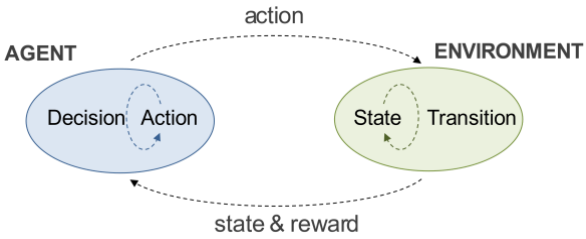


Figure 4.6. A typical framework of reinforcement learning

For the stealth FDIA problem, the RL environment is configured to simulate the distribution system with BESS, including the field measurement time series, the SE and BDD of the control center, and the battery SoC observer. The agent is an attacker modeled as a deep neural network. It collects field measurements and rewards as state inputs. The attack vector for FDIA is then generated as the action. This attack vector is first run through the SE-based BDD. If it is determined that the data is abnormal, the episode is terminated, and the reward function returns a large negative value as a penalty. If the attack vector passes BDD, the V_{dc} and I_{dc} obtained through SE will be used for the SoC estimation based on EKF. An SoC will be calculated as φ_a. The reward function will compare the original SoC (φ) with the φ_a, and output a positive reward accordingly, which is used to motivate the agent to generate a policy that can pass BDD and maximize SoC error.

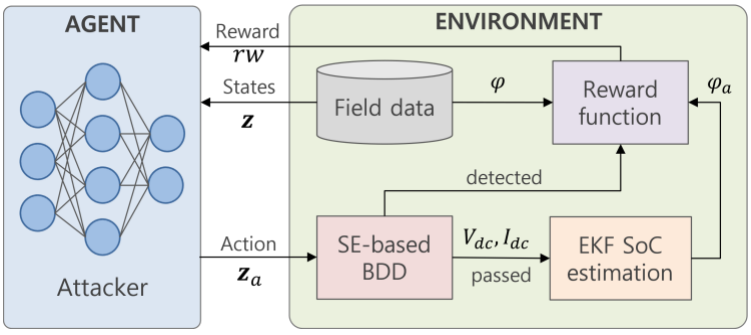


Figure 4.7. RL framework for stealth FDIA against SoC

Instead of maximizing the SoC error in [85], this article proposes a target SoC mode, which means the attacker desires to cause the SoC to reach the target percentage at a specific time by injecting false data into the measurement vector; the injected data must pass BDD detection as a prerequisite. As shown in Eq. (4.24), a time-weighted reward mechanism is proposed to achieve this objective as

$$reward = \begin{cases} max\left(0, 1 - |\varphi_a - \varphi_{tar}| \cdot \left(\frac{t}{T}\right)^2\right) & \text{successful attack} \\ -100 & \text{detected or constraints violation} \\ 100 & \text{bonus if } |\varphi_a - \varphi_{tar}| < 1\% \text{ at end} \end{cases} \tag{4.25}$$

where φ_{tar} is the target SoC value, t is current time step, T is the total time steps in one episode. The constraints include the SoC limitation, battery voltage and current limitation. To accommodate the FDIA problem, the discrete action space injects bias to DC voltage and current defined as 50 bias pairs of DC voltage and current, accumulating to the original V_{dc} and $I_{dc.}$

Q learning is a simple method for agents to learn how to act optimally in controlled Markovian domains. It is an incremental method for dynamic programming that imposes minimal computational demands. It functions by gradually enhancing its evaluations of the quality of particular actions at particular states [88]. Given strategy π for FDIA, the action value function Q for selecting action a under input state s can be defined as

$$Q^{\pi}(s, a) = \sum_{t=0}^{\infty}\{\gamma^t R(s_t, a_t) | s_0 = s, a_0 = a\} \tag{4.26}$$

where $\gamma \in (0,1)$ is the discount factor for the future reward value, $R(\cdot)$ is the reward function depicted in Eq. (4.24). The Q function measures the benefits of performing an action under a given set of conditions by calculating the accumulated value of the long-term expected reward. This function is utilized in the process. The optimal Q function Q^* can be defined as

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a) \tag{4.27}$$

In a deep reinforcement learning algorithm, deep neural networks (DNN) are adopted to fit the Q function, which can give extensive capabilities for feature extraction and function fitting, making it suited for handling more complicated practical situations like stealth FDIA. To improve the neural network training, decayed ϵ-greedy [89] and double Q learning [90] strategies are employed. Table 4.1 shows the pseudocode of the DQL algorithm.

Table 4.1. Pseudocode of DQL algorithm

Algorithm 4.1 Deep Q learning

Initialize action value network Q target action value network $\hat{Q}$ and update steps interval C

Initialize experience replay memory D and threshold τ

while not converged

Update ϵ with ϵ-decay

Select action a based on state s using policy ϵ-greedy

Input action to environment, acquire $done$ signal, reward r and next state s'

Memorize $(s, a, r, s', done)$ in experience replay buffer D

if size(D) > τ

Randomly sample a minibatch of N transitions from D

for each $(s_i, a_i, r_i, s_i', done_i)$ in minibatch

if $done_i$

$y_i = r_i$

else

$y_i = r_i + \max_{a' \in A} \hat{Q}(s_i', a')$

End

end

Compute loss $L = \frac{1}{n}\sum_{i=0}^{N}(Q(s_i, a_i) - y_i)^2$

Update Q parameters using gradient decent algorithm to minimize loss L

Deep copy network parameters from Q to $\hat{Q}$ every C steps

end

end

4.4 Stealth FDIA Detection Method

4.4.1 Overview of Deep Learning-Based FDIA Detection in Smart Grid

As explained in the preceding sections, the FDIA is one of the main types of attacks that can damage smart grid systems. Current studies demonstrate that deep learning-based approaches are effective in FDIA detection. [91, 92] proposed using auto-encoder (AE) as a classification strategy to detect FDIA, as AE are able to learn latent correlation structures in the data in an unsupervised manner, enabling them to detect corrupted data. [93] utilized a Convolutional Neural Network (CNN) and a Long Short Term Memory (LSTM) network to identify anomalies that cannot be recognized by conventional SE-based BDD. [94] proposed a Conditional Deep Belief Network (CDBN) that employs Conditional Gaussian-Bernoulli RBM (CGBRBM) to extract high-dimensional temporal characteristics for assessing temporal attack patterns presented by real-time measurement data. [95] used a conditional deep belief network (CDBN) to analyze time-series input data and collected characteristics to detect the FDIA. [96] integrated the autoencoders into an advanced generative adversarial network (GAN) framework that could detect anomalies under FDIAs by capturing the discrepancy between abnormal and secure measurements.

The above anomaly detection models are proven effective. However, they overlooked the characteristic of power grids as graphs. A graph is a natural representation of a collection of things and their connections. For over a decade, researchers have created neural networks that act on graph data called graph neural networks [97] (GNNs). Physics simulations [98], fake news detection [99], and recommendation systems [100] are beginning to see practical applications. In a smart grid, anomaly detection research [101] proposed an approach based on GNN to identify the presence and location of the FDIA. A graphical detection technology that uses GNN is

developed in [102] for detecting tampered measurements without external knowledge and manual preprocessing of historical data. However, both graph-based methods only take a snapshot of the system as the model input and neglect to mine temporal information associations between consecutive frames.

