ABSTRACT

HU, YI. Generative AI for Power System Data Analysis. (Under the direction of Dr. Ning Lu).

The digital transformation of modern power systems has created a critical dependence on data for real-time operation, planning, and decision-making. However, the effective use of smart meter and synchrophasor data is often limited by challenges such as data scarcity, incompleteness, and privacy constraints. These issues hinder the deployment of advanced machine learning and analytics tools, particularly in contexts requiring high-quality, diverse, and robust datasets.

This dissertation explores the use of Generative Artificial Intelligence to address data limitations in power system applications. It presents a series of solutions that leverage state-of-the-art generative models to synthesize realistic data, restore corrupted or missing measurements, and enhance event detection capabilities.

First, a GAN-based framework is proposed to generate synthetic load profile groups that preserve both temporal dynamics and spatial correlations across customers. These synthetic datasets replicate the characteristics of real data, overcoming both privacy concerns and the challenges posed by limited data availability. Second, a deep learning approach based on BERT architecture is developed for restoring missing segments in smart meter data. By learning contextual dependencies within the data, it outperforms traditional imputation methods in accuracy. Third, this work investigates the fine-tuning of pre-trained Large Language Models for time series restoration tasks in power systems, demonstrating comparable accuracy but requires significantly less training data and computing resources. Lastly, a generative AI-powered methodology is introduced to detect events in synchrophasor data, capable of identifying both known and unseen grid disturbances with improved interpretability and resilience.

Together, these contributions advance the state of the art in AI-powered grid data analytics and lay the foundation for more robust, intelligent, and data-rich power system applications. This work supports the long-term vision of enabling scalable, secure, and adaptive energy systems through generative AI innovations.

Generative AI for Power System Data Analysis

by
Yi Hu

A dissertation submitted to 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

2025

APPROVED BY:

Dr. Ning Lu

Committee Chair

Dr. Wenyuan Tang

Dr. Anderson Rodrigo de Queiroz

Dr. Kaixiong Zhou

BIOGRAPHY

Yi Hu is a Ph.D. candidate in Electrical Engineering at North Carolina State University and a member of the FREEDM Systems Center. His doctoral research focuses on applying generative artificial intelligence to power system data challenges, including synthetic data generation, missing data restoration, and event detection.

He received the M.S. degree in Electrical Engineering from Peking University, China, in 2018, and the B.E. degree from Chongqing University of Posts and Telecommunications, China, in 2014. His work has contributed to multiple academic publications and collaborative projects with industry partners. He also serves as Secretary of the IEEE PES Task Force on Power System Synthetic Data Generation and Sharing.

ACKNOWLEDGMENTS

First and foremost, I would like to express my heartfelt gratitude to my advisor, Dr. Ning Lu, for her strong support, insightful guidance, and generous mentorship throughout my Ph.D. journey. Her deep expertise has shaped the style of my research, and her patience and encouragement have been instrumental every step of the way. I feel incredibly fortunate to have had the opportunity to work with and learn from her at NC State. Dr. Lu, you will always be a lifelong mentor to me.

I would also like to extend my sincere appreciation to Dr. Wenyuan Tang, Dr. Kaixiong Zhou, and Dr. Anderson Rodrigo de Queiroz for serving on my advisory committee. Their thoughtful feedback and constructive insights have significantly enriched the quality of this dissertation. I am honored to have had the opportunity to learn from their profound expertise and perspectives.

I am thankful to all my colleagues and friends at the FREEDM Systems Center. Their collaboration, support, and engaging discussions have played a vital role in my development as a researcher. In particular, I would like to acknowledge, listed in alphabetical order, Eyad Fahad Alsulayhim, Rongxing Hu, Hyeonjin Kim, Hanpyo Lee, Yiyan Li, Lidong Song, Jongha Woo, Qi Xiao, Kai Ye, and Si Zhang, for their camaraderie and support throughout this journey.

Finally, my deepest thanks go to my family. To my wife, Liannian Wang, thank you for standing by me through every challenge, your strength, love, and belief in me have been my greatest source of inspiration. I am also especially grateful to my son, Ocean Hu. Welcoming him into the world during my Ph.D. journey has been the most joyful and meaningful achievement of all, and truly, he is the best "model" I have ever trained.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Background and Motivation

The increasing complexity and digitalization of modern power systems have created unprecedented opportunities and challenges in system operation, monitoring, and planning. As renewable energy sources, electric vehicles, and inverter-based resources (IBRs) are integrated at scale, power systems must evolve to remain reliable, efficient, and resilient. Central to this transformation is the availability and quality of data, which plays a critical role in enabling intelligent grid applications ranging from state estimation to event detection and fault recovery.

However, the power industry continues to face significant data-related challenges, including scarcity of real-world data, missing or corrupted measurements, and data privacy constraints. High-resolution data from smart meters and synchrophasor systems offer promise, but their effective use is often limited by issues such as incomplete coverage, communication errors, or the sensitivity of operational data. These limitations hinder the deployment of advanced algorithms and decision-support tools that require large-scale, diverse, and high-quality datasets.

Recent advances in Generative Artificial Intelligence (Generative AI), including Generative Adversarial Networks (GANs), Transformer-based models, and Large Language Models (LLMs), offer transformative solutions to address these challenges. Generative AI provides new pathways for synthetic data generation, missing data restoration, and data-driven grid intelligence, with the potential to significantly enhance the accuracy, robustness, and interpretability of power system analytics.

1.3 Dissertation Objectives

This dissertation focuses on developing generative AI-powered solutions to address data scarcity, quality, and usability challenges in power systems. Specifically, it explores methods to:

- Generate realistic synthetic load profile groups considering spatial and temporal correlations,
- Restore missing segments in smart meter load data using BERT-based deep learning architectures,

- Fine-tune large pre-trained language models for enhanced missing data recovery, and
- Apply generative AI to detect and classify power system events using synchrophasor (PMU) data.

The contributions of this research aim to support the creation of reliable, privacy-preserving, and AI-compatible datasets and models that can empower both operational and planning applications in future grid environments.

1.4 Structure of the Dissertation

The remainder of this dissertation is organized as follows:

- **Chapter 2** presents a method for generating synthetic load profile groups that account for spatial-temporal correlations in power systems.
- **Chapter 3** focuses on a BERT-based approach for restoring missing load data, addressing the challenges of incomplete datasets.
- **Chapter 4** explores the fine-tuning of ChatGPT for missing data restoration, comparing its performance with specialized architecture.
- **Chapter 5** investigates a generative AI framework for power system event detection using PMU data, addressing both known and novel event types.
- **Chapter 6** concludes the dissertation by summarizing the findings and discussing future research directions in generative AI for power system analysis.

2. SYNTHETIC LOAD PROFILE GROUP GENERATION

2.1 Background

Smart meter data are essential in power distribution system analysis, for instance, modeling load behaviors, conducting renewable integration studies, and developing demand response programs. However, due to security and privacy considerations, utilities cannot share a large amount of smart meter data with the research community for carrying out such analysis. Therefore, using synthetic load profiles (SLPs) derived from smart meter data becomes an increasingly attractive solution.

SLPs are generated load profiles bearing similar characteristics as the real ones. In general, there are two approaches for generating SLPs: simulation-based and data-driven. Table 2-1 summarizes the advantages and disadvantages of the state-of-the-art SLP generation methods and compares our algorithm with the existing ones. As can be seen in the table, up till now, all existing generative methods generate SLPs one at a time. There is no generative method proposed for generating a group of SLPs served by the same distribution transformer or the same feeder, where the SLPs have strong spatial-temporal correlations.

Note that such spatial-temporal correlations exist because consumers at the same geographical location experience similar weather conditions and share similar demographical characteristics (e.g., house type, income level, and living pattern), making the weather-dependent loads and consumption patterns have similar variations. Therefore, the time-series load profiles served by the same transformer or feeder exhibit distinct group-level characteristics. By simply selecting load profiles randomly from a database or generating load profiles for each user one at a time to form a load group, one cannot capture such group-level spatial-temporal correlations. Some forecasting methods based on copula select load profiles by influence factor to form a group, as shown in previous studies [13][14]. These methods typically rely on historical data and involve a two-step forecasting and selecting process. However, copula alone cannot fully capture the spatial and temporal correlations between loads served by the same distribution transformer or feeder. This is because high-level hidden features are usually difficult to explicitly formulate and can only be learned using Deep Learning techniques.

Table 2-1 Comparison of Multiload-GAN Model with State-of-The-Art Generative Methods

<table>
<tr><th colspan="2"></th><th>Description</th><th>Advantages</th><th>Disadvantages</th><th>Model output</th></tr>
<tr><td colspan="2">Model-based methods [1][2]</td><td>Use physical models, such as building thermodynamics and customer behavioral models, to simulate electricity consumption profiles.</td><td>Explainable as the models reflect the laws of physics when describing the behavior behind field measurements</td><td>Require detailed physics-based models with many inputs and require parameter tuning.</td><td rowspan="4">Single load profile

(When generating a load profile, the methods do not consider the spatial-temporal correlations among a group of generated load profiles)</td></tr>
<tr><td rowspan="4">Data-driven methods</td><td>Clustering based [3][4]</td><td>Cluster existing load profiles into different categories so that by combining the load profiles across different categories, SLPs are generated.</td><td>Easy to implement and can represent some realistic load profile characteristics.</td><td>Lack of diversity when using combinations of a limited number of existing profiles.</td></tr>
<tr><td>Forecasting based [5]-[8]

(the benchmark method)</td><td>Generate SLPs based on publicly available load or weather data.</td><td>Easy to implement and flexible to generate load profiles with different lengths and granularities.</td><td>Depend heavily on historical data. The generated load profiles have similar patterns with historical data, therefore, lack of diversity.</td></tr>
<tr><td>SingleLoad-

GAN-based [10]-[12]

(the benchmark method)</td><td>GAN-based generative methods to generate the SLP for one customer at a time.</td><td>Learn from the real data distribution to generate diversified load profiles with high-frequency details.</td><td>Hard to train.</td></tr>
<tr><td>MultiLoad-GAN

(the proposed method)</td><td>GAN-based generative methods to generate a group of spatial-temporal correlated load profiles simultaneously. Such load profiles can be loads served by the same transformer or feeder.</td><td>Learn from the distribution of real data to generate diversified load profiles with high-frequency details. Preserve the spatial-temporal correlations between loads.</td><td>Hard to train.</td><td>Multiple spatial-temporal correlated load profiles</td></tr>
</table>

To bridge this gap, we propose a deep-learning framework, called the Multi-load Generative Adversarial Network (MultiLoad-GAN), to generate a group of SLPs simultaneously. The contributions are three-fold.

- First, MultiLoad-GAN captures the spatial-temporal correlations among loads in a load group to enable the generation of correlated realistic SLPs in large quantity for meeting the emerging need in microgrid and distribution system planning. This is achieved by the novel profile-to-image encoding-decoding method and the MultiLoad-GAN architecture design.
- Second, two complementary metrics for evaluating realisticness of generated load profiles are developed: computing statistics based on domain knowledge and comparing high-level features via a deep-learning classifier.
- Third, to tackle data scarcity, a novel iterative data augmentation mechanism is developed to generate training samples for enhancing the training of both the deep-learning classifier and the MultiLoad-GAN model, which can improve the performance of MultiLoad-GAN by approximately 4.07%.

The rest of the chapter is organized as follows. Section 2.2 introduces the methodology, Section 2.3 introduces the simulation results, and Section 2.4 concludes the chapter.

2.2 Synthetic Load Profile Group Generation Design and Implementation Details

In this section, we first present the terminologies used in the chapter and the overall modeling framework. Next, we present MultiLoad-GAN, statistical based evaluation metrics, the training of a classifier to assess the realisticness of generated load profile groups, and the implementation of Automatic Data Augmentation to enhance the algorithm performance.

2.2.1 Terminologies and the Modeling Framework

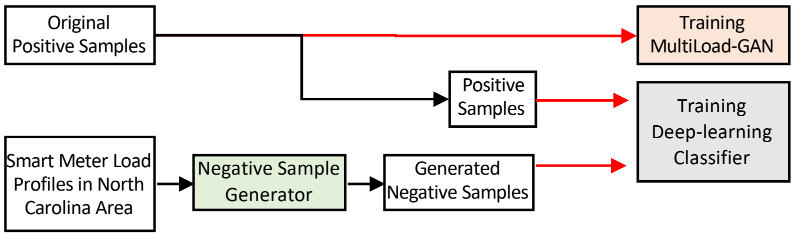
In this chapter, we define **a *load group*** as loads served by the same transformer. A "***positive sample***" is defined as a group of load profiles from customers supplied by the same transformer. The "***original positive samples***" is the labelled data set given by a utility in North Carolina area including 8 transformers with each serving 8 loads from 2017 to 2020. The "***negative samples***" is the data set that consists of groups of load profiles from customers unlikely to be supplied by the same transformer. This is a unique definition because in power distribution systems, even if a load profile is from a user supplied by another transformer, the load profile is likely to be similar to loads supplied under the same transformer. For example, in a neighborhood, serving which 8 out of 10 neighboring houses is sometimes a random choice by design engineers. In most cases, all $C(10,8)$ combinations can be considered as positive samples. Thus, in the training, the "***original positive samples***" is the ground-truth data set labelled by utility engineers while all negative samples are "generated negative samples" generated by us.

Figure 2.1 shows the overview of the framework. As shown in Figure 2.1 (a), the MultiLoad-GAN framework includes three modules: MultiLoad-GAN, Deep-learning classifier (DLC), and Negative Sample Generator (NSG). Initially, because there are no labelled negative samples for training DLC, we develop the NSG module for generating negative samples to enhance the training of the DLC.