4.4.2 Stealth FDIA detection using temporal graph convolution network

(1) Model framework

To better mine and utilize the graph nature of the power grid and the temporal information, a grid-temporal graph convolution network (Grid-TGCN) is proposed to identify stealth FDIA attacks against SoC in the distribution system with BESS integration as shown in Fig.4.8. The FDIA detection task is specified as a binary graph-level classification problem. Under this framework, multiple frames of system measurements are encoded in graph format as inputs of TGCN. The GCN is first used to extract spatial features, and the gated recurrent unit (GRU) is used for temporal feature analysis. The output of the final classification is generated by fully connected layers, indicating the possibility of frames containing stealth FDIA.

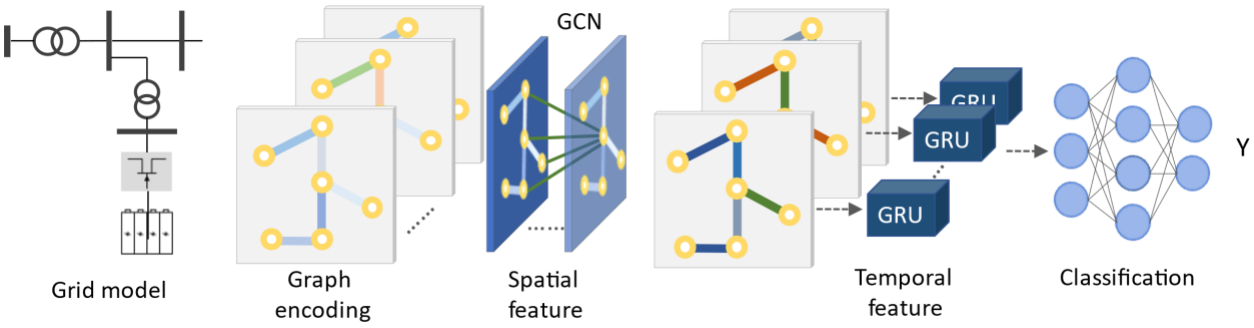


Figure 4.8. Framework of Grid-TGCN

The GCN filter, which operates on the nodes of a graph, is responsible for capturing the spatial properties that are located between the nodes through the use of its first-order neighborhood follow

$$h^{(k)} = D^{-1}\hat{A} \cdot h^{(k-1)}W^{(k)^T} + h^{(k-1)}B^{(k)^T} \tag{4.28}$$

where $h^{(k)}$ represents the features output at k layer, W and B are learnable parameters of weight and bias, $\hat{A} = A + I$ is the adjacent matrix with added self-connection, I is identity matrix, $D = \sum_j \hat{A}_{ij}$ is the degree matrix for normalization purpose. In the Grid-TGCN model, a 5-layer of GCN is designed to obtain spatial dependence between buses. The average pool operation is applied at the last layer to summarize spatial features.

Since the measurement data in the smart distribution system are time-series signals, obtaining temporal dependence between frames is another crucial aspect of the FDIA detection problem. LSTM model [103] and GRU model [104] have been shown to be effective in extracting temporal features. The LSTM and GRU are equally effective at different activities and employ gated mechanisms to memorize as much long-term knowledge as possible. However, because of its complicated structure, LSTM takes longer to be trained than GRU, because GRU has fewer parameters and a more straightforward structure. Therefore, in order to extract temporal dependence from the smart meter traffic data, we applied the GRU layers followed by GCN in the Grid-TGCN model. Fig 4.9 shows the calculation of GRU as

$$\boldsymbol{R}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xr} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hr} + \boldsymbol{b}_r) \tag{4.29}$$

$$\boldsymbol{Z}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xz} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hz} + \boldsymbol{b}_z) \tag{4.30}$$

$$\widetilde{\boldsymbol{H}}_t = tanh(\boldsymbol{X}_t \boldsymbol{W}_{xh} + (\boldsymbol{R}_t \odot \boldsymbol{H}_{t-1}) \boldsymbol{W}_{hh} + \boldsymbol{b}_h) \tag{4.31}$$

$$\boldsymbol{H}_t = \boldsymbol{Z}_t \odot \boldsymbol{H}_{t-1} + (1 - \boldsymbol{Z}_t) \odot \widetilde{\boldsymbol{H}}_t \tag{4.32}$$

where $\boldsymbol{X}$ is spatial features inputs from GCN layers extracted from multiple measurement frames. $\widetilde{\boldsymbol{H}}_t$ is the candidate latent state, $\boldsymbol{H}_t$ is the new state, $\boldsymbol{R}_t$ is the reset gate that helps capture short-term dependencies in sequences, $\boldsymbol{Z}_t$ is update gate that helps capture long-term dependencies in sequences, $\boldsymbol{W}$ and $\boldsymbol{b}$ are learnable parameters.

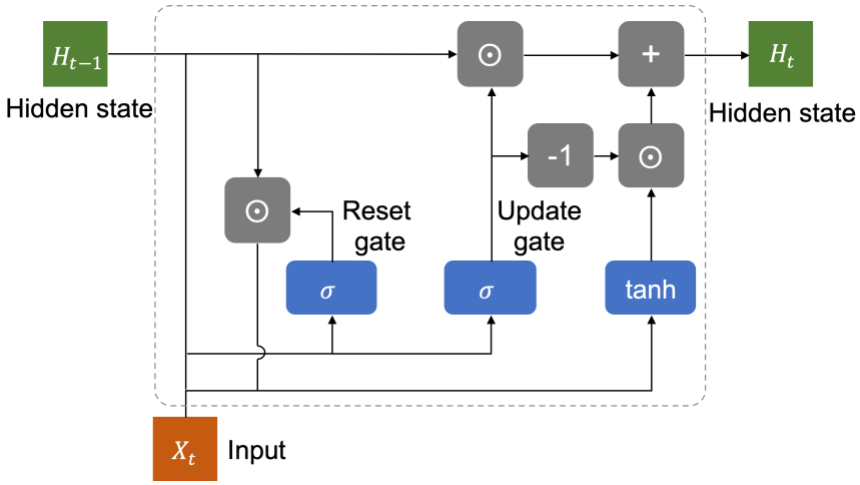


Figure 4.9. Calculation flow of GRU

(2) Loss function design

In this study, the detection of FDIA is formulated as a binary classification problem. The final layer is a fully connected network, which employs the Sigmoid activation function to match labels in the training dataset. For the labeling configuration adopted in the simulation, the binary label Y of a measurement frame is based on the following criterion:

$$Y = \begin{cases} 1 & \text{under FDIA} \\ 0 & \text{no attack} \end{cases} \tag{4.33}$$

To achieve this goal, the loss function that measures the Binary Cross Entropy (BCE) between the binary label and the output probabilities is adopted as

$$L = Y \log \hat{Y} + (1 - Y) \log\left(1 - \hat{Y}\right) \tag{4.34}$$

where Y and $\hat{Y}$ denotes the real label and model prediction.

(3) Evaluation metrics

This research uses F1-Score and accuracy to evaluate the efficacy and practicability of the suggested false data attack detection approach in Eq. in order to evaluate the performance detection of the model (4.35-4.38). First, the following variables are defined: True positive (TP) is the FDIA's identification of an attack. False negative (FN) refers to FDIA recognized as normal data. The true negative value (TN) identifies the normal measurement for normal

operation. False positive (FP) refers to a normal sample is recognized as FDIA attack. There are three evaluation metrics specified as:

$$accuracy = TP + TN/(TP + TN + FN + FP) \tag{4.35}$$

$$precision = TP/(TP + FP) \tag{4.36}$$

$$recall = TP/(TP + FN) \tag{4.37}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{4.38}$$

4.4 Case Study

4.4.1 Simulation Model and Dataset

To mimic the operational behavior of a actual distribution feeder more closely, a centralized feeder model is developed based on the IEEE 123 bus system in Simulink, as shown in Figure 4.10. The feeder head (Bus 1) is equivalent to Bus 149 in the IEEE 123 bus system as depicted in Fig. 4.10 and Fig. 4.11. All feeder load is aggregated and treated as a centralized load connected to Bus 3. In order to emulate the power consumption behavior of actual residential users, 83 users in Austin, TX with load profile from August 1st, 2017, to September 1st, 2017 [45], are selected to aggregate the node load in the IEEE123 bus system. The reference power for node i (P_{ref}^{i}) is provided by IEEE 123 bus specification. Random user will be added to the node i until the average annual power is accumulated to 40% of P_{ref}^{i}. The VSC model designed in [105] is used to bridge the battery to the AC grid. The battery model from [106] and the BESS operation data from [107] are utilized to simulate BESS operations. The battery parameters are summarized in Table 4.3. As illustrated in Fig. 4.12, the OCV curve of a battery is approximated

by a 9th order polynomial regression. Figures 4.12 and 4.13 depict a 24-hour illustration of feeder load power, BESS power, and battery SoC on August 1st, 2017.

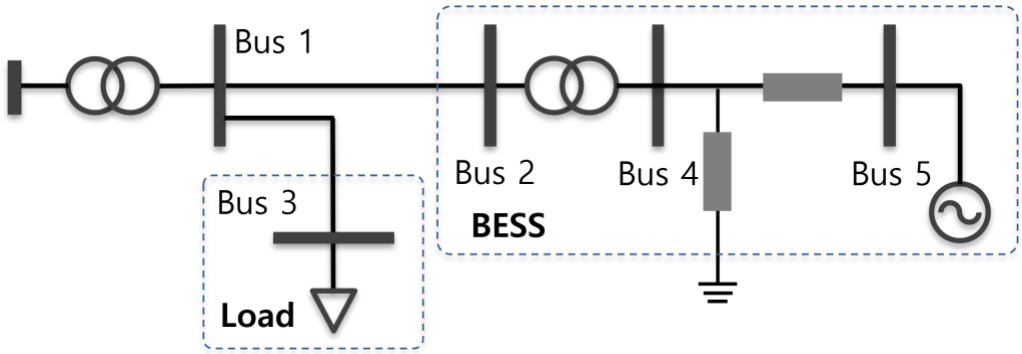


Figure 4.10. One line diagram of test grid model

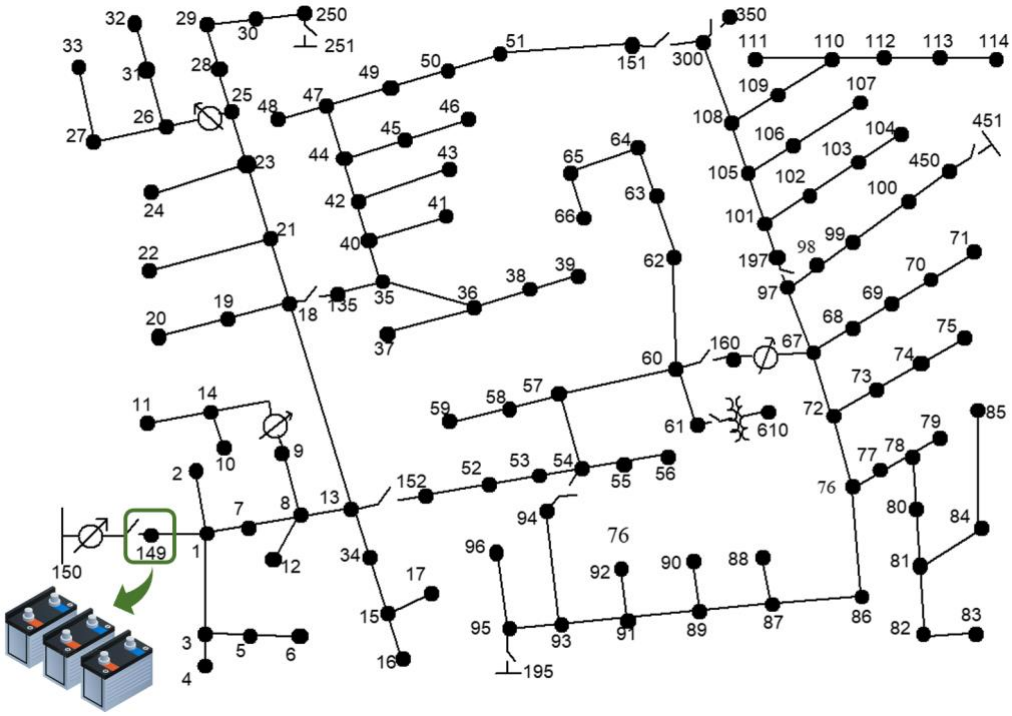


Figure 4.11. IEEE 123 bus test feeder

Table 4.2. Battery model parameters

Nominal Capacity and power	571.9kWh/816Ah 250kW
Nominal DC voltage and current	700.8V / 357A
DC voltage range	595.2 – 787.2V
DC current range	-400A – 400 A