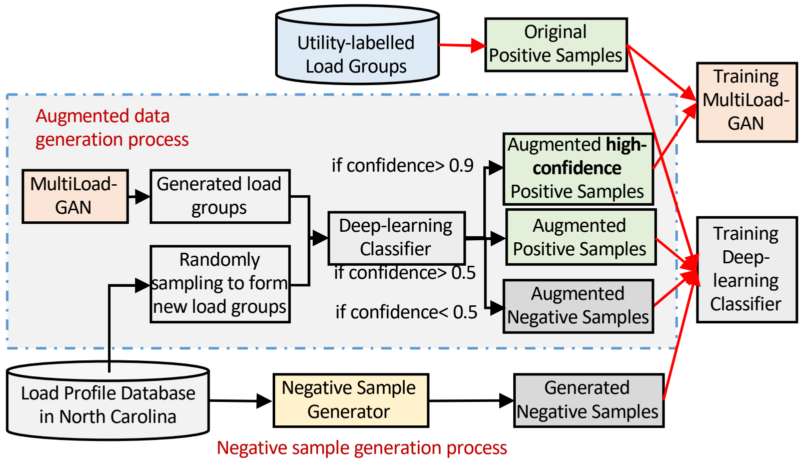
As shown in Figure 2.1 (b), due to security and privacy considerations, the amount of labelled data provided by the utility to train MultiLoad-GAN is usually insufficient. Therefore, to further improve the performance of MultiLoad-GAN, we develop an interactive process, Automatic Data Augmentation (ADA), for generating augmented labelled data, which allows the training of DLC and MultiLoad-GAN to iteratively evolve with the augmented data generation process.

As shown in Figure 2.1 (c), the realisticness of the generated load groups is evaluated by comparing the generated load groups with the "original positive samples" using two kinds of realisticness metrics: statistics metrics based on domain knowledge and a deep-learning classifier for comparing high-level features.

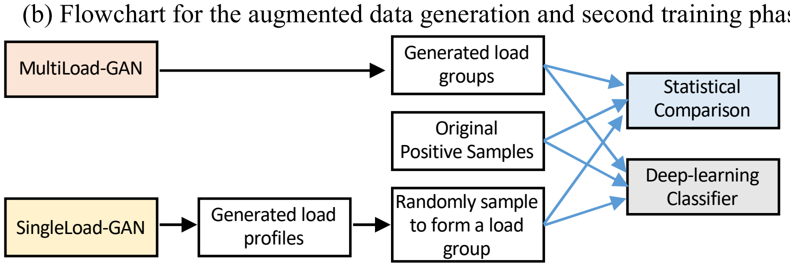
To the best of our knowledge, there is no other existing approach for generating a group of highly correlated load profiles in the literature. Thus, the goal of our comparison is to demonstrate that when an algorithm generates load profiles one at a time, it cannot generate a group of load profiles that bear correct group-level characteristics. We select SingleLoad-GAN as the benchmark model for performance comparison, because SingleLoad-GAN and MultiLoad-GAN formulate an ablation study in nature. In addition, GAN based models produce more realistic and diversified shape-wise load profile details than the other existing methods by learning the distribution of real data, as shown in [10]. Therefore, the SingleLoad-GAN reproduces the method presented in [10]. Due to differences in input data, SingleLoad-GAN uses different parameters.



(a) Flowchart for the first training phase



(b) Flowchart for the augmented data generation and second training phase



(c) Flowchart for performance evaluation and comparison

Figure 2.1 An Overview of The Overall Modeling Framework

2.2.2 GAN-based Approach

A GAN model consists of two components: a generator network (G) and a discriminator network (D). A latent vector $\mathbf{z}$**,** usually a Gaussian noise, is used as the input to generate the target output $G(\mathbf{z})$. Then, the generated data $G(\mathbf{z})$ and the real data $\mathbf{x}$ are sent to D. The goal of D is to distinguish which data samples 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 distinguish between $G(\mathbf{z})$ and $\mathbf{x}$ by assigning larger probabilities to $\mathbf{x}$ and smaller ones to $G(\mathbf{z})$. As introduced in [9], this process is formulated as a minimax game

$$\min_{G}\max_{D}\left[E_{\boldsymbol{x}\sim P_r}\left[\log\left(D(\boldsymbol{x})\right)\right]+\ E_{\widehat{\boldsymbol{x}}\sim P_g}\left[\log\left(1-D(\widehat{\boldsymbol{x}})\right)\right]\right] \tag{2.1}$$

where P_r and P_g are the probability distributions of the training data and the generated data, E is the expectation operator, and $\widehat{\boldsymbol{x}} = G(\boldsymbol{z})$. According to Wasserstein Generative Adversarial Networks (WGAN) [15][16], the training process will be more stable than the original GAN with the following loss function

$$\min_{G}\max_{D\in\omega}\left[E_{\boldsymbol{x}\sim P_r}[D(\boldsymbol{x})]-\ E_{\widehat{\boldsymbol{x}}\sim P_g}[D(\widehat{\boldsymbol{x}})]\right] \tag{2.2}$$

where ω is the set of 1-Lipschitz function. A gradient penalty method [16] is proposed to further improve the performance of WGAN. Thus, we adopt the following loss function in our framework

$$L = E_{\widehat{\boldsymbol{x}}\sim P_g}[D(\widehat{\boldsymbol{x}})] - E_{\boldsymbol{x}\sim P_r}[D(\boldsymbol{x})] + \lambda E_{\widehat{\boldsymbol{x}}\sim P_{\widehat{\boldsymbol{x}}}}[(\|\nabla_{\widehat{\boldsymbol{x}}}D(\widehat{\boldsymbol{x}})\|_2 - 1)^2] \tag{2.3}$$

where $P_{\widehat{\boldsymbol{x}}}$ is the distribution sampled uniformly along straight lines between pairs of points sampled from the data distribution P_r and the generator distribution P_g.

In this chapter, we will use SingleLoad-GAN as the baseline model for benchmarking the performance of MultiLoad-GAN, for the reasons explained in section 2.2.1. The SingleLoad-GAN based approach is introduced in [10]-[12] and in this chapter we reproduced the model in [10].

Let $\mathbf{P}_M^i = \left[p_1^i, p_2^i, \dots, p_M^i\right]^T$ represent the load profile of the i^{th} individual user with *M* data samples. SingleLoad-GAN generates synthetic profiles, $\widehat{\mathbf{P}}_M^i = \left[\hat{p}_1^i, \hat{p}_2^i, \dots, \hat{p}_M^i\right]^T$ with similar distribution as $\mathbf{P}_M^i$. After all profiles are generated for *N* loads, we obtain a load group, $\widehat{\mathbf{P}}_{M\times N}$,

$$\widehat{\mathbf{P}}_{M\times N} = \left[\widehat{\mathbf{P}}_M^1, \widehat{\mathbf{P}}_M^2 \dots, \widehat{\mathbf{P}}_M^N\right] = \begin{bmatrix} \hat{p}_1^1 & \hat{p}_1^2 & \cdots & \hat{p}_1^N \\ \hat{p}_2^1 & \hat{p}_2^2 & & \hat{p}_2^N \\ \vdots & & \ddots & \vdots \\ \hat{p}_M^1 & \hat{p}_M^2 & \cdots & \hat{p}_M^N \end{bmatrix} \tag{2.4}$$

2.2.3 MultiLoad-GAN Model

The configuration of MultiLoad-GAN is shown in Figure 2.2. The MultiLoad-GAN generator network is a deep Convolutional Neural Network (CNN). First, a fully connected layer is used to extract features from the input data to a 2D data matrix. Then, transpose convolutional layers with decreasing number of kernels are used to generate load profile groups. ReLU is used as the activation function. Inspired by [17], we use batch normalization following each transpose convolutional layer to enhance the training process. A $Tanh$ layer is added to the end of the generator to normalize the output values into [-1, 1].

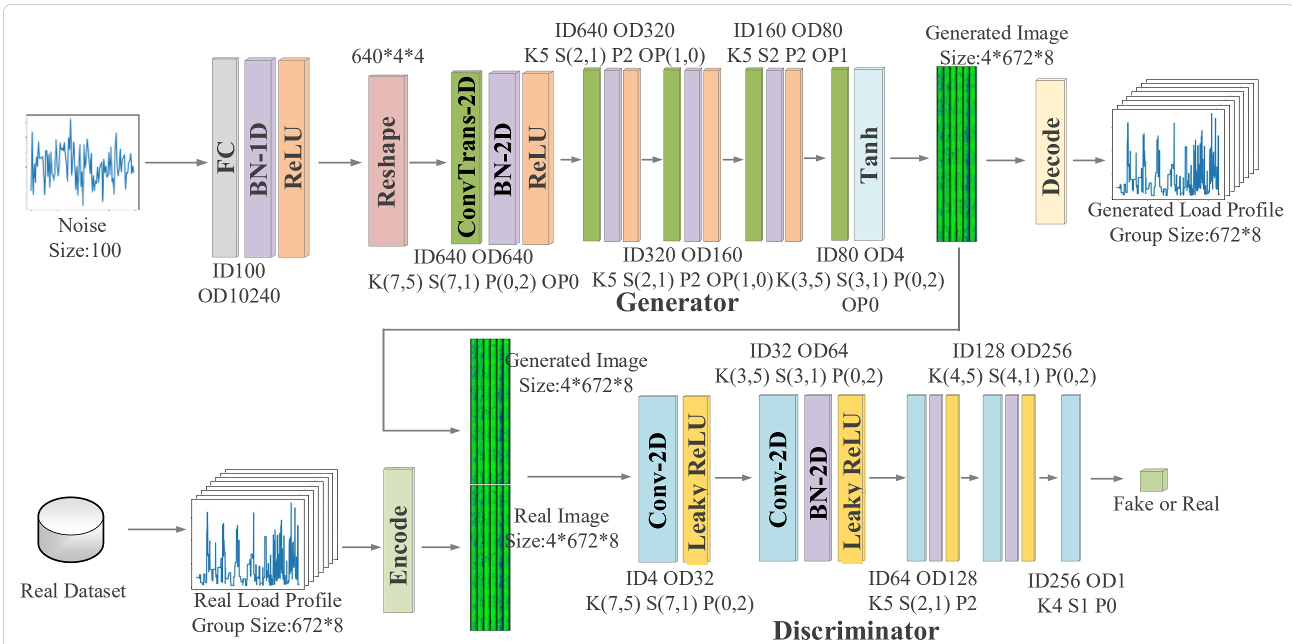


Figure 2.2 MultiLoad-GAN architecture with corresponding input dimension (ID), output dimension (OD), kernel size (K), stride (S), padding (P), output padding (OP) for each convolutional layer. The parameter is an example for generating weekly 15-min load group with 8 households.

The MultiLoad-GAN discriminator is built with a set of convolutional layers with increasing number of kernels. The activation function is Leaky ReLU. Also, a batch normalization layer is added following each convolution layer.

Compared with SingleLoad-GAN, a distinct advancement of MultiLoad-GAN is that it generates N load profiles simultaneously, so we have

$$\ddot{\mathbf{P}}_{M\times N} = \left[\ddot{\mathbf{P}}_M^1, \ddot{\mathbf{P}}_M^2 \dots, \ddot{\mathbf{P}}_M^N\right] = \begin{bmatrix} \ddot{p}_1^1 & \ddot{p}_1^2 & \cdots & \ddot{p}_1^N \\ \ddot{p}_2^1 & \ddot{p}_2^2 & & \ddot{p}_2^N \\ \vdots & & \ddots & \vdots \\ \ddot{p}_M^1 & \ddot{p}_M^2 & \cdots & \ddot{p}_M^N \end{bmatrix} \tag{2.5}$$

As shown in Figure 2.3, inspired by the image processing encoding process, a unique profile-to-image encoding method is developed to encode a group of load profiles (an M×N matrix) into 3 color channels (red-R, green-G and blue-B). Dependency of load on temperature is investigated in many load forecasting researchers such as [18][19]. Then, we add a fourth channel to represent temperature (T) to reflect the weather dependence.

To encode $\mathbf{P}_{M\times N}$, map $p_m^n \in \mathbf{P}_{M\times N}$ to the RBG channels, $[r_m^n, g_m^n, b_m^n]$, by

$$r_m^n = \begin{cases} 0, & 0 \le p_m^n < l_1 \\ \dfrac{p_m^n - l_1}{l_2 - l_1}, & l_1 \le p_m^n < l_2 \\ 1 - \dfrac{p_m^n - l_2}{l_3 - l_2}, & l_2 \le p_m^n < l_3 \\ 0, & l_3 \le p_m^n \end{cases}$$

$$g_m^n = \begin{cases} 1 - \frac{p_m^n}{l_1}, & 0 \le p_m^n < l_1 \\ 0, & l_1 \le p_m^n \end{cases} \tag{2.6}$$

$$b_m^n = \begin{cases} \dfrac{p_m^n}{l_1}, & 0 \le p_m^n < l_1 \\ 1 - \dfrac{p_m^n - l_1}{l_2 - l_1}, & l_1 \le p_m^n < l_2 \\ 0, & l_2 \le p_m^n \end{cases}$$

$$l_1 = \tfrac{1}{3} l_3,\ l_2 = \tfrac{2}{3} l_3,$$

$$l_3 = \max(p_m^n,\ \ for\ \ m \in [0, M], n \in [0, N]).$$

The fourth channel is the temperature channel. Temperature measurement T_m at time *m* is first normalized by 120°F in order to encode it to the brown channel, $[t_m]$, so we have

$$t_m = \frac{T_m}{120} \tag{2.7}$$

Thus, the load p_m^n and temperature T_m measurement at time point *m* is encoded into a normalized vector $[r, g, b, t]$ within [0, 1]. Then we further convert them into [-1, 1] to benefit the model training process by

$$[r, g, b, t] = \frac{[r, g, b, t] - 0.5}{0.5} \tag{2.8}$$

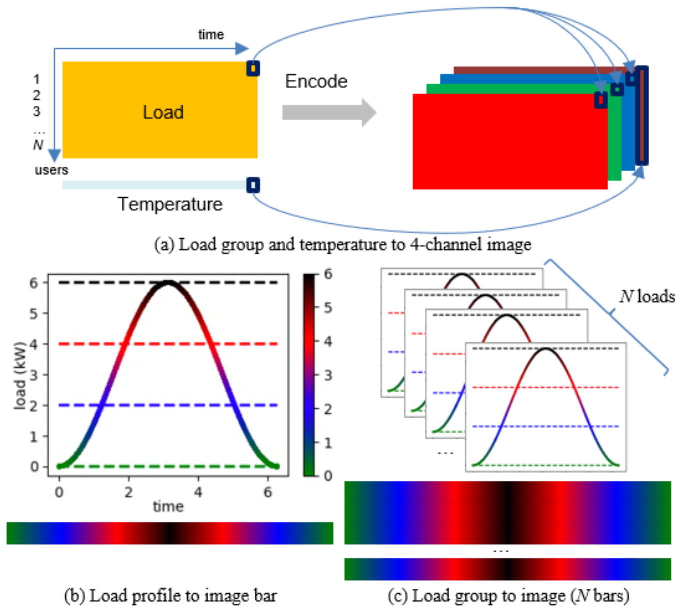


Figure 2.3 An illustration of the profile–to-image encoding process.