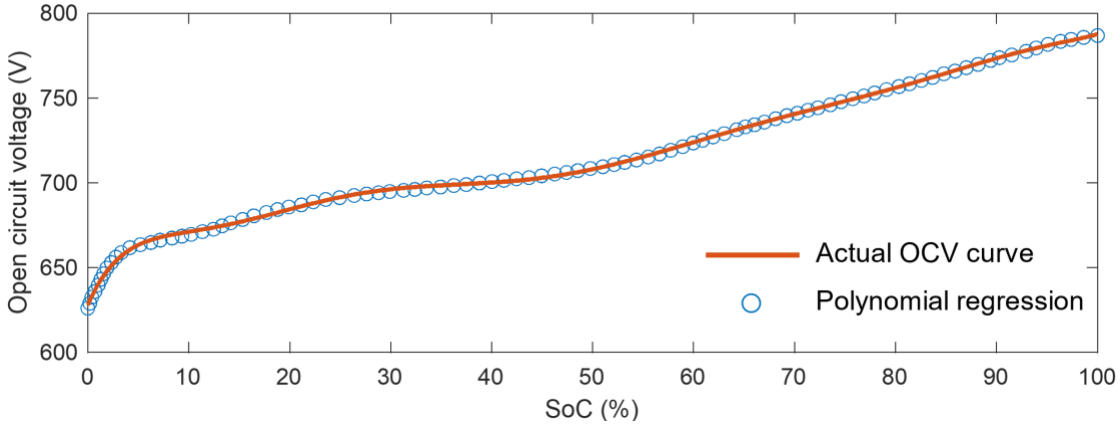


Figure 4.12. OCV-SoC curve regression

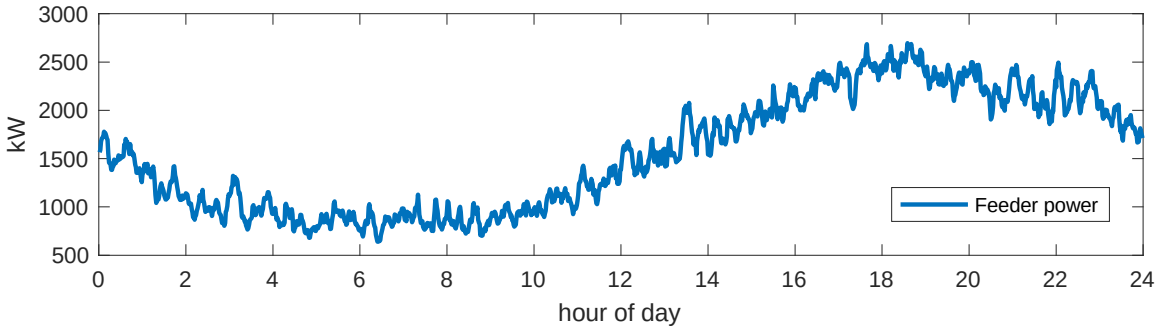


Figure 4.13. Example of feeder load profile on August 1st, 2017

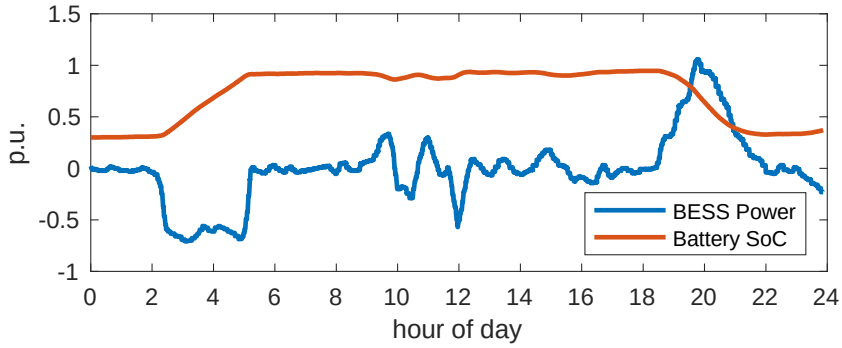


Figure 4.14. BESS power and battery SoC on August 1st, 2017

4.4.2 RL-Based Stealth FDIA against SoC Estimation

For the simulation environment configuration, it is assumed that the control center collects measurement data every five minutes for SE-based BDD. The BDD trigger threshold was set at 1.1 times the maximum permissible residual error under normal working conditions. The attack duration was set to 4 hours (60 steps every episode). The target SoC is randomly

selected within the feasible range of battery SoC. The attack starts at a random moment between August 1st and August 25th, and the test set is the remaining data from August 25th to September 1st. The action-value function is modeled using a 512*256 fully connected neural network. The framework for deep learning is implemented using the Pytorch platform. The model was for 20,000 episodes trained on an RTX 3080 GPU. Table 4.3 provides a summary of the training hyper parameters.

Table 4.3. Hyper parameter for RL training

Hyper parameter	**Values**
Learning rate	5e-4
Batch size	256
epsilon	decay from 0.1 to 0.01
Target Q network update frequency	every 10000 steps
Discount factor	0.99

Figure 4.15. illustrates the mean curve of epoch scores during the training procedure (sliding average per 100 epochs). At the beginning of training, the average score is negative because the attack is easily identified by BDD or causes the measurement data to exceed the device limit constraints (e.g., the SoC is above 100% or the battery voltage or current exceeds the maximum range), resulting in penalty as describe in Eq. (4.25). As training progresses, the episode reward gradually rises and tends to converge after 17,500 episodes since the model gradually learns to achieve the attack target without triggering BDD motivated by the reward function in Eq. (4.25).

Figure 4.16 depicts several examples of attacks after training. Generally, the attack can be categorized into three types: maximum-underestimate, maximum-overestimate, and goal achieved. It should be noted that the attack type is not specified in advance but rather determined by the agent based on the SoC target and environmental feedback. When the target SoC is

significantly lower than the feasible SoC, the model will attempt to reduce the SoC as much as possible within the legal range allowed by BDD, as illustrated in 4.16. (a). Similarly, the model maximizes the estimate of SoC if the target SoC is excessively higher than the feasible SoC, as seen in 4.16. (b). If the target SoC is within a reachable range, the RL agent will launch an FDI attack to finally achieve the target SoC value at the end of the episode, as depicted in Figure 4.16. (c)

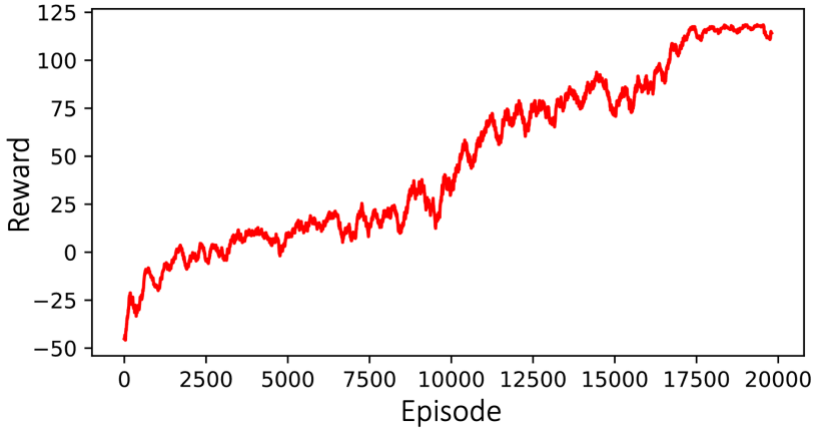
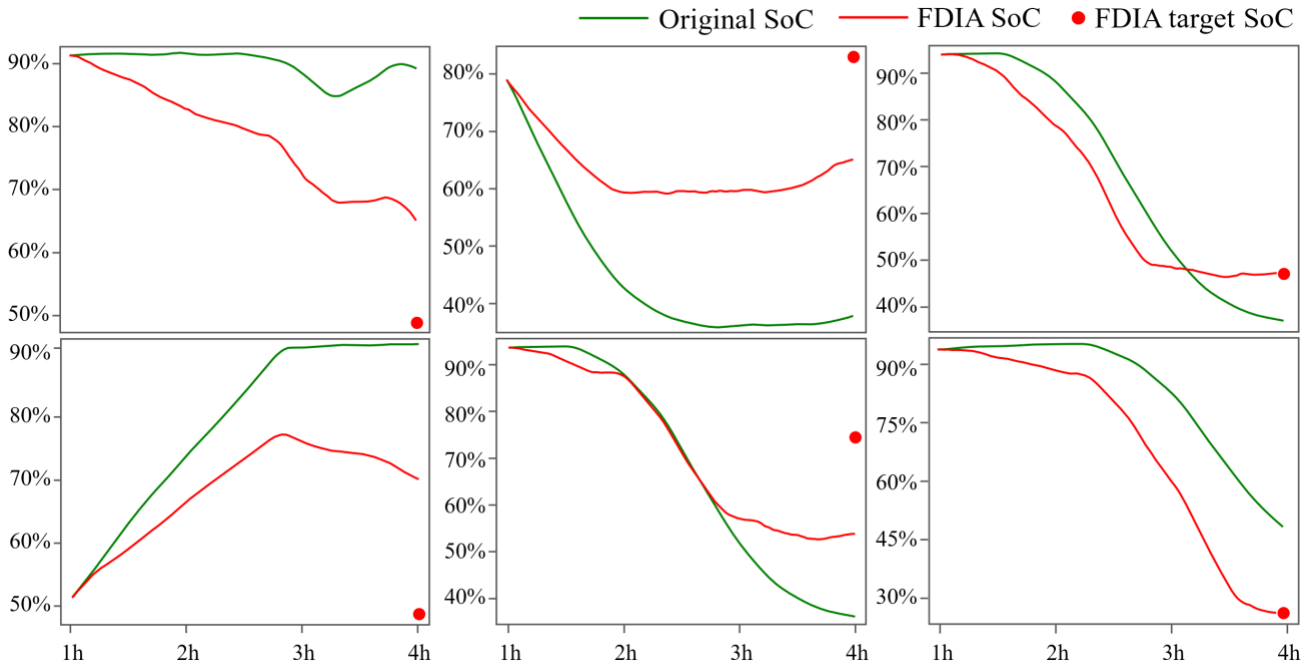


Figure 4.15. Stealth FDIA attacker RL training process



(a) max-underestimate (b) max-overestimate (c) goal achieved

Figure 4.16. Examples of RL-based stealth FDIA

4.4.3 Stealth FDIA Against SoC Detection Using TGCN

(1) Training setup

The data sets for training detection algorithms are generated based on the well-trained RL attackers. Over the span of 31 days, the attackers launched 93 attacks, each lasting four hours, at random. The data has a resolution of 5 minutes. For the TGCN model, each sample is constructed sequentially along with the measurement data frames with the previous one hour (12 frames for each sample). All data samples are randomly shuffled, with 70% used for training, 15% for validation, and 15% for testing. The model parameters that perform the best on the validation set are then utilized on the test set. The AE model from [92] is trained and evaluated in order to compare the proposed Grid-TGCN model with the current state-of-the-art algorithm. In addition, the TGCN model without the GRU layer (named as GCN) is employed for training and evaluation to demonstrate the impact of temporal characteristics. Table 4.4 provides a summary of the training hyperparameters.

Table 4.4. Hyper parameters for FDIA detection model training

Hyperparameter	**Values**
Learning rate	1e-3
Batch size	16
Optimizer	Adam
Training epochs	100
Num of GCN layers	5
GCN feature size	24/16/2/16/24
Num of GRU layers	1
GRU feature size	8

(2) Training and metrics evaluation

Figure 4.17 and Table 4.5 depict the loss function curves and metrics evaluation. It can be seen that AE performs slightly better than the GCN model without GRU layers, but it should be noticed that the model parameters number of GCN model is only 1.13K, which is only about 5%

of AE model parameters. Grid-TGCN has a considerably better loss performance than the other models. In addition, the higher accuracy and F1 score of Grid-TGCN model illustrate that the proposed algorithm can not only preserves the advantages of lightweight structure, but also significantly improves the stealth FDIA detection capability, as shown by the evaluation of metrics in Table 4.5.

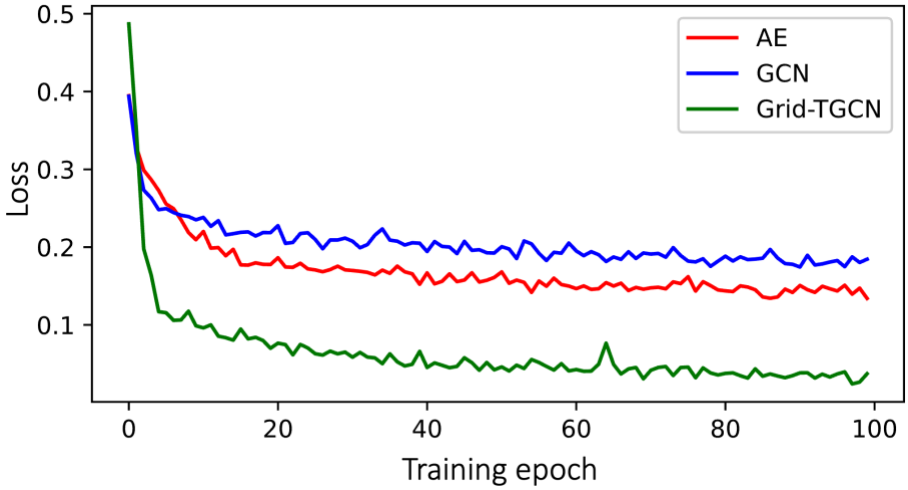


Figure 4.17. Training loss of FDIA detection algorithms

Table 4.5. Metrics evaluation of stealth FDIA detection algorithms

Metrics	**AE**[92]	**GCN**	**TGCN**
Accuracy	96.83%	95.37%	**99.51%**
F1 score	0.966	0.949	**0.995**
Parameters (10^3)	22.93	**1.13**	1.49

4.4 Conclusion

To mitigate the risk of FDIA against the battery SOC of the distribution system with BESS integration, a stealth FDIA generation technique based on reinforcement learning and an FDIA detection algorithm based on a temporal graph convolution network are presented.

First, in order to resolve the contradiction between the high computational cost of nonlinear stealth FDI constraints and real-time online deployment, a reinforcement learning

algorithm based on deep Q-learning was designed. This algorithm then interacted with the simulation environment, which contains SE-based BDD as well as the EKF-based SoC observer. Through these interactions, the RL agent learns to generate stealth SoC FDIA that can pass BDD detection.

Then, this study proposes a Grid-TGCN model to detect Stealth FDIA, a deep learning model that combines GCN and GRU that is able to fully mine and utilize the graph characteristics of the power grid as well as the temporal characteristics of the time-series measurement. The results of the experiments show that the proposed Grid-TGCN model not only outperforms the state-of-the-art algorithm in terms of accuracy and F1 score, but it also maintains very few model parameters, which provides a very promising solution for the deployment and application of large-scale complex power grids.

CHAPTER 5 SUMMARY AND FUTURE WORK

5.1 Summary of Previous and Current Research

5.1.1 Load Profile Super Resolution

In this work, ProfileSR-GAN, a two-stage GAN-based method, is proposed for solving LPSR problems. In the first stage, a GAN-based model is trained to restore the high-frequency components from the low-resolution data. In the second stage, a polishing network is developed to remove unrealistic power fluctuations in the GAN generated high-resolution load profiles. Compared with conventional up-sampling methods, such as interpolation and CNN-based methods, the proposed ProfileSR-GAN achieves superior performance in restoring high-frequency components inside sampling intervals. The overall performance improvements attribute to three aspects: the adversarial training of the GAN-based model, the inclusion of weather data, and the fine-tuning of the polishing network.

The simulation results demonstrate that ProfileSR-GAN achieved 36%-62% improvements in shape-related evaluation metrics compared with the baseline method (i.e., the linear interpolation method). An application of ProfileSR-GAN is presented as a case study to demonstrate that applying ProfileSR-GAN on upsampling can benefit downstream tasks that require the use of high-resolution load profiles. Simulation results show that when using ProfileSR-GAN to upsample the low-resolution profiles before conducting NILM, appliance-level activities can be better recognized by the NILM algorithms.