By encoding load profiles into an RBG image, machine learning tools developed in 2D image processing domain can be readily used to extract the spatial-temporal correlations among multiple loads. Also, RGB images make it easier for human to visually recognize load variations patterns. Thus, we consider the profile-to-image encoding method and the corresponding adjustments on the conventional GAN architecture as one of the contributions of the chapter.

2.2.4 Realisticness Evaluation

Denote the load group generated by SingleLoad-GAN as $\widehat{\mathbf{P}}_{M\times N} \in \boldsymbol{\Omega}_{LG}^{SLGAN}$; denote the load group generated by MultiLoad-GAN as $\ddot{\mathbf{P}}_{M\times N} \in \boldsymbol{\Omega}_{LG}^{MLGAN}$; denote the ground-truth load group as $\mathbf{P}_{M\times N} \in \boldsymbol{\Omega}_{LG}$. As shown in Figure 2.1 (c), first, we compare statistical evaluation indices to quantify the realisticness at the household level (i.e., for each individual user) and the aggregation level (i.e., at the transformer level). Ideally, $\ddot{\mathbf{P}}_{M\times N}$ is expected to have similar transformer-level statistics with $\mathbf{P}_{M\times N}$ than those of $\widehat{\mathbf{P}}_{M\times N}$because spatial-temporal correlations among a group of users are implicitly learned by MultiLoad-GAN. Next, to compare features unable to be captured by human-defined indices, a specialized DLC is trained to assess the realisticness of $\ddot{\mathbf{P}}_{M\times N}$ and $\widehat{\mathbf{P}}_{M\times N}$ by comparing high-level features captured in $\mathbf{P}_{M\times N}$.

1) Method 1: Statistical Evaluation

The statistical evaluation metrics are summarized in Table 2-2.

Table 2-2 Statistical evaluation indexes for the n^{th} load Group

No.	Indexes
1	Peak load distribution
2	Mean power consumption distribution
3	Load ramps distribution
4	Hourly energy consumption distribution
5	Daily energy consumption distribution

First, distributions of each load character index for the generated and ground-truth load groups, $\ddot{\mathbf{P}}_{M\times N}$, $\widehat{\mathbf{P}}_{M\times N}$, and $\mathbf{P}_{M\times N}$, are first calculated at both the household- and transformer- levels. Then, similarities between the distributions of each index for $\mathbf{\Omega}_{LG}^{MLGAN}$, $\mathbf{\Omega}_{LG}^{SLGAN}$ and $\mathbf{\Omega}_{LG}$ are compared to quantify the realisticness of the generated load groups.

2) Method 2: Deep-Learning Classification

DLC is trained in parallel with MultiLoad-GAN. DLC can identify real and fake load groups with higher accuracy than the MultiLoad-GAN *discriminator* because both positive and negative samples are used to train DLC, making it a highly "specialized" classifier for identify real and fake load groups.

As shown in Figure 2.4, the configuration of DLC includes a deep convolutional network consisting of 5 2-D convolutional layers with increasing number of kernels and 5 fully connected layers with decreasing number of features. The DLC input is a $M \times N$ load group and the output is the probability of realisticness, which reflects how well realistic group-wise spatial-temporal correlations can be captured.

Assume there are Q samples (each sample $\mathbf{P}_{M\times N} \in \mathbf{\Omega}_{LG}$ is a group of load profile with size $M \times N$) used to train the classifier. For the i^{th} sample, the classifier output is

$$C(\mathbf{P}_{M\times N}) = P_{true}(i) \tag{2.9}$$

where $P_{true}(i) \in [0,1]$ is the probability for the i^{th} load group to be "real". Thus, we consider the sample to be "positive" if $P_{true}(i) > 0.5$ and "negative" otherwise.

Let Q_{real} be the number of samples classified as "positive". The Percentage of Real (POR) of the dataset is calculated as

$$POR = \frac{Q_{real}}{Q} \times 100\% \tag{2.10}$$

Although POR can be used to evaluate the accuracy of the classifier, it cannot reflect the confidence level of the classification results. For example, considering a sample "positive" when $P_{true}(i) = 0.51$ is a less certain judgement than when $P_{true}(i) = 1$. So, we further calculate the Mean Confidence Level of the dataset (MCL) as

$$MCL = \frac{1}{Q}\sum_{i=1}^{Q} P_{true}(i) \tag{2.11}$$

The similarity of real dataset $\mathbf{\Omega}_{LG}$ and MultiLoad-GAN generated dataset $\mathbf{\Omega}_{LG}^{MLGAN}$ can be calculated by the Fréchet inception distance [20][21] between the two distributions

$$Similarity = FID\left(C(\mathbf{\Omega}_{LG}), C(\mathbf{\Omega}_{LG}^{MLGAN})\right) \tag{2.12}$$

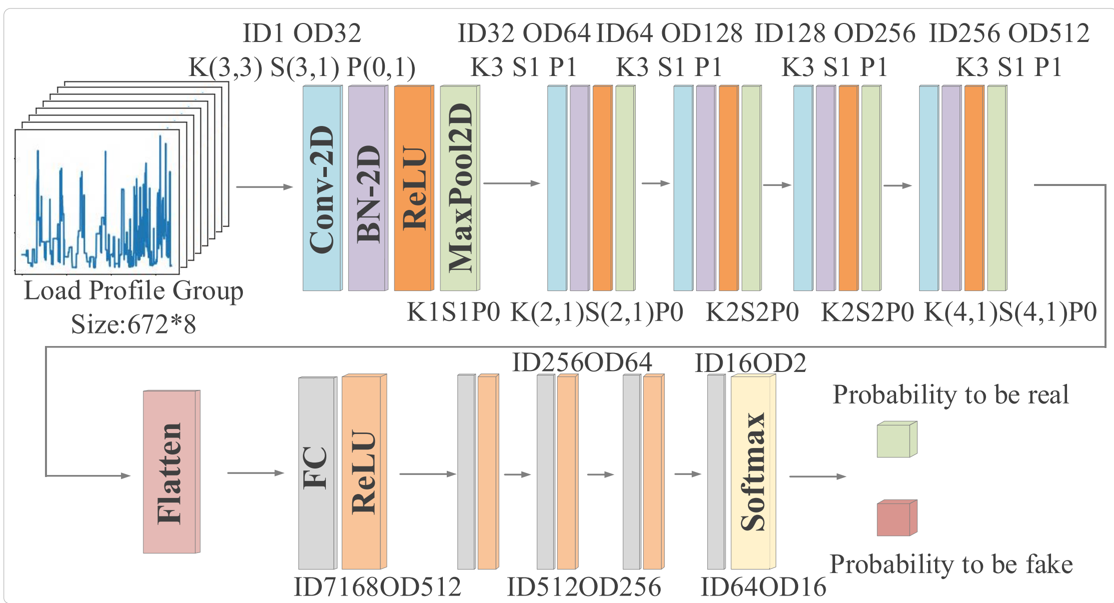


Figure 2.4 Classifier architecture with corresponding input dimension (ID), output dimension (OD), kernel size (K), stride (S), and padding (P) for each convolution layer, max pool layer, and fully connected layer. The parameter is an example for generating weekly 15-min load group with 8 households.

2.2.5 Negative Sample Selection

To train DLC, both positive samples (i.e., realistic load groups labeled by 1) and negative samples (i.e., unrealistic load groups labeled by 0) are required. The "original positive samples", $\mathbf{P}_{M\times N}$, is the labelled ground-truth data set. However, because negative samples (i.e., load groups served by different transformers) do not "naturally" exist, they are generated by pulling load profiles from various transformer load groups.

These negative samples play a vital role in preventing bias in load group classification. Without them, the classifier may mistakenly classify all load groups as positive samples, resulting in subpar performance. Therefore, generating high-quality negative samples is crucial for achieving a balanced training dataset, reducing biases, and enabling the classifier to learn how to classify unseen data effectively. This process also enhances the classifier's robustness in the presence of noise or ambiguous cases.

However, negative sample generation is a nontrivial task. Randomly selecting a group of users from a smart meter database that contains load profiles collected in the same area in the same season is a straightforward way to generate a negative sample. However, it is an uncontrolled approach with several drawbacks. First, a significant amount of the generated negative samples are too easy to be classified as "negative". Thus, DLC cannot learn complex hidden features. Second, an unknown number of the generated negative samples are actually positive samples. This is because, often times, a load served by one service transformer is equally likely to be served by an adjacent transformer with the same size. This phenomenon is quite common when supplying small residential loads. Thus, randomly drawing loads from a regional smart meter database to obtain negative samples is not a reliable negative sample generation strategy.

Therefore, a statistic-based negative sample generation method is developed. First, we obtain operational statistics from the "original positive samples". As shown in Figure 2.5 (a), we evenly divide the mean power value distribution of the real load profiles into 6 parts. A negative sample can thus be obtained if we randomly select K load profiles from the red box region and $N-K$ load profiles from the black box regions, where K is a random integer in $[0, N/2]$ and randomized in each selection process. By doing so, the obtained load group has a much higher chance to be a negative sample. Similarly, based on Figure 2.5 (b), we can select negative samples to let the load group having different weekly peak distributions from that of the "original positive samples". By controlling the distance between the "real" and "fake" distributions, one can generate "very negative", "negative", "slightly negative", and "almost positive" samples. This gives the modeler the flexibility to tune the DLC to capture different level of realisticness.

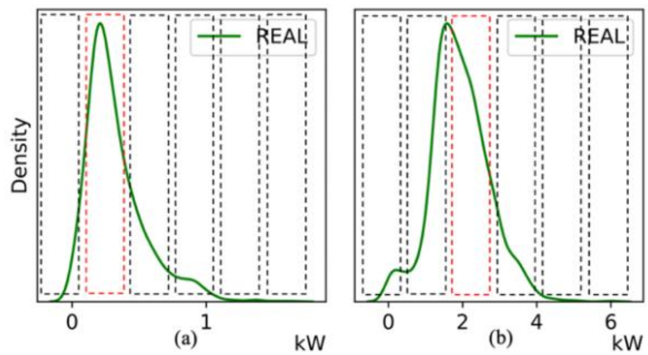


Figure 2.5 (a) Mean power distribution and (b) Peak load distribution.

2.2.6 Automatic Data Augmentation

As shown in Figure 2.6, we propose an iterative ADA mechanism that leverages the generation ability of MultiLoad-GAN and the classification capability of DLC to iteratively create augmented training samples in the training (e.g. at step $t-1$) in order to boost subsequent training (e.g., at step t) of both MultiLoad-GAN and DLC. ADA includes three processes: unlabeled data set creation, labelling data for classifier training, and augmented data for MultiLoad-GAN training.

First, we use two methods to create unlabeled load groups, $\boldsymbol{\Omega}_{LG}^{unlabeled}$: 1) using MultiLoad-GAN to generate load groups directly to obtain $\boldsymbol{\Omega}_{LG}^{MLGAN}$, and 2) randomly sampling N load profiles from a smart meter database, $\boldsymbol{\Omega}_{Load}$, to obtain $\boldsymbol{\Omega}_{LG}^{Rand}$. Note that at this stage, $\boldsymbol{\Omega}_{LG}^{unlabeled}$ contains both positive and negative samples.

Next, $\boldsymbol{\Omega}_{LG}^{unlabeled}$ will be labeled by the DLC with parameter, θ_C obtained from the previous training step, $t-1$. Note that such labels can include errors, depending on what the accuracy of the DLC is at the training stage. Then, the labeled data, together with the “original positive samples” ($\boldsymbol{\Omega}_{LG}$) and negative samples ($\boldsymbol{\Omega}_{LG}^{Neg}$, selected by NSG in Section 2.2.5) will be used to train the DLC.

Third, once the Classifier is trained, it will immediately be used for identifying positive samples from $\boldsymbol{\Omega}_{LG}^{Rand}$, which will then be used as the augmented dataset $\boldsymbol{\Omega}_{LG}^{Aug}$ for training MultiLoad-GAN. Note that only samples with a high confidence level (e.g., samples with scores > 0.9) will be selected to enhance the quality of the augmented data.

When the interactive training process progresses, the DLC training will improve significantly because it receives an increasing number of high-quality training data generated by MultiLoad-GAN and random sample selection. In return, the classifier can help identify positive samples with higher confidence level to enhance the training of MultiLoad-GAN. The training of MultiLoad-GAN and DLC will be both enhanced until the accuracy of the classifier saturates to a certain level.

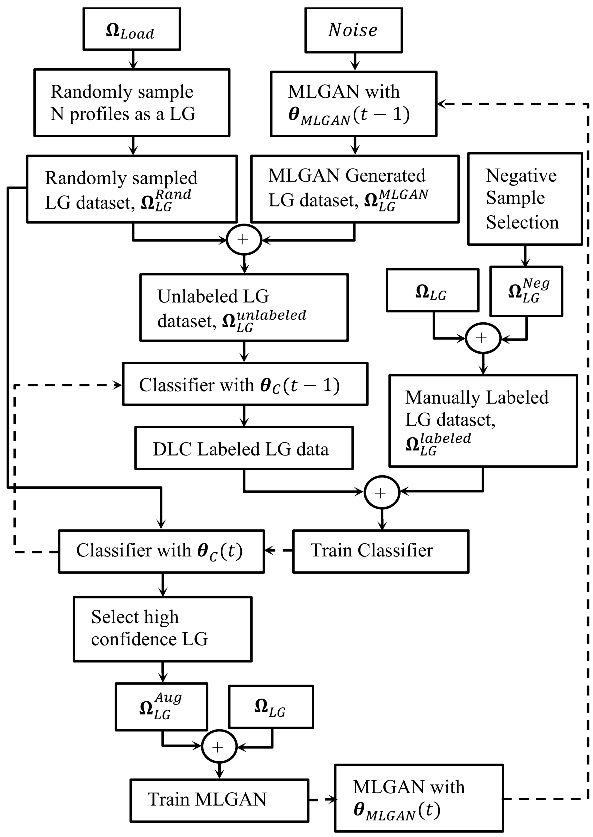


Figure 2.6 Flowchart for the iterative ADA process.

2.3 Simulation Results

In this chapter, we use transformer-level load group generation as an example to illustrate the group-load generation process and evaluate algorithm performance.