5.1.2 Customer Baseline Load Estimation

In this work, a novel deep-learning model Load-PIN is proposed to solve the missing data restoration and CVR baseload estimation problem. Load-PIN merges Gated Convolution and Multi-head self-attention mechanisms into the GAN based framework to enhance the estimation

accuracy. Load-PIN is trained using the dynamic-masking strategy so that it can handle CVR events with varying durations. We first demonstrate that at higher load aggregation levels, higher data resolution can achieve better estimation accuracy. In general, 5-min and 15-minute resolutions are sufficient for feeder-level studies. Next, we demonstrate that the Load-PIN model can achieve a 15-30% accuracy improvement under the suggested data granularity, compared with five benchmarking methods. Using the trained Load-PIN model for CVR baseline identification, we computed the CVR factors for CVR programs with fixed- and variable- CVR durations. We show that CVR can achieve load reduction in the first 1 hour. However, after 1.5 hours, the CVR effect starts to diminish, and a pay-pack period can be observed. This may cause unexpected load peaks in post-CVR periods. From the results, we want to make two recommendations. First, the CVR execution duration should be less than 2 hours. Second, feeders with high penetration of thermostatically controlled loads may not be good candidates for prolonged CVR programs.

5.1.3 Attack and Anomaly Detection for Smart Distribution Grid With BESS

To mitigate the risk of false data injection attacks on the battery SOC of the distribution system with BESS integration, a reinforcement learning-based stealth FDIA generation approach and a TGCN-based detection algorithm are proposed. First, an RL technique based on a Deep Q learning agent was designed to resolve the contradiction between the high computing cost of nonlinear stealth FDI restrictions and real-time online deployment. SE-based BDD and EKF-based SoC observers are designed in the simulation environment. Through the interactions with the environment, the RL agent learns to launch stealth SoC FDIA that avoids BDD detection. Then, Grid-TGCN is a deep learning model that combines GCN and GRU to extract and leverage the graph information of power grid and temporal features of timeseries measurement's

for FDIA detection. Experiments show that the Grid-TGCN model outperforms the state-of-the-art method in terms of accuracy and F1 score while maintaining a light model structure, making it a promising solution for large-scale complex power grid deployment and application.

5.2 Vision and Plan of Future Work

The current state of deep learning in terms of theory and applications represents the initial stage of artificial intelligence. Even while it has found success in power system areas, it still faces theoretical, technological, economic, social, and ethical obstacles. As a black-box system, for instance, it is difficult for the deep learning model to have interpretability and solid constraint guarantees, which are crucial for power systems due to the fact that the operation of power grids often requires explicit security bounds. At the same time, there are a large number of data-driven tasks that do not involve safe operations that can be solved by deep learning methods in power systems.

The research works in the three subfields will be expanded and progressed respectively. The following is a summary of the sub-areas future research plan:

For the load profile super resolute study, the proposed super-resolution algorithm for load curves can be extended to the application of other power system data. For example, the generation of high-resolution irradiance curves can help provide more realistic PV output fluctuations. Similarly, the load profile inpainting for missing load data restoration can also be extended to other power system data recovery.

The next step of research on FDIA can refer to the framework of GAN, which combines the training of the attacker and the detector, allowing them to learn and evolve from the feedback of the opponent. By adjusting the training parameters to balance the two roles, it is hopeful of

designing super-attackers that can fool traditional and deep learning-based detection algorithms or super-detector with extraordinary recognition ability.

REFERENCES

[1] W. Gan *et al.*, "Two-stage planning of network-constrained hybrid energy supply stations for electric and natural gas vehicles," *IEEE Transactions on Smart Grid,* vol. 12, no. 3, pp. 2013-2026, 2020.

[2] K. Hou *et al.*, "A reliability assessment approach for integrated transportation and electrical power systems incorporating electric vehicles," *IEEE Transactions on Smart Grid,* vol. 9, no. 1, pp. 88-100, 2016.

[3] L. Xie, C. Singh, S. K. Mitter, M. A. Dahleh, and S. S. J. J. Oren, "Toward carbon-neutral electricity and mobility: Is the grid infrastructure ready?," *Joule,* vol. 5, no. 8, pp. 1908-1913, 2021.

[4] S. Rahman *et al.*, "Analysis of power grid voltage stability with high penetration of solar PV systems," *IEEE Transactions on Industry Applications,* vol. 57, no. 3, pp. 2245-2257, 2021.

[5] K. Nasrollahi and T. B. Moeslund, "Super-resolution: a comprehensive survey," *Machine Vision and Applications,* vol. 25, no. 6, pp. 1423-1468, 2014.

[6] R. Morin, A. Basarab, and D. Kouamé, "Alternating direction method of multipliers framework for super-resolution in ultrasound imaging," in *2012 9th IEEE International Symposium on Biomedical Imaging (ISBI)*, 2012, pp. 1595-1598: IEEE.

[7] H. K. Aghajan and T. Kailath, "Sensor array processing techniques for super resolution multi-line-fitting and straight edge detection," *IEEE Transactions on Image Processing,* vol. 2, no. 4, pp. 454-465, 1993.

[8] K. Nguyen, S. Sridharan, S. Denman, and C. Fookes, "Feature-domain super-resolution framework for Gabor-based face and iris recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 2642-2649: IEEE.

[9] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," *arXiv preprint arXiv:.00853,* 2017.

[10] G. Liu, J. Gu, J. Zhao, F. Wen, and G. Liang, "Super Resolution Perception for Smart Meter Data," *Information Sciences,* 2020.

[11] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister, and M.-H. Yang, "Learning to super-resolve blurry face and text images," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 251-260.

[12] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems,* vol. 27, pp. 2672-2680, 2014.

[13] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1646-1654.

[14] Z. Wang, T. Hong, and Buildings, "Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN)," *Energy,* vol. 224, p. 110299, 2020.

[15] Y. Gu, Q. Chen, K. Liu, L. Xie, and C. Kang, "Gan-based model for residential load generation considering typical consumption patterns," in *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2019, pp. 1-5: IEEE.

[16] Y. Zhang, Q. Ai, F. Xiao, R. Hao, T. Lu, and E. Systems, "Typical wind power scenario generation for multiple wind farms using conditional improved Wasserstein generative adversarial network," *International Journal of Electrical Power,* vol. 114, p. 105388, 2020.

[17] Y. Wang, G. Hug, Z. Liu, and N. Zhang, "Modeling load forecast uncertainty using generative adversarial networks," *Electric Power Systems Research*, vol. 189, p. 106732, 2020.
[18] A. Harell, R. Jones, S. Makonin, and I. V. Bajić, "TraceGAN: Synthesizing Appliance Power Signatures Using Generative Adversarial Networks," *IEEE Transactions on Smart Grid*, 2021.
[19] M. Kaselimi, A. Voulodimos, E. Protopapadakis, N. Doulamis, and A. Doulamis, "Energan: A generative adversarial network for energy disaggregation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 1578-1582: IEEE.
[20] Y. Chen *et al.*, "Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings," vol. 195, pp. 659-670, 2017.
[21] F. Wang, K. Li, C. Liu, Z. Mi, M. Shafie-Khah, and J. P. J. I. T. o. S. G. Catalão, "Synchronous pattern matching principle-based residential demand response baseline estimation: Mechanism analysis and approach description," vol. 9, no. 6, pp. 6972-6985, 2018.
[22] F. Wang *et al.*, "Smart households' aggregated capacity forecasting for load aggregators under incentive-based demand response programs," vol. 56, no. 2, pp. 1086-1097, 2020.
[23] Y. Weng, J. Yu, R. J. I. J. o. E. P. Rajagopal, and E. Systems, "Probabilistic baseline estimation based on load patterns for better residential customer rewards," vol. 100, pp. 508-516, 2018.
[24] A. B. Alhassan, X. Zhang, H. Shen, H. J. I. J. o. E. P. Xu, and E. Systems, "Power transmission line inspection robots: A review, trends and challenges for future research," vol. 118, p. 105862, 2020.
[25] L. Hatton, P. Charpentier, and E. J. I. T. o. P. S. Matzner-Løber, "Statistical estimation of the residential baseline," vol. 31, no. 3, pp. 1752-1759, 2015.
[26] X. Ge *et al.*, "Spatio-Temporal Two-Dimensions Data Based Customer Baseline Load Estimation Approach Using LASSO Regression," 2022.
[27] M. Sun, Y. Wang, F. Teng, Y. Ye, G. Strbac, and C. J. I. T. o. S. G. Kang, "Clustering-based residential baseline estimation: A probabilistic perspective," vol. 10, no. 6, pp. 6014-6028, 2019.
[28] S. Matsukawa, C. Ninagawa, J. Morikawa, T. Inaba, and S. Kondo, "Stable segment method for multiple linear regression on baseline estimation for smart grid fast automated demand response," in *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*, 2019, pp. 2571-2576: IEEE.
[29] Y. Zhang, W. Chen, R. Xu, and J. J. I. T. o. s. g. Black, "A cluster-based method for calculating baselines for residential loads," vol. 7, no. 5, pp. 2368-2377, 2015.
[30] S. Park, S. Ryu, Y. Choi, J. Kim, and H. J. E. Kim, "Data-driven baseline estimation of residential buildings for demand response," vol. 8, no. 9, pp. 10239-10259, 2015.
[31] J. Oyedokun, S. Bu, Z. Han, and X. Liu, "Customer baseline load estimation for incentive-based demand response using long short-term memory recurrent neural network," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, 2019, pp. 1-5: IEEE.
[32] Y. Chen *et al.*, "Privacy-Preserving Baseline Load Reconstruction for Residential Demand Response Considering Distributed Energy Resources," 2021.

[33] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, 2017, pp. 933-941: PMLR.
[34] A. Vaswani *et al.*, "Attention is all you need," vol. 30, 2017.
[35] Y. Wang, Q. Chen, T. Hong, and C. J. I. T. o. S. G. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," vol. 10, no. 3, pp. 3125-3148, 2018.
[36] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681-4690.
[37] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson, "Super-resolved surface reconstruction from multiple images," in *Maximum Entropy and Bayesian Methods*: Springer, 1996, pp. 293-308.
[38] B. Schölkopf, A. J. Smola, and F. Bach, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.
[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
[40] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:.03167*, 2015.
[41] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:.06434*, 2015.
[42] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, pp. 2234-2242, 2016.
[43] M. A. Ranzato, Y.-L. Boureau, and Y. LeCun, "Sparse feature learning for deep belief networks," *Advances in neural information processing systems*, vol. 20, pp. 1185-1192, 2007.
[44] S. R. Bulo, G. Neuhold, and P. Kontschieder, "Loss max-pooling for semantic image segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7082-7091: IEEE.
[45] (2014). *Pecan Street Dataport*. Available: https://www.pecanstreet.org/dataport/
[46] V. C. Corp. Visual Crossing Weather Services [Online]. Available: https://www.visualcrossing.com/weather/weather-data-services#/login
[47] D. P. Kingma and J. J. a. p. a. Ba, "Adam: A method for stochastic optimization," 2014.
[48] U. Ramer and i. processing, "An iterative procedure for the polygonal approximation of plane curves," *Computer graphics*, vol. 1, no. 3, pp. 244-256, 1972.
[49] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of machine learning research*, vol. 9, no. 11, 2008.
[50] J. Kelly and W. Knottenbelt, "Neural nilm: Deep neural networks applied to energy disaggregation," in *Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments*, 2015, pp. 55-64.
[51] C. Zhang, M. Zhong, Z. Wang, N. Goddard, and C. Sutton, "Sequence-to-point learning with neural networks for non-intrusive load monitoring," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018, vol. 32, no. 1.

[52] N. Batra *et al.*, "NILMTK: An open source toolkit for non-intrusive load monitoring," in *Proceedings of the 5th international conference on Future energy systems*, 2014, pp. 265-276.

[53] C. Klemenjak and P. Goldsborough, "Non-intrusive load monitoring: A review and outlook," *arXiv preprint arXiv:.01191,* 2016.

[54] Z. Wang and J. Wang, "Review on implementation and assessment of conservation voltage reduction," *IEEE Transactions on Power Systems,* vol. 29, no. 3, pp. 1306-1315, 2013.

[55] Y. Zhang, S. Ren, Z. Y. Dong, Y. Xu, K. Meng, and Y. Zheng, "Optimal placement of battery energy storage in distribution networks considering conservation voltage reduction and stochastic load composition," *IET Generation, Transmission & Distribution,* vol. 11, no. 15, pp. 3862-3870, 2017.

[56] Z. Wang and J. Wang, "Time-varying stochastic assessment of conservation voltage reduction based on load modeling," *IEEE Transactions on Power Systems,* vol. 29, no. 5, pp. 2321-2328, 2014.

[57] M. Diaz-Aguiló *et al.*, "Field-validated load model for the analysis of CVR in distribution secondary networks: Energy conservation," *IEEE Transactions on Power Delivery,* vol. 28, no. 4, pp. 2428-2436, 2013.

[58] K. P. Schneider, J. C. Fuller, F. K. Tuffner, and R. Singh, "Evaluation of conservation voltage reduction (CVR) on a national level," Pacific Northwest National Lab.(PNNL), Richland, WA (United States)2010.

[59] K. Coughlin, M. A. Piette, C. Goldman, and S. Kiliccote, "Estimating demand response load impacts: Evaluation of baselineload models for non-residential buildings in california," Lawrence Berkeley National Lab.(LBNL), Berkeley, CA (United States)2008.

[60] B. Xiang, K. Li, X. Ge, F. Wang, J. Lai, and P. Dehghanian, "Smart Households' Available Aggregated Capacity Day-ahead Forecast Model for Load Aggregators under Incentive-based Demand Response Program," in *2019 IEEE Industry Applications Society Annual Meeting*, 2019, pp. 1-10: IEEE.

[61] T. K. Wijaya, M. Vasirani, and K. Aberer, "When bias matters: An economic assessment of demand response baselines for residential customers," *IEEE Transactions on Smart Grid,* vol. 5, no. 4, pp. 1755-1763, 2014.

[62] Z. Wang, M. Begovic, and J. Wang, "Analysis of conservation voltage reduction effects based on multistage SVR and stochastic process," *IEEE Transactions on Smart Grid,* vol. 5, no. 1, pp. 431-439, 2013.

[63] J. Yoon, J. Jordon, and M. Schaar, "Gain: Missing data imputation using generative adversarial nets," in *International conference on machine learning*, 2018, pp. 5689-5698: PMLR.

[64] Y. Luo, X. Cai, Y. Zhang, and J. Xu, "Multivariate time series imputation with generative adversarial networks," *Advances in neural information processing systems,* vol. 31, 2018.

[65] K. Zhang, X. Dou, and X. Xiao, "Grid Defect Data Completion Based on Generative Adversarial Imputation Nets," in *2021 IEEE Sustainable Power and Energy Conference (iSPEC)*, 2021, pp. 952-957: IEEE.