A utility in the North Carolina area provides us with a labelled data set, which includes 3-year (from 2017 to 2020), 15-minute load profiles collected from 64 residential customers, which are served by eight 50kVA transformers (8 customers per transformer). The corresponding temperature data are from National Oceanic and Atmospheric Administration (NOAA) [22] website. After excluding the missing data, we obtain 1424 load group (each sample is a matrix of $\mathbf{P}_{672\times 8}$), which are considered as "original positive samples". Thus, the output of the MultiLoad-GAN model should be weekly load profiles (i.e., $M = 4 \times 24 \times 7 = 672$) for a group of 8 loads (i.e., N=8) serving by the same transformer.

The power and temperature encoding parameters are given in Table 2-3 and the profile-to-image process is illustrated in Figure 2.3 (b)(c). Hyper-parameter settings of MultiLoad-GAN are given in Table 2-4. We use the root mean square propagation (RMSProp) optimizer. The model is built in the PyTorch environment and trained on a single NVIDIA GeForce RTX 1080 GPU. Each training takes approximately 2 hours. The architecture of the benchmark model, SingleLoad-GAN is implemented with a set of hyper-parameter settings shown in Table 2-4 Each training takes approximately 1 hour.

It is important to point out that we do not split the data into training and testing sets for a GAN model, as it is done in other supervised learning. Because the GAN model learns the mapping from the latent vector distribution to the actual data distribution. As a result, the evaluation of the GAN model is not a point-to-point comparison between the generated results and the actual results (because the generated results should be different from any existing ones and therefore is not comparable). Instead, the evaluation focuses on the realisticness of the generated results, such as statistical evaluation, visual inspection, deep learning classification, etc.

Table 2-3 Parameters used in the Profile-to-image encoding process

Load (kW)	**Vector [r, g, b]**	**Temperature (Fahrenheit)**	**Vector [t]**
0	[0, 1, 0]	0	[0]
(0, 2)	g↓, b↑	(0, 120)	t↑
2 (l_3)	[0, 0, 1]		
(2, 4)	b↓, r↑		
4 (l_3)	[1, 0, 0]		
(4, 6)	r↓		
[6 (l_3), +∞)	[0, 0, 0]	120	[1]

The loss curves when training MultiLoad-GAN is shown in Figure 2.7 (a) stage 1. Initially, there is a sharp decrease of the discriminator loss. This means that the discriminator quickly captures the differences between the real ($\mathbf{P}_{672\times 8}$) and fake ($\ddot{\mathbf{P}}_{672\times 8}$) load groups generated by the naive generator. When the generator network is stronger than the discriminator network, and able to generate more realistic samples that can fool the discriminator, the loss of the discriminator will increase, and the loss of the generator will decrease. Otherwise, when the discriminator is stronger, the loss of the discriminator will decrease, and the loss of the generator will increase. Such adversarial training process allows both the generator and the discriminator to continuously improve themselves. After about 300 epochs, the generator and discriminator of MultiLoad-GAN reach a balanced state, showing that the generator can generate realistic load groups. The training process of SingleLoad-GAN shown in Figure 2.7 (b) is similar.

The generated load groups are shown in Figure 2.8. It is hard to evaluate the realisticness of a load profile by visually comparing the load profiles, even harder for a human to judge whether a group of load profiles bear similar spatial-temporal correlations. The results show that it is necessary to use statistical metrics and DLC for quantifying realisticness in synthetic load profile generation instead of relying on visual inspection, which is commonly used in image processing domain.

Table 2-4 Hyperparameter Setup for the GAN model

Parameter	**MultiLoad-GAN**	**SingleLoad-GAN**
Learning rate	1e-4(D) 1.4e-4(G)	1e-4(D) 1.2e-4(G)
Gradient penalty weight - λ	10	10
Slop of LeakyReLU	0.2	0.2
Batch size	16	64
Training epochs	300	100
Training time	Stage 1: ~2hrs Stage 2: ~4hrs	~1hr

2.3.1 Statistical Evaluation

To compare the performance improvement, we compared 1424 load groups ($\ddot{\mathbf{P}}_{672\times8}$) generated by MultiLoad-GAN with 1424 load groups ($\widehat{\mathbf{P}}_{672\times8}$) generated by SingleLoad-GAN. By doing so, we have a real load group database ($\boldsymbol{\Omega}_{LG}$), a MultiLoad-GAN generated load group database ($\boldsymbol{\Omega}_{LG}^{MLGAN}$), and a SingleLoad-GAN generated database ($\boldsymbol{\Omega}_{LG}^{SLGAN}$), each having 1424 samples. The load statistics can be calculated at both the household and transformer levels. By comparing the distance between the metric distribution of the generated load groups and the real load groups, we can assess the realisticness of the generated load profiles.

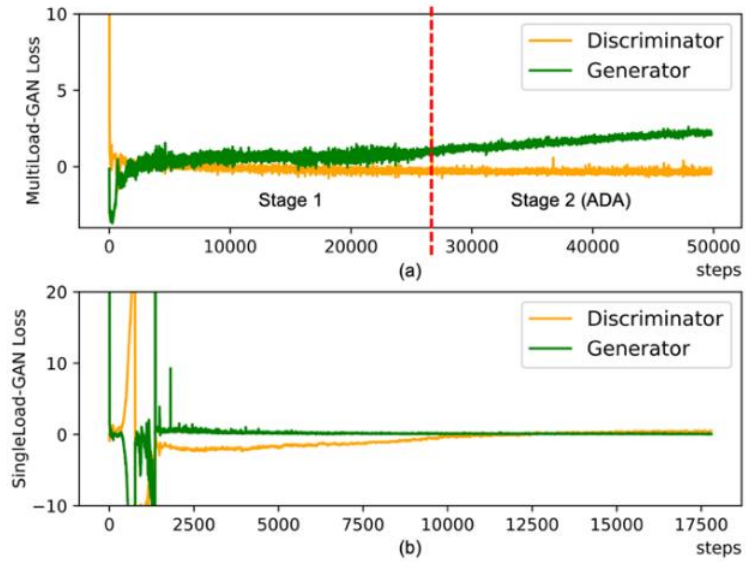
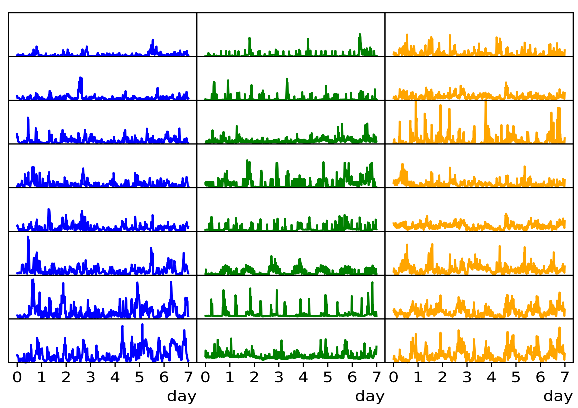


Figure 2.7 (a) Loss curves of MultiLoad-GAN, stage 1: Loss curve of discriminator and generator in the MultiLoad-GAN training stage; stage 2: Loss curve of discriminator and generator in the Automatic Data Augmentation (ADA) training stage. (b) Loss curves of SingleLoad-GAN



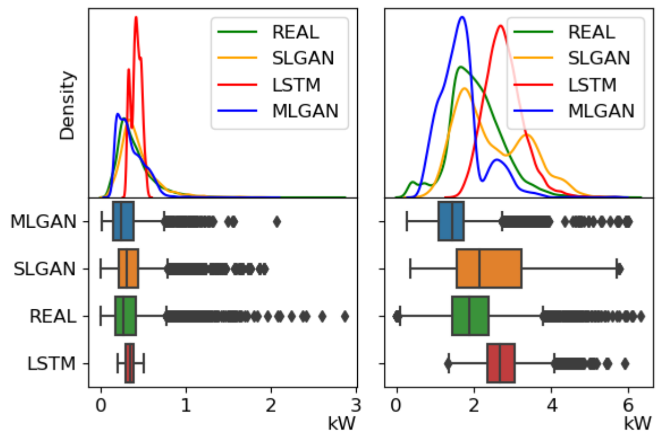
(a) (b) (c)

Figure 2.8 (a) a MultiLoad-GAN generated load group ($\ddot{\mathbf{P}}_{672\times8}$), (b) a real load group ($\mathbf{P}_{672\times8}$), (c) a SingleLoad-GAN generated load group ($\widehat{\mathbf{P}}_{672\times8}$).

To ensure a comprehensive evaluation, an LSTM-based load forecasting method has been used as an additional benchmark. This method forecasts a set of load profiles using real load groups. The data processing procedure follows a similar approach to SingleLoad-GAN.

1) Evaluation at the Household-level

For the household level evaluation, statistics are calculated based on individual load profile. Note that each database contains 1424×8=11392 weekly load profiles.

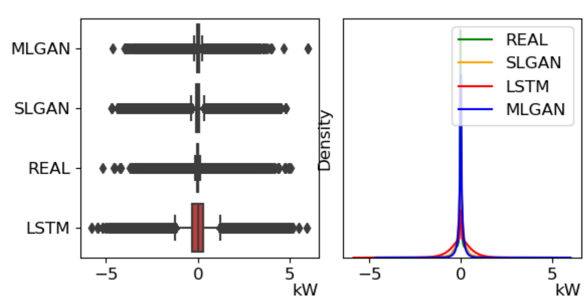


(a) Mean (b) Peak

Figure 2.9 (a) Household-level mean power distribution curves and boxplots, and (b) Household-level peak load value distributions and boxplots.

Mean and peak. As shown in Figure 2.9, MultiLoad-GAN and SingleLoad-GAN can all generate load profiles with the mean value distribution close to that of real load groups. However, SingleLoad-GAN tends to generate load profiles with higher peak values (e.g., from 3 to 5kW), making its peak value distribution deviate from the ground truth. This is because when generating load profiles one at a time, the correlation between users are not considered, making SingleLoad-GAN results less realistic. Due to its nature as a forecasting method relying on historical data, LSTM tends to generate load profiles with narrow ranges for mean and peak values. This limitation makes it challenging to produce diversified load profiles.

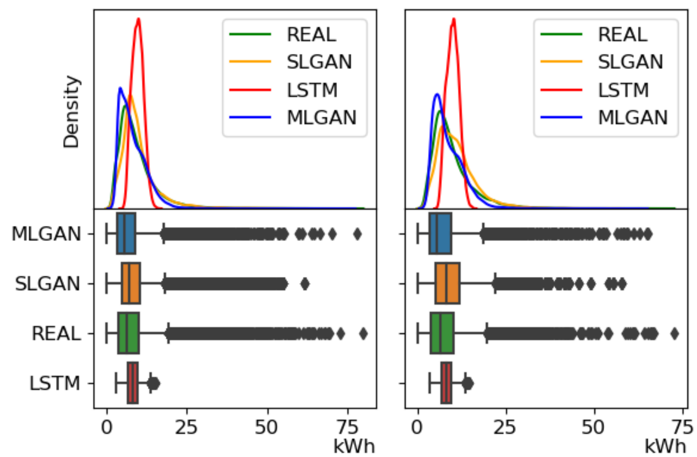
Load ramps. The distributions of load ramps on the four data sets are shown in Figure 2.10. We can see that MultiLoad-GAN and SingleLoad-GAN show comparable performance on this metric, but LSTM performs worse than them.



(a) Boxplot (b) Probability density function

Figure 2.10 Household-level load ramp distributions curves and boxplots.

Daily and hourly power consumption. As shown in Figures 2.11 and 2.12, MultiLoad-GAN and SingleLoad-GAN has similar performance on daily power consumption but is slightly worse than SingleLoad-GAN on hourly power consumption. LSTM still perform worse than them.



(a) Weekday (b) Weekend

Figure 2.11 Household-level daily power consumption distributions and boxplots in weekday and weekend.

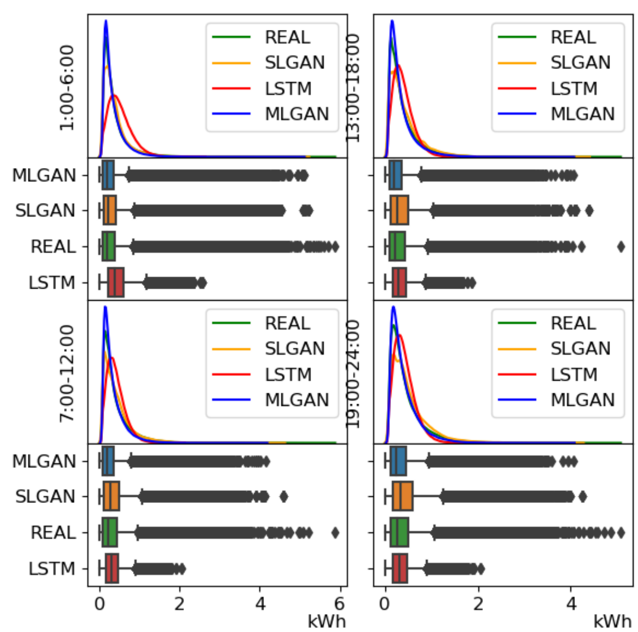
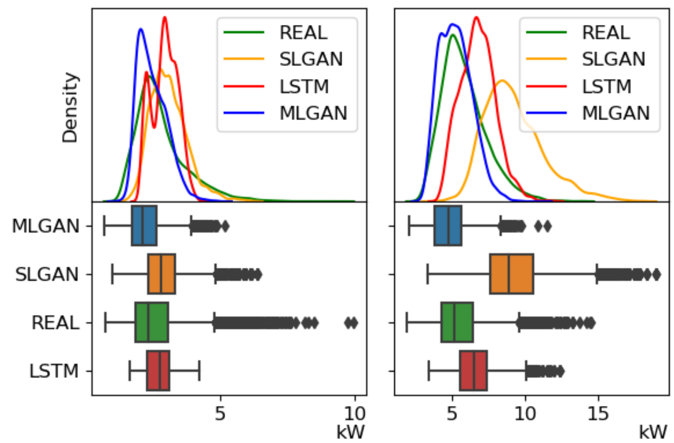


Figure 2.12 Household-level hourly power consumption distributions and boxplots in four time periods of a day.

2) Evaluation at the Transformer-level

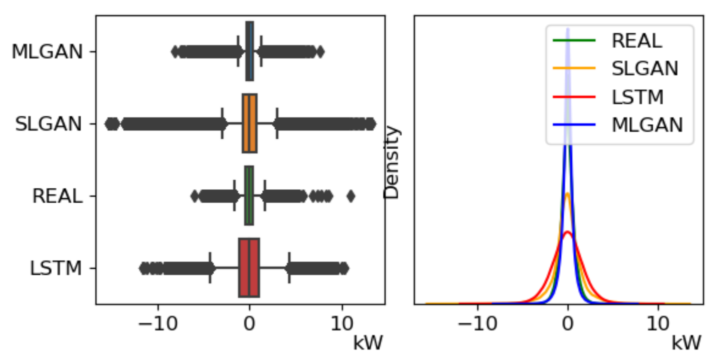
Next, we compare the load group characteristic for the 1424 aggregated profiles in each of the four databases.



(a) Mean (b) Peak

Figure 2.13 (a) Transformer-level mean power distributions and boxplots, and (b) Transformer-level peak load value distributions and boxplots.

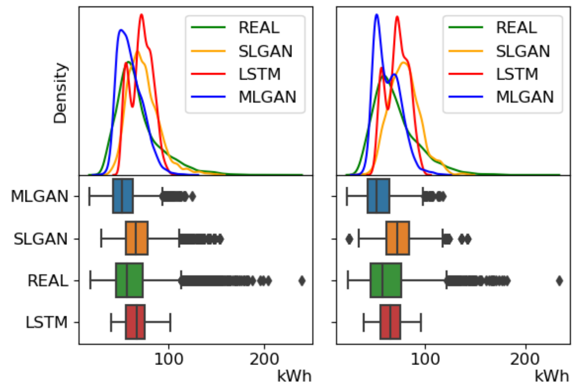
Mean and peak. As shown in Figure 2.13, all the models have similar performance on the mean value distribution. But SingleLoad-GAN and LSTM tend to generate load groups with higher peak values.



(a) Boxplot (b) Probability density function

Figure 2.14 Transformer-level load ramp distributions and boxplots.

Load ramps. As shown in Figure 2.14, MultiLoad-GAN results are smoother than the actual data (i.e., the distribution is more centered towards 0), while SingleLoad-GAN results have more fluctuation. However, LSTM generates more large ramping. Overall, MultiLoad-GAN distribution is closer to the ground truth.



(a) Weekday (b) Weekend

Figure 2.15 Transformer-level daily power consumption distributions and boxplots in weekday and weekend.

Daily and hourly power consumption. As shown in Figures 2.15 and 2.16, MultiLoad-GAN has comparable performance with the other two methods on daily power consumption and is better than them on hourly power consumption.

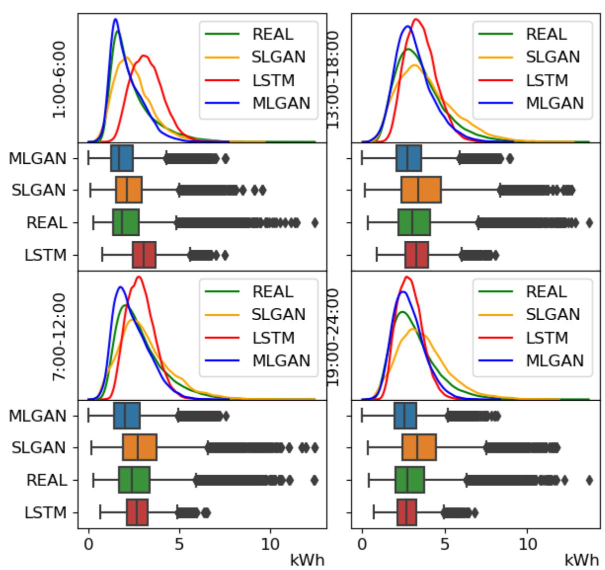


Figure 2.16 Transformer-level hourly power consumption distributions and boxplots in four time periods of a day.

To make quantitively comparison, we calculate the FID between the distributions of the generated dataset and real dataset and summarize the results in Table 2-5. We can see that

MultiLoad-GAN has comparable performance with SingleLoad-GAN and LSTM on the household-level statistics (3 indices out of 5 perform the best), but show significant advantages on the aggregation-level ones (outperform SingleLoad-GAN and LSTM in all the indices). This means the MultiLoad-GAN can successfully capture correlations between users served by the same transformer. Thus, it can generate load groups with more realistic aggregation-level features while preserving the characteristics of each individual load.

Table 2-5 Statistical Evaluation Results

Evaluation Criteria		MultiLoad-GAN	SingleLoad-GAN	LSTM
Household Level	Mean	**5.319e-4**	6.473e-4	3.854e-3
	Peak	**2.431e-2**	4.884e-2	9.6.6e-2
	Ramp	6.700e-4	**3.445e-4**	1.650e-1
	Hourly Consumption	(4.125e-4 1.115e-3 4.556e-4 6.726e-4) **6.639e-4**	(1.464e-4 5.098e-4 5.225e-4 1.482e-3) 6.652e-4	(1.890e-3 2.644e-3 2.200e-3 3.362e-3) 2.524e-3
	Daily Consumption	(3.245e-1 3.610e-1) 3.428e-1	(2.516e-1 3.411e-1) **2.964e-1**	(2.457 2.561) 4.509
Aggregation Level	Mean	**3.454e-2**	3.491e-2	4.168e-2
	Peak	**5.822e-2**	2.252	1.748e-1
	Ramp	**6.893e-4**	1.074e-1	1.331e-1
	Hourly Consumption	(1.611e-2 3.608e-2 1.728e-2 2.738e-2) **2.421e-2**	(5.634e-3 2.821e-2 3.507e-2 7.370e-2) 3.565e-2	(9.740e-2 4.661e-2 3.045e-2 6.072e-2) 5.879e-2
	Daily Consumption	(2.030e1 1.943e1) **1.987e1**	(1.649e1 3.230e1) 2.440e1	(2.490e1 2.195e1) 2.342e1

2.3.2 Realisticness Evaluation based on DLC Classification

To train DLC, 4272 generated negative samples and the 1424 “original positive samples” are used as the training set. The positive-negative sample ratio is 1:3. The data set are split into training (80%) and testing (20%) sets.

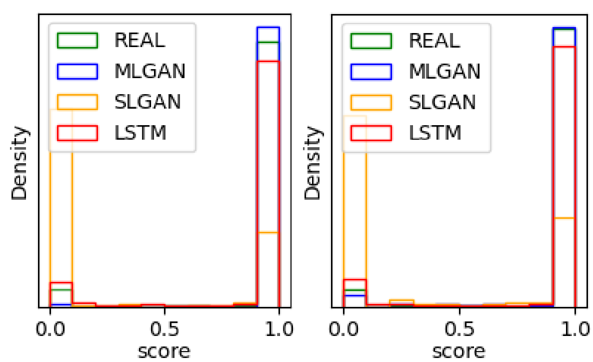
Three negative sample generation methods are compared: 1) randomly select 8 weekly load profiles from the regional smart meter database; 2) select negative samples based on mean value distribution (see Figure 2.5 (a)); 3) select negative samples using mean and peak distributions (see Figure 2.5, the proposed method).

As shown in Table 2-6, randomly selecting load profiles as negative samples results in poor identification accuracy, while using the proposed method for NSG, the accuracy can improve to approximately 94%, which is a 20% improvement.

Table 2-6 Classifier Accuracy with Different NSG Criteria

Method No.	**Negative sample generation criteria**		**Test Accuracy (%)**	**Mean Confidence Level**
	Mean	Peak		
DLC			70.37	
	√		92.42	
	√	√	**94.38**	**0.9371**
RF	√	√	93.77	0.8663
K-NN	√	√	90.61	0.9254
DT	√	√	89.56	1.0

To demonstrate the importance of using DLC, several non-DL classifiers, such as Random Forest, K-Nearest Neighbors, Decision Tree, are used for comparison. Negative sample generated based on mean and peak values are used as they have been shown to enhance classifier performance in previous analyses. The result shows that DLC exhibits the highest accuracy and a relative high confidence level. Therefore, we choose DLC as the evaluation model.



(a) Without ADA (b) With ADA

Figure 2.17 Distribution of DLC scores (a) without ADA and (b) with ADA.

The trained DLC will be used to evaluate the realisticness of the load groups generated by MultiLoad-GAN and SingleLoad-GAN. For all four data sets (i.e., real-world load group samples $\mathbf{P}_{672\times8}$, MultiLoad-GAN generated samples $\ddot{\mathbf{P}}_{672\times8}$, SingleLoad-GAN generated samples $\widehat{\mathbf{P}}_{672\times8}$, and LSTM forecasted samples). The DLC will give a score (i.e., the confidence level) for each sample to show realisticness. The probability density distribution of the scores is shown in Figure 2.17 (a). Key statistics are summarized in the first column of Table 2-7.

Table 2-7 Results of DLC-Based Evaluation

Indices	Real data	SingleLoad-GAN	LSTM	MultiLoad-GAN	MultiLoad-GAN (with ADA)
Percent of Real	94.38%	19.69%	84.83%	99.06%	**94.99%**
Mean Confidence Level	0.9371	0.1913	0.8919	0.9899	**0.9491**
Fréchet inception distance with the $\boldsymbol{Real}$ load group	N/A	0.5173	0.00706	0.01106	**0.000055**

From the results, we have the following observations:

- As shown in Figure 2.17, DLC is confidence about the classification results, because most scores are close to 1 (real) or 0 (fake).
- As shown in Table 2-7, 99.06% of the MultiLoad-GAN generated samples are classified as real by DLC, 84.83% of the LSTM forecasted samples are classified as real, while only 19.69% of the SingleLoad-GAN generated samples are classified as real. This means that MultiLoad-GAN generates load groups with similar high-level features with those of the actual load groups.
- The FID index defined in (2.12) is calculated to measure the similarities between two distributions in Figure 2.17 (a). The FID between "MultiLoad-GAN" and "Real" is 0.01106, FID between "LSTM" and "Real" is 0.007.58, while between SingleLoad-GAN and "Real" is 0.5173.

This result shows that the MultiLoad-GAN generated load groups are much closer to the ground truth ones from the classifier's viewpoint.

2.3.3 Automatic Data Augmentation

ADA training starts from the MultiLoad-GAN and DLC trained in the previous sections. The loss curves of MultiLoad-GAN in ADA process are shown in Figure 2.7 (a) stage 2. The performance indices for the with/without ADA-boosted MultiLoad-GAN cases are summarized

Table 2-7 (indices are defined in section 2.2.4) and Figure 2.17 (b). The results show that the ADA process significantly shorten the distance between MultiLoad-GAN generated data set and the real data set. This shows that the ADA process avoids MultiLoad-GAN to be over-trained so that it only generates load groups strongly resemble the "original positive samples".

Without incorporating the ADA process, the MultiLoad-GAN model generates high-confidence load groups according to the DLC's perspective, resulting in a near 99% real percentage. However, this does not align well with the actual data, where low-confidence load groups exist, leading to a *POR* of 94.38%. By implementing the ADA process, the MultiLoad-GAN model generates additional low-confidence samples, improving its performance to achieve a *POR* of 94.99%, which is closer to the real dataset. Figure 17 (b) visually demonstrates the resulting dataset's distribution, which is closer to that of the original dataset. By considering the *POR* before and after the ADA process, it can be concluded that ADA has enhanced the performance of MultiLoad-GAN by approximately 4.07% (moving closer to the real dataset).

2.4 Conclusion

In this chapter, we present MultiLoad-GAN framework for generating a group of load profiles simultaneously while preserving the spatial-temporal correlations between load profiles in the group. Inspired by the successful application of the GAN-based model in both image processing and power system domain, we develop a novel profile-to-image coding method to convert time-series plots to image patches, so that GAN-based models can be readily used for processing groups of load profiles. To solve the data scarcity problem, we developed an iterative data augmentation process to train MultiLoad-GAN and a classifier alternatively. Thus, the classifier can be used to automatically label positive and negative samples for augmenting the training of both the classifier and the MultiLoad-GAN in subsequent steps. Our simulation results, based on statistical and DLC evaluation, show that compared with the state-of-the-art synthetic load generation process, MultiLoad-GAN better preserves both household-level and group-level load characteristics.

3. BERT-BASED MISSING LOAD DATA RESTORATION

3.1 Background

The restoration of missing data holds significant importance in power system analysis. In power system load profiles, two types of missing data exist. First, temporary equipment malfunctions or communication losses result in missing data, adversely affecting data quality. This impedes various data-related tasks such as load forecasting, load disaggregation, and anomaly detection. Second, demand response (DR) or conservation voltage reduction (CVR) baseline presents a unique case of missing data. For instance, utilities widely employ CVR for peak load reduction, where system voltage at the substation bus is decreased by 2-4% during a CVR event to achieve load reduction. However, the original load profile during a CVR event (the baseline), assuming no voltage reduction, remains unknown. Accurately estimating this baseline is crucial for load service providers to quantify the load reduction caused by CVR.

Current techniques for recovering missing data fall into two categories: model-based and data-driven methods. Table 3-1 provides a comprehensive overview and comparison of existing missing data restoration methods in power systems. Notably, Load-PIN [50], a GAN-based approach, outperforms model-based, similarity-based, regression-based, and other GAN-based methods in restoring missing data segments (MDSs) of fixed length.

However, existing inpainting techniques typically offer only a single solution. As illustrated in Figure 3.1, we observed that MDSs can be analogized to missing words in sentences, thus equating the restoration process to recovering missing words in sentences or paragraphs. In the word completion task depicted in Figure 3.1, if someone is fluent in French, "France" might emerge as the most probable choice. However, "Quebec" and "Cameroon" also represent viable alternatives, as they are regions where French is an official language. Similarly, when restoring an MDS, it's essential to provide multiple patching options, each with comparable likelihoods of being the best match.

Inspired by this observation, we use advanced Natural Language Processing (NLP) techniques to solve missing data restoration problem in power system domain. One of the advantages of employing NLP models is their capability to generate multiple alternatives for a missing word, each accompanied by a confidence level.

Table 3-1 Comparison of Existing Power System Load Profile Inpainting Methods

		Description	Advantages	Disadvantages
Model-based methods [23]-[27]		Use physical system models to simulate responses to external disturbances for restoring missing data segments.	Explainable as the models reflect the laws of physics.	Require accurate distribution system model.
Data-driven methods	Similarity-based [28]-[31]	Group load profiles by day type, weather conditions, and shape characteristics of load profiles. The missing data segments are restored by referencing to the data on the load profiles having the best similarity match.	Easy to implement and explainable.	Accuracy of the method dependent on selections of similarity metrics and weights.
	Regression-based (the benchmark method)	Use models including linear regression [32], Long Short Term Memory (LSTM) [33][34], Autoencoders [35][36], Gaussian Regression [37], Support Vector Regression (SVR) [38][39], etc. Or combine multiple regression models [40]-[43].	Provide transparent insights into the relationships between input and output. More efficient for small to medium-sized datasets.	Limited complexity and Limited contextual understanding. Require manual feature engineering.
	GAN-based [44]-[49]	Use Generative Adversarial Nets solve the missing data restoration problems in power system.	Discover underlying patterns in the data without explicit supervision.	Training instability, mode collapse, and hard to evaluate.
	Load-PIN [50] (the benchmark GAN-based method)	Combine Generative Adversarial Nets with Convolutional layers and multi-head self-attention blocks to improve accuracy.	More accurate than model-, similarity-, regression-, and other GAN based models.	Computationally expensive and require large amount of data. **Produce only 1 restoration candidate.**
	BERT-PIN (the proposed method)	Bidirectional Encoder can capture long-range dependencies though self-attention mechanisms.	The **most accurate** method and can produce **multiple restoration** candidates.	Computationally expensive and require large amount of data.

Since 2017, the Transformer model [51] and its variants, such as Bidirectional Encoder Representations from Transformers (BERT) [52], and Vision Transformer (ViT) [53], achieved remarkable success in NLP and computer vision (CV). They excel at handling sequential data and capturing extensive long-range dependencies by employing self-attention mechanisms that allow simultaneous consideration of all positions in a sequence. This introduction of self-attention empowers models to discern the importance of individual elements within input data, facilitating the understanding of intricate dependencies, relationships, and contextual information, ultimately leading to enhanced performance. It's important to clarify that the "Transformer" referred to in this context is an advanced machine learning model and not the transformer device used in power systems, to avoid any potential misunderstanding.

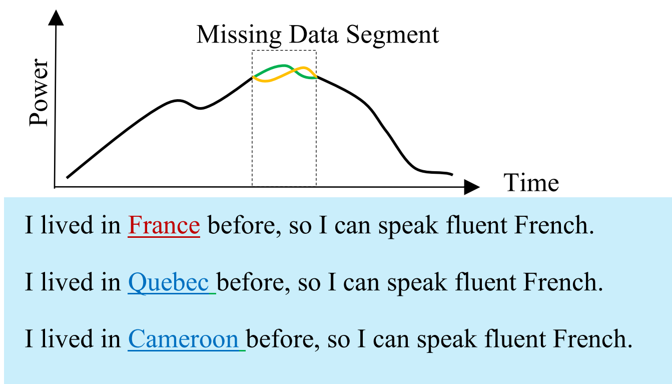


Figure 3.1 An illustration of the load profile inpainting problem.

Therefore, we introduce the BERT-based Profile Inpainting Network (BERT-PIN), a flexible framework designed specifically for restoring multiple missing data segments within power system load profiles. To adapt the standard Transformer model structure for profile inpainting, we divide the load and temperature profiles into line segments, treating each segment as a word, the daily profile as a sentence, and the weekly/monthly load profile as a paragraph comprising multiple sentences. Furthermore, we integrate a top candidate selection process into BERT-PIN, allowing it to generate a series of probability distributions. Based on these distributions, users can produce multiple plausible imputed datasets, each reflecting varying confidence levels.

Although many machine-learning methods exist for restoring missing data, we selected the BERT-based model for three primary reasons. *First*, BERT's bidirectional nature makes it adept at capturing contextual information within time-series load profiles. *Second*, leveraging self-attention mechanisms enables BERT to effectively capture long-range dependencies, enhancing its performance in processing sequential data. *Third*, BERT's architecture outputs a sequence of probability distributions, enabling the generation of multiple results.

Our contributions are summarized as following:

- First, to our knowledge, we are the first to introduce a BERT-based approach for power system load profile inpainting. BERT-PIN surpasses the state-of-the-art by approximately 5%-30%.

- Second, unlike existing methods, BERT-PIN can produce multiple data restoration candidates with varying confidence levels. This feature is particularly valuable when exploring all potential options is necessary to ensure algorithm robustness.

- Third, BERT-PIN can restore multiple MDSs within long-time windows. This flexibility allows BERT-PIN to be used in Demand Response baseline estimation, as well as various downstream tasks, such as load profile disaggregation [54] and super resolution [55].

The rest of the chapter is organized as follows. Section 3.2 introduces the methodology, Section 3.3 introduces the simulation results in different cases, and Section 3.4 concludes the chapter.

3.2 BERT-Based Missing Data Restoration Method Design and Implementation Details

In this section, we first introduce the load profile inpainting problem formulation. Then, the BERT-PIN model architecture is illustrated in detail. Finally, the performance evaluation metrics are defined.

3.2.1 BERT-PIN Problem Formulation

Define $\boldsymbol{X}_m$ as a MDS (the green segment in Figure 3.2) in a time series load profile, $\boldsymbol{X} = [x_1, x_2, \dots, x_N]$, where N denotes the length of the time series.

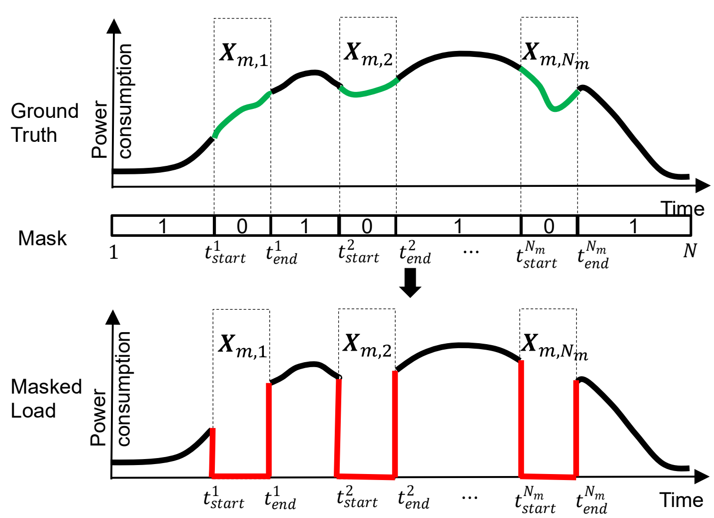


Figure 3.2 Illustration of missing data segments and masking methods.

As shown in Figure 3.2, if $\boldsymbol{X}$ contains N_m MDSs, i.e., $[\boldsymbol{X}_{m,1}, \boldsymbol{X}_{m,2}, \dots \boldsymbol{X}_{m,i}, \dots, \boldsymbol{X}_{m,N_m}]$, the objective of the inpainting problem is to find a set of model parameters, θ, to recover all MDSs using the non-missing data $\widetilde{\boldsymbol{X}}$ in $\boldsymbol{X}$ and the corresponding ambient temperature profile, $\boldsymbol{T} = [x_1, x_2, \dots, x_N]$ as inputs. So, the problem can be formulated as

$$[\widehat{\boldsymbol{X}}_{m,1}, \widehat{\boldsymbol{X}}_{m,2}, \dots, \widehat{\boldsymbol{X}}_{m,i}, \dots, \widehat{\boldsymbol{X}}_{m,N_m}] = f_\theta(T, \widetilde{\boldsymbol{X}}) \tag{3.1}$$

where $\widehat{\boldsymbol{X}}_{m,i}$ is the i^{th} recovered MDS.

Next, we will introduce the design of the proposed BERT-PIN model to address the problem formulated above.

3.2.2 BERT-PIN Model Architecture Overview

As illustrated in Figure 3.3, the proposed BERT-PIN model comprises three fundamental processes: input data adaptation, BERT model for recovering MDSs, and top candidate selection.

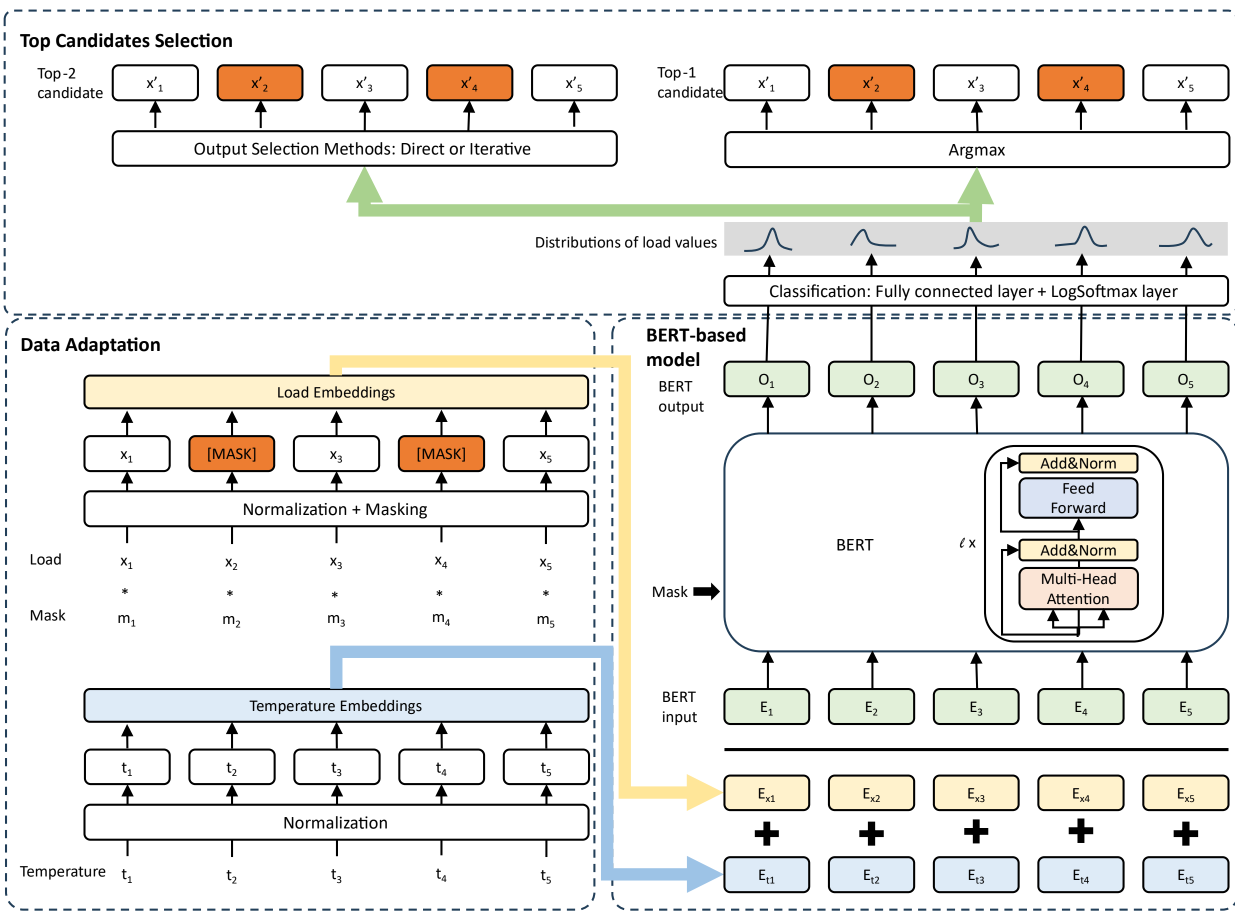


Figure 3.3 An overview of the overall modeling framework. The illustration of the BERT was inspired by [51]. It's an example of sequence length $N = 5$.

Given the established efficacy of the BERT model structure in addressing NLP tasks, our intention is not to alter the original BERT model architecture. However, as the BERT model is initially designed for processing NLP problems, its inputs are a sequence of word tokens. Therefore, in the *Input Data Adaptation* process, we first align the load profile with its corresponding temperature profile. Then, we divide the two aligned time-series profiles into segments to generate the load and temperature embeddings, respectively. As depicted in Figures 3.1 and 3.3, each of these segments resembles a missing word in a sentence, thus rendering the task of restoring missing data akin to recovering missing words within sentences or paragraphs.

During the *Top Candidates Selection* process, our objective is not to merely choose a single candidate with the highest likelihood. Instead, we aim to generate multiple candidates that meet predetermined confidence thresholds. This functionality enables BERT-PIN to generate an ensemble of patching options for a MDS. This feature proves especially critical when confronted with MDSs where several candidates display similar probabilities of being potential outcomes.

In the following sections, we will introduce those three main components in detail.

3.2.3 Data Adaptation

1) Preparation of Ground Truth Load Profiles

To prepare the training data, we collected smart meter data at 15-minute interval over a three-year period (2018-2020) from 8000 customers in North Carolina. Let P_i be the load profile for the i^{th} user containing N data points. Initially, we select a starting time (t_{start}) and calculate the end time (t_{end}) of the time series by $t_{end} = t_{start} + N - 1$. Next, we randomly draw N_{Agg} (ranges from 10 to 5000) load profiles (P) from the pool of 8000 load profiles and aggregate them into one load profile. This step is repeated for 200 times to ensure diversity in the training data. Finally, we normalize the aggregated load profile by its peak power (P_{MAX}) to create the ground truth load profile (X). This normalization ensures that X falls within the range of [0, 1]. This process can be summarized as

$$X = \frac{1}{P_{MAX}} \sum_{i=1}^{N_{Agg}} P_i \quad (t_{start}: t_{end}) \tag{3.2}$$

$$P_{MAX} = \max \sum_{i=1}^{N_{Agg}} P_i \quad (0: N) \tag{3.3}$$

2) Preparation of Load Profiles with Missing Data

To generate the time series data with N_m MDSs, we create a mask vector $\boldsymbol{M}$ for each $\boldsymbol{X}$, so that

$$\boldsymbol{M} = [m_i \; for \; i = 1:N], m_i = \begin{cases} 0, & missing \; data \\ 1, & otherwise \end{cases} \tag{3.4}$$

Then, the masked load profile, $\boldsymbol{X^m}$, can be represented as

$$\boldsymbol{X^m} = \boldsymbol{X} \cdot \boldsymbol{M} \tag{3.5}$$

Note that we set all missing data segments to be 0 kW because all the aggregated power values are greater than 0 kW, making it a unique value to be distinguishable.

3) Input Data Adaptation Layer

To align the BERT-PIN inputs with the BERT required format, we map the values in $\boldsymbol{X^m}$ to integers between 0 to 200. Note that we chose 200 because it strikes a balance between the model's size and resolution. In our dataset with 2000 aggregation level, the power range is [210, 1751] kW, as shown in Figure 3.4. So, the mapping provides a resolution of 8.755 kW.

The mapped load profile is embedded into a $N \times 200$ matrix, represented by the load embeddings (yellow boxes) in Figure 3.3. This data adaptation process allows the model to generate a probability distribution for each data point, making it possible to generate an ensemble of candidates. This transforms the missing data restoration problem, which is usually a regression problem, into a classification problem.

To address the influence of temperature on load [56], we include the normalized ambient temperature profile data, designated as T, as an additional modality input to assist in the recovery of missing load data. T is subjected to normalization based on the highest and lowest annual temperatures, ensuring that the normalized temperature values also range from 0 to 1. Then, T is rescaled to the [0, 200] range using the same approach employed for load embedding.

Lastly, we combine $\boldsymbol{X^m}$and T embeddings together by element-wise addition to obtain the final input matrix, the dimension of which is $N \times 200$. As an illustration, we show the input data adaptation process when $N = 5$ in Figure 3.3.

3.2.4 BERT Model

The combined embeddings of load and temperature can be directly fed into the BERT model using its original model architecture introduced in [52]. Represent a data sequence as $\boldsymbol{D} = \{\{\boldsymbol{k}_1, \boldsymbol{v}_1\}, \ldots \{\boldsymbol{k}_N, \boldsymbol{v}_N\}\}$, where $\boldsymbol{k}$ and $\boldsymbol{v}$ are N tuples of keys and values, respectively. For a query $\boldsymbol{q}$**,** the attention [57] over $\boldsymbol{D}$ in the BERT model is formulated as

$$Attention(\boldsymbol{q}, \boldsymbol{D}) = \sum_{i=1}^{N} \alpha(\boldsymbol{q}, \boldsymbol{k}_i)\, \boldsymbol{v}_i \tag{3.6}$$

where $\alpha(\boldsymbol{q}, \boldsymbol{k}_i) \in \mathbb{R}$ $(i = 1, \dots, N)$ are scalar values calculated through the dot product of the query vector and each key vector. These weights, referred to as attention scores, are normalized to ensure a sum of 1. This calculation is performed for every element in the input sequence. The output vectors are combined to form the final output of the model.

When processing sequential data, using self-attention can selectively attend to different parts of the input sequence based on their relevance to the given query vector. Thus, the BERT model can effectively capture long-range dependencies to form a context for the sequence. This significantly facilitates the recovery of the missing data because the context reflects the relevance of all known data points from all modalities (e.g., load and temperature) with the missing data points in those time-series profiles.

In the training, we use $\widehat{\boldsymbol{X}}^{\boldsymbol{1}}$ to calculate the cross-entropy losses [58] as

$$CrossEntropy = -\frac{1}{N} \sum_{o=1}^{N} \sum_{c=1}^{C} x_{o,c} \log\left(d_{o,c}\right) \tag{3.7}$$

where C is the number of classes, $x_{o,c}$ is the truth label denoting the power consumption value for observation o, and $d_{o,c}$ is the predicted probability observation o belonging to class c. N is the length of the sequence.

Note that the objective of BERT-PIN is to effectively restore the missing segments in a load profile, so we need to train the network to place more focus on restoring the MDSs. Thus, we construct the loss function as

$$Loss = (1 - \lambda) * CrossEntropy\left(\boldsymbol{X}, \widehat{\boldsymbol{X}}^{\boldsymbol{1}}\right)$$

$$+\lambda * CrossEntropy(\boldsymbol{X}_m, \widehat{\boldsymbol{X}}^{\boldsymbol{1}}_{\boldsymbol{m}}) \tag{3.8}$$

where λ is a hyper parameter for balancing between the global and local losses. Thus, $(1 - \lambda) * CrossEntropy\left(\boldsymbol{X}, \widehat{\boldsymbol{X}}^{\boldsymbol{1}}\right)$ represents the global loss for assessing how well the whole load profile can be restored, and $\lambda * CrossEntropy(\boldsymbol{X}_m, \widehat{\boldsymbol{X}}^{\boldsymbol{1}}_{\boldsymbol{m}})$ represents the local loss for assessing how well the MDSs are restored.

3.2.5 Top Candidates Selection Layer

As illustrated in Figure 3.3, we add a *Top Candidates Selection* process. Define the output of the BERT model, $\boldsymbol{O}$, as:

$$\boldsymbol{O} = BERT(\boldsymbol{X}^m, \boldsymbol{T}) \tag{3.9}$$

where $\boldsymbol{O}$ is an $N \times 200$ matrix.

We feed $\boldsymbol{O}$ into a classification layer, which comprises a fully connected layer followed by a SoftMax layer, to obtain an $N \times 200$ probability distribution matrix, $\boldsymbol{D}$. Note that the i^{th} column of $\boldsymbol{D}$ represents the probability distribution function (PDF) of the value of the i^{th} data point falling within the range of 1 to 200.

$$\boldsymbol{D} = Classification(\boldsymbol{O}) \tag{3.10}$$

Using $\boldsymbol{D}$, we can generate an ensemble of curves for patching an MDS rather than outputting just a single curve with the highest likelihood.

The conventional method for restoring the MDS from $\boldsymbol{D}$ is to use an *argmax* layer as illustrated as the orange boxes in Figure 3.3 by

$$\widehat{\boldsymbol{X}}^{\mathbf{1}} = argmax(\boldsymbol{D}) \tag{3.11}$$

where $\widehat{\boldsymbol{X}}^{\mathbf{1}}$ is considered as the *top-1 candidate*.

Nevertheless, it is often necessary to explore the top-2 or even the top-3 candidates as potential patching options to enhance the inclusion. As depicted in Figure 3.1, a single blank in a sentence can have multiple possible words (i.e., "France", "Quebec" and "Cameroon") that fit the context of the original sentence. When used for nationality identification, it becomes imperative to supply a list of regions where French is spoken as an official language, ranked in order of population size.

Similarly, there may exist multiple plausible curves for patching an MDS. The *top-1* method depicted in (11) selects the candidate with the highest probability for each missing data point. If the PDF of the missing data has a higher peak or are more narrowly concentrated around a particular value (see the red PDF in Figure 3.4), this method may have a higher accuracy in selecting the best candidate. However, when the PDF curve is a somewhat flattened one (see the green distribution in Figure 3.4), selecting only the best candidate will greatly limit the inclusion of the original missing data point.

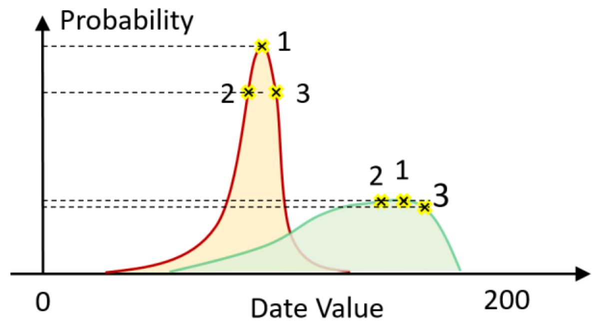


Figure 3.4 Illustration of the probability distribution in D.

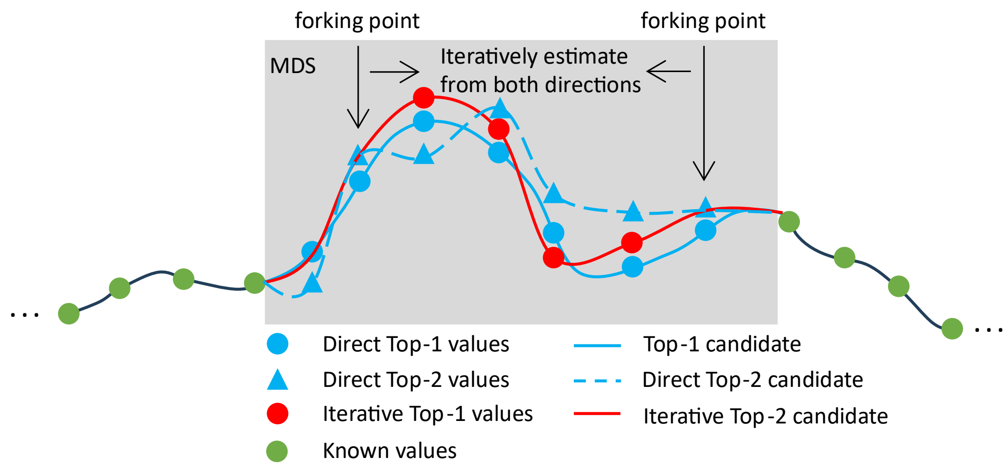


Figure 3.5 Illustration of the top-2 candidate selection process.

Therefore, we extend the top-1 method to the top-2 method, by comparing two selection approaches. Note that this approach can be readily extended to top-X candidate selection by repeatedly applying the same selection criterion.

Method 1: The simplest approach involves selecting the candidate with the second-highest probability from each probability distribution to create an additional set of profiles. As illustrated in Figure 3.5, the solid blue curve connecting the blue circles represents the curve generated by the top-1 candidates, while the dashed blue curve connecting the blue triangles represents the patching curve generated by directly selecting the top-2 candidates using Method 1.

However, Method 1 suffers from a significant drawback: it ignores the autocorrelation among adjacent data points. In practical terms, selecting the second candidate may lead to shifts in the probability density functions (PDFs) of subsequent missing data points.

Method 2: To overcome the inherent limitation of Method 1, we propose an iterative selection approach centered on the initial identification of "fork points". By using the identified fork points as reference pivots, we can create the top-2 candidate curve accounting for the autocorrelation among the subsequent data points.

Algorithm 1 *Iterative Top Candidates Selection*.
Given the output of BERT-PIN $\boldsymbol{D}$, find the fork points located on both the left and right sides of the top-1 curve. Next, replace the top-1 values at the fork points with the top-2 values. Then, iteratively generate the remaining missing data using top-1 values. Note that e is the threshold for selecting the fork points.

Let $l = (t_{end} - t_{start})/2$
for $t = t_{start}: t_{start} + l$ **do**
    # find the left fork point.
    if $top1(\boldsymbol{D}_t) - top2(\boldsymbol{D}_t) < e$ **do**
        $fork_{left} = t$
        x_t^{masked} = index of $top2(\boldsymbol{D}_t)$
        break
    else
        x_t^{masked} = index of $top1(\boldsymbol{D}_t)$
end for
for $t = t_{end}: t_{end} - l$ **do**
    # find the right fork point.
    if $top1(\boldsymbol{D}_t) - top2(\boldsymbol{D}_t) < e$ **do**
        $fork_{right} = t$
        x_t^{masked} = index of $top2(\boldsymbol{D}_t)$
        Break
    else
        x_t^{masked} = index of $top1(\boldsymbol{D}_t)$
end for

for $k = fork_{left} - t_{start}: l$ **do**
- shift the daily profile by k steps to the left.
- feed shifted data into the BERT-PIN model.
  $\boldsymbol{D}' = BERT(\boldsymbol{X}^{shifted}, \boldsymbol{T}^{\boldsymbol{shifted}})$
- update the first unknown value in $\boldsymbol{X}^{masked}$
  $x_{t_{start}+k}^{masked}$ = index of $top1(\boldsymbol{D}'_{t_{start}})$

end for

for $k = t_{end} - fork_{right}: l$ **do**
- shift the daily profile by k steps to the right.
- feed shifted data into the BERT-PIN model.
  $\boldsymbol{D}' = BERT(\boldsymbol{X}^{shifted}, \boldsymbol{T}^{\boldsymbol{shifted}})$
- update the last unknown value in $\boldsymbol{X}^{masked}$
  $x_{t_{end}-k}^{masked}$ = index of $top1(\boldsymbol{D}'_{t_{end}})$

end for

Use the final $\boldsymbol{X}^{masked}$ as the restored *top-2* load profile, $\widehat{\boldsymbol{X}}$.

Define the "right-side" (or "left-side") fork point as the first point, counting from the rightmost (or leftmost) side of the MSD, where the probability difference between the top-1 and top-2 candidates is less than e. As depicted in Figure 3.5, after identifying the fork points, we can start an iterative process to estimate the top-1 values for the remaining missing data points, one at a time, by shifting the load profile either forward or backward. This method ensures that the shifted load profile maintains the same masking position as the original one, effectively capturing the shift in PDFs for subsequent data points.

The resultant top-2 curve is shown by the red continues line in Figure 3.5 and the detailed algorithm description is depicted in Algorithm 1.

3.2.6 Performance Metrics

The performance metrics used for evaluating the accuracy of the restored data segments are calculated as

$$MPE = \frac{1}{K}\sum_{t=1}^{K}\frac{|\hat{x}_t^m - x_t^m|}{x_t^{event}} \tag{3.12}$$

$$RMSE = \sqrt{\frac{1}{K}\sum_{t=1}^{K}(\hat{x}_t^m - x_t^m)^2} \tag{3.13}$$

$$PKE = \frac{|\hat{x}^{MAX} - x^{MAX}|}{x^{MAX}} \tag{3.14}$$

$$VLE = \frac{|\hat{x}^{MIN} - x^{MIN}|}{x^{MIN}} \tag{3.15}$$

$$EGYE = \frac{|\sum_{t=1}^{K}\hat{x}_t^m - \sum_{t=1}^{K}x_t^m|}{\sum_{t=1}^{K}x_t^{event}} \tag{3.16}$$

$$FCE = \frac{1}{K}\sum_{t=1}^{K}\frac{|FFT(\hat{x}_t^m) - FFT(x_t^m)|}{FFT(x_t^m)} \tag{3.17}$$

where K is the total number of data points in the MDS, $\hat{x}$ represents the restored data segment, x^{MAX} and $x^{MIN}$are the maximum and minimum power values in the MDS, respectively, and FFT stands for Fast Fourier Transform. These indices offer insights into different aspects of errors, including point-to-point errors, inaccuracies in peak and valley points, discrepancies in total energy consumption, and errors within the frequency domain components.

3.3 Simulation Results

In this section, we evaluate BERT-PIN's capability to recover varying numbers of MDSs in different scenarios and compare it with three benchmark methods: LSTM [33], SAE [35], and

Load-PIN [50]. The evaluation encompasses various aggregation levels and training data volumes. Additionally, we assess the efficacy of the top-1 and top-2 candidate methods in CVR baseline estimation to highlight the benefits of having several restoration alternatives at our disposal.

3.3.1 Data Preparation

The load profiles used in this study consists of 15-minute resolution smart meter data obtained from 8,000 users (including residential, commercial, and industrial users) in North Carolina between 2018 and 2020. The corresponding temperature data is downloaded from the National Oceanic and Atmospheric Administration (NOAA) [59] website and is used as a second modality input.

We randomly select a group of users from the pool of 8,000 and combine their data to form aggregated load profiles. The selected group size ranges from 10 to 5000, to show the model's capability to deal with different scenarios. An example of the distribution of load values and daily peak values in 2000 user aggregation level is shown in Figure 3.6. Moreover, to increase the data diversity, the selection is repeated for multiple times. These aggregated load profiles are aligned with the temperature profile and then normalized based on the annual load and temperature peaks, respectively. Then, we partition the profiles into either daily (96 data points) or weekly profiles (672 data points). Each missing load data segment is consistently set at 4 hours (16 data points), a choice guided by the observation that around 70% of missing load data segments in actual utility data are less than 4 hours in duration. It's important to note that there are no missing data segments in the temperature profile. The dataset is divided into an 80% training set and a 20% testing set. In the next sections, we will show the simulation results.

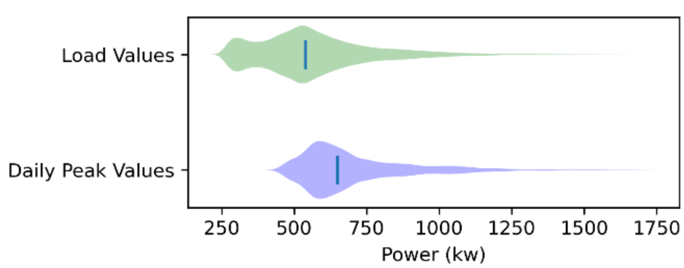


Figure 3.6 Distribution of the 2000 users aggregated load values and daily peak values.

3.3.2 Restoration of One MDS

In this base case, we aggregate randomly selected 2000 users for 200 times. Our objective is to evaluate the performance of BERT-PIN when restoring a single 4-hour MDS. The daily load profile, with a 4-hour gap, together with the temperature data of the same day is used as input. And BERT-PIN restores the MDS. Several restored daily load profiles are shown in Figure 3.7, while the corresponding performance metrics are depicted in Figure 3.8 and summarized in Table 3-2. It is evident from the outcomes that BERT-PIN (represented by the red lines) exhibits the smallest medians and ranges in this case. Furthermore, Table 3-2 provides the mean errors for each distribution shown in Figure 3.7. The results clearly indicate that BERT-PIN achieves the lowest errors and outperforms the benchmark methods by 5%-27% across all six indexes. This demonstrates that our model consistently outperforms the existing approaches by higher missing data patching accuracy.

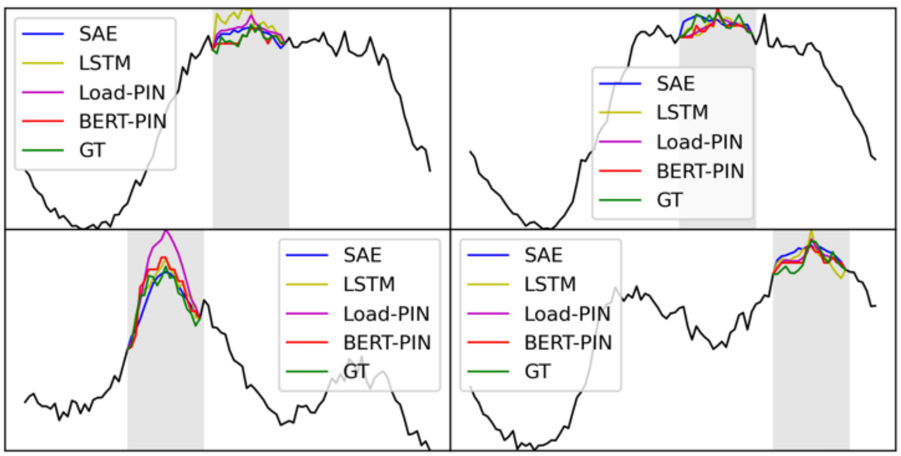


Figure 3.7 Examples of missing data restoration with central-mask using different models: SAE (blue), LSTM (yellow), Load-PIN (magenta), BERT-PIN (red), and ground truth (green).

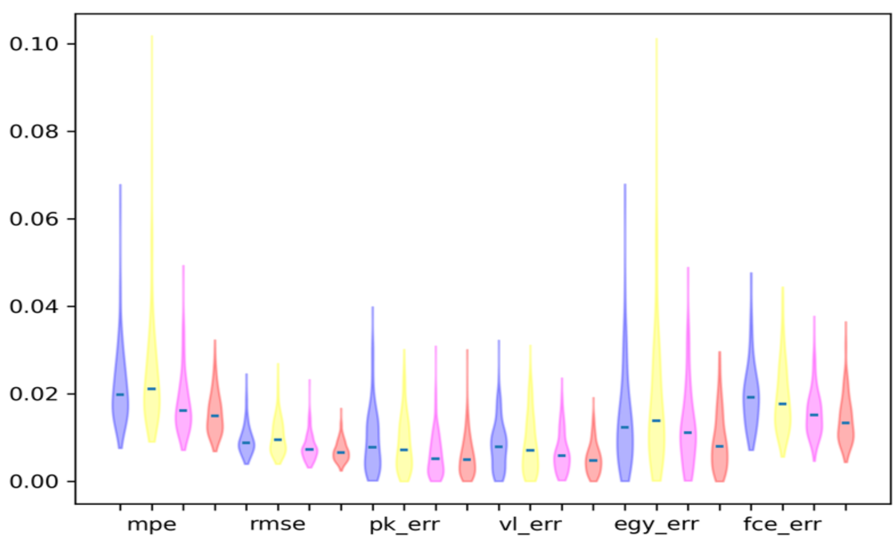


Figure 3.8 Error distributions of different models: SAE (blue), LSTM (yellow), Load-PIN (magenta), BERT-PIN (red).

Table 3-2 Errors of Single MDS inpainting

	SAE (%)	LSTM (%)	Load-PIN (%)	**BERT-PIN** (%)	**Improvement**
MPE	2.231	2.414	1.670	**1.523**	8.80%
RMSE	1.035	1.112	0.7951	**0.7404**	6.88%
PKE	1.065	0.8491	0.6183	**0.5130**	17.10%
VLE	0.8687	0.8991	0.6185	**0.5870**	5.09%
EGYE	1.525	1.762	1.165	**0.8410**	27.81%
FCE	2.138	2.046	1.615	**1.509**	6.56%

3.3.3 Expand the Simulation to Different Aggregation Levels and Different Data Sizes

1) Accuracy vs. Aggregation Level

In the preceding section, we maintained a fixed aggregation level of 2000. However, we now extend the aggregation level range from 10 to 5000 to demonstrate BERT-PIN's ability to handle diverse datasets.

The findings are illustrated in Figure 3.9. As the aggregation level rises, errors diminish. A lower aggregation level typically implies greater randomness in the aggregated load profile, posing challenges for MDS restoration by the models. Moreover, irrespective of the aggregation level, BERT-PIN consistently demonstrates higher accuracy compared to the other three methods. This showcased BERT-PIN's effectiveness across various load aggregation levels.

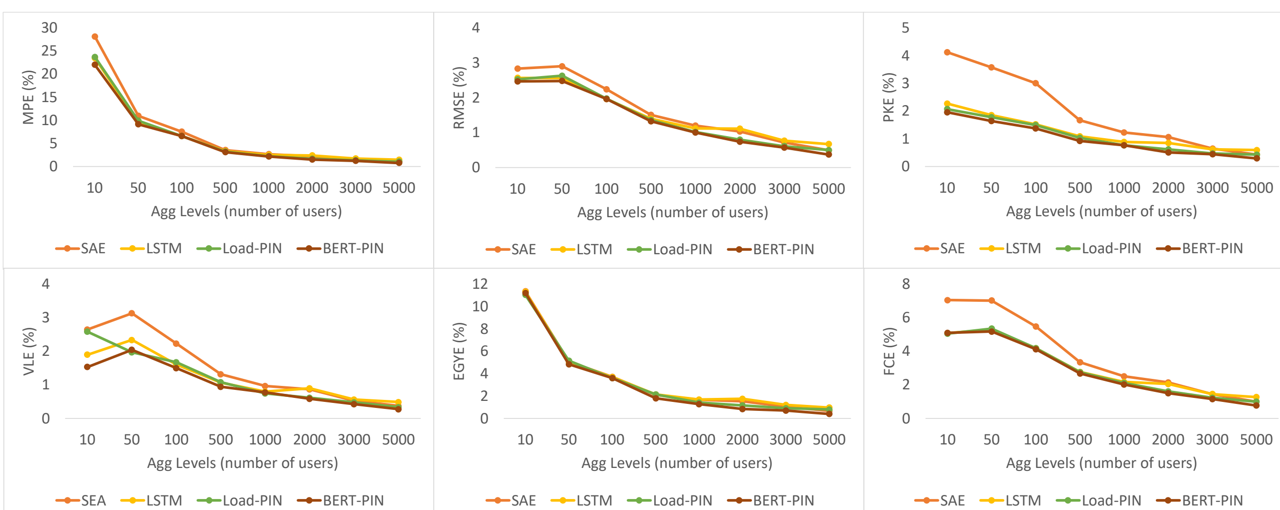


Figure 3.9 Errors with different aggregation levels.

1) Accuracy vs. Data Size

In this section, we assess how the size of the dataset affects training precision, taking the aggregation of 2000 loads as a case study. The model is trained with datasets ranging from 10 to 400 sets of load profiles. As illustrated in Figure 3.10, a clear reduction in error rates is observed as the dataset size expands from 10 to 50 sets of load profiles, after which the improvements plateau. This suggests that aggregating data from 2000 randomly selected users and replicating this aggregation 200 times to generate 200 aggregated load profiles yields adequate outcomes. Expanding the dataset beyond this point does not significantly enhance model performance but results in disproportionately extended training durations.

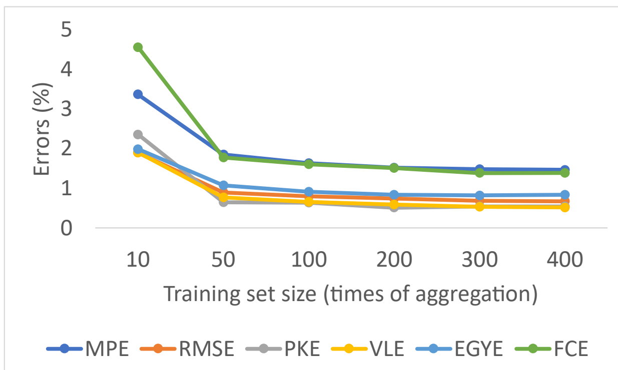


Figure 3.10 BERT-PIN errors with different training data sizes.

3.3.4 Top-2 Candidate Selection

As described in section 3.2, BERT-PIN has the capability to generate multiple restored load profiles using *Top Candidate Selection* methods. In this section, we compare the two top candidate selection methods discussed in section 3.2.5, taking the selection of the top-2 candidates as an illustrative example. It's important to note that this method can be employed iteratively to choose the top-X candidates.

The result is presented in Figure 3.11. BERT-PIN is the default *top-1* candidate where the candidate with the highest probability value is selected. BERT-PIN_2 results are obtained using Method 1, where candidates having the *second highest* probability is selected. BERT-PIN_2i results are obtained using method 2, where an iterative method is used to select a fork point, based on which, subsequent restoration candidates are selected.

The errors of *top-1* candidate and *top-2* candidate with different parameters are computed and presented in Table 3-3.

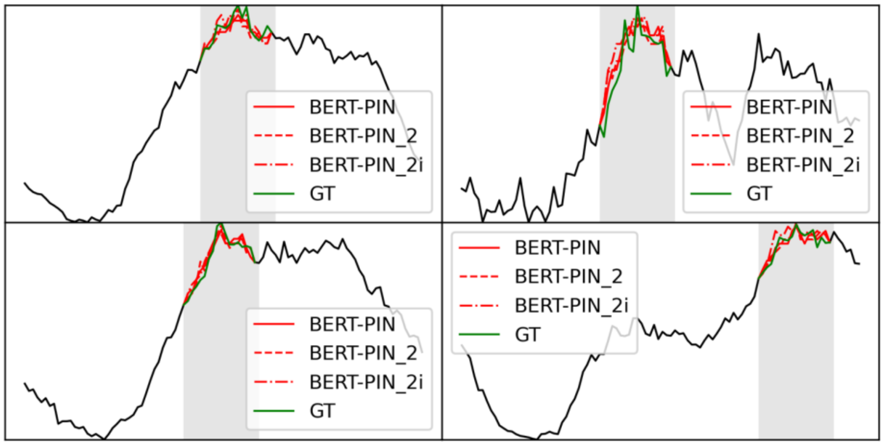


Figure 3.11 Examples of top-1 and top-2 restored MDSs.

Given that the process can be repeated for generating an ensemble of candidates for MDS restoration, we proceed to assess the quality of the top-2 results by calculating the "percentage-of-closer-points" (PoCP). This metric signifies the percentage of estimated points that are closer to the ground truth when compared to those estimated using the top-1 method, indicating the potential expansion of inclusion through the incorporation of the top-2 restored MDS.

Table 3-3 Errors of top-1 candidate, top-2 candidates and combined outputs (%)

	BERT-PIN	BERT-PIN_2	BERT-PIN_2i						**Combine**
			e=0.8	e=0.5	e=0.3	e=0.1	e=0.05	e=0.02	
MPE	**1.523**	**1.744**	2.556	2.433	2.407	2.138	1.87	1.761	1.211
RMSE	**0.7404**	0.9144	1.317	1.173	1.211	1.071	0.896	**0.899**	0.577
PKE	**0.5130**	**0.5917**	1.044	0.939	0.927	0.871	0.665	0.663	0.426
VLE	**0.5870**	0.9260	0.827	0.617	0.817	0.573	**0.543**	0.669	0.407
EGYE	**0.8410**	0.9582	1.618	1.441	1.412	1.301	1.043	**0.986**	0.633
FCE	**1.509**	1.942	2.447	2.18	2.273	2.007	**1.727**	1.78	1.209
PoCP	-	45.12%	23.88%	24.40%	22.94%	17.29%	12.25%	**6.29%**	-

From the results shown in Table 3-3, we made the following observations:

- As expected, the result accuracy of the *top-1* method is higher than the *top-2* method.
- Nevertheless, the *top-2* method can generate points that are closer to the ground