[66] W. Zhang, Y. Luo, Y. Zhang, and D. Srinivasan, "SolarGAN: Multivariate solar data imputation using generative adversarial network," *IEEE Transactions on Sustainable Energy,* vol. 12, no. 1, pp. 743-746, 2020.

[67] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4471-4480.

[68] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[69] G. Dudek, "Multilayer perceptron for short-term load forecasting: from global to local approach," *Neural Computing and Applications*, vol. 32, no. 8, pp. 3695-3707, 2020.

[70] J. Zheng, C. Xu, Z. Zhang, and X. Li, "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," in *2017 51st Annual conference on information sciences and systems (CISS)*, 2017, pp. 1-6: IEEE.

[71] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[72] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *2013 IEEE workshop on automatic speech recognition and understanding*, 2013, pp. 273-278: IEEE.

[73] J. Gehring, Y. Miao, F. Metze, and A. Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *2013 IEEE international conference on acoustics, speech and signal processing*, 2013, pp. 3377-3381: IEEE.

[74] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2218-2234, 2019.

[75] A. de la Villa Jaén, E. Acha, and A. G. Expósito, "Voltage source converter modeling for power system state estimation: STATCOM and VSC-HVDC," *IEEE transactions on power systems*, vol. 23, no. 4, pp. 1552-1559, 2008.

[76] A. Monticelli, "Electric power system state estimation," *Proceedings of the IEEE*, vol. 88, no. 2, pp. 262-282, 2000.

[77] B. Stott, J. Jardim, and O. Alsaç, "DC power flow revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290-1300, 2009.

[78] H. M. Merrill and F. C. Schweppe, "Bad data suppression in power system static state estimation," *IEEE Transactions on Power Apparatus and Systems*, no. 6, pp. 2718-2725, 1971.

[79] M. Daowd, N. Omar, P. Van Den Bossche, and J. Van Mierlo, "Passive and active battery balancing comparison based on MATLAB simulation," in *2011 IEEE Vehicle Power and Propulsion Conference*, 2011, pp. 1-7: IEEE.

[80] P. Zhuang and H. Liang, "False data injection attacks against state-of-charge estimation of battery energy storage systems in smart distribution networks," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2566-2577, 2020.

[81] Z. Chen, Y. Fu, and C. C. Mi, "State of charge estimation of lithium-ion batteries in electric drive vehicles using extended Kalman filtering," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 3, pp. 1020-1030, 2012.

[82] S. Lee, J. Kim, J. Lee, and B. H. Cho, "State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge," *Journal of power sources*, vol. 185, no. 2, pp. 1367-1373, 2008.

[83] A. Monticelli and A. Garcia, "Reliable bad data processing for real-time state estimation," *IEEE transactions on power apparatus and systems*, no. 5, pp. 1126-1139, 1983.

[84] E. Handschin, F. C. Schweppe, J. Kohlas, and A. Fiechter, "Bad data analysis for power system state estimation," *IEEE Transactions on Power Apparatus and Systems,* vol. 94, no. 2, pp. 329-337, 1975.

[85] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC),* vol. 14, no. 1, pp. 1-33, 2011.

[86] B. Stott, J. Jardim, and O. J. I. T. o. P. S. Alsaç, "DC power flow revisited," vol. 24, no. 3, pp. 1290-1300, 2009.

[87] Z. Zhang, D. Zhang, and R. C. Qiu, "Deep reinforcement learning for power system applications: An overview," *CSEE Journal of Power and Energy Systems,* vol. 6, no. 1, pp. 213-225, 2019.

[88] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning,* vol. 8, no. 3, pp. 279-292, 1992.

[89] M. Srinivasan, V. J. Kotagi, and C. S. R. Murthy, "A Q-learning framework for user QoE enhanced self-organizing spectrally efficient network using a novel inter-operator proximal spectrum sharing," *IEEE Journal on Selected Areas in Communications,* vol. 34, no. 11, pp. 2887-2901, 2016.

[90] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence,* 2016, vol. 30, no. 1.

[91] M. M. Aboelwafa, K. G. Seddik, M. H. Eldefrawy, Y. Gadallah, and M. Gidlund, "A machine-learning-based technique for false data injection attacks detection in industrial IoT," *IEEE Internet of Things Journal,* vol. 7, no. 9, pp. 8462-8471, 2020.

[92] J. Wang, D. Shi, Y. Li, J. Chen, H. Ding, and X. J. I. T. o. s. g. Duan, "Distributed framework for detecting PMU data manipulation attacks with deep autoencoders," vol. 10, no. 4, pp. 4401-4410, 2018.

[93] X. Niu, J. Li, J. Sun, and K. Tomsovic, "Dynamic detection of false data injection attack in smart grid using deep learning," in *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 2019, pp. 1-6: IEEE.

[94] Y. He, G. J. Mendis, and J. Wei, "Real-time detection of false data injection attacks in smart grid: A deep learning-based intelligent mechanism," *IEEE Transactions on Smart Grid,* vol. 8, no. 5, pp. 2505-2516, 2017.

[95] Y. Ding, K. Ma, T. Pu, X. Wang, R. Li, and D. Zhang, "A deep learning-based classification scheme for false data injection attack detection in power system," *Electronics,* vol. 10, no. 12, p. 1459, 2021.

[96] Y. Zhang, J. Wang, and B. Chen, "Detecting false data injection attacks in smart grids: A semi-supervised deep learning approach," *IEEE Transactions on Smart Grid,* vol. 12, no. 1, pp. 623-634, 2020.

[97] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE transactions on neural networks,* vol. 20, no. 1, pp. 61-80, 2008.

[98] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia, "Learning to simulate complex physics with graph networks," in *International Conference on Machine Learning*, 2020, pp. 8459-8468: PMLR.

[99] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning," *arXiv preprint arXiv:1902.06673,* 2019.

[100] C. Eksombatchai *et al.*, "Pixie: A system for recommending 3+ billion items to 200+ million users in real-time," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1775-1784.
[101] O. Boyaci, M. R. Narimani, K. R. Davis, M. Ismail, T. J. Overbye, and E. Serpedin, "Joint detection and localization of stealth false data injection attacks in smart grids using graph neural networks," *IEEE Transactions on Smart Grid,* vol. 13, no. 1, pp. 807-819, 2021.
[102] Y. Li and Y. Wang, "Developing graphical detection techniques for maintaining state estimation integrity against false data injection attack in integrated electric cyber-physical system," *Journal of systems architecture,* vol. 105, p. 101705, 2020.
[103] S. Hochreiter and J. J. N. c. Schmidhuber, "Long short-term memory," vol. 9, no. 8, pp. 1735-1780, 1997.
[104] J. Chung, C. Gulcehre, K. Cho, and Y. J. a. p. a. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.
[105] B. Xu, V. Paduani, H. Yu, D. Lubkeman, and N. Lu, "A novel grid-forming voltage control strategy for supplying unbalanced microgrid loads using inverter-based resources," in *2022 IEEE Power & Energy Society General Meeting (PESGM)*, 2022, pp. 1-5: IEEE.
[106] G. Rancilio *et al.*, "Modeling a large-scale battery energy storage system for power grid application analysis," vol. 12, no. 17, p. 3312, 2019.
[107] E. Reihani, S. Sepasi, L. R. Roose, M. J. I. J. o. E. P. Matsuura, and E. Systems, "Energy management at the distribution grid using a Battery Energy Storage System (BESS)," vol. 77, pp. 337-344, 2016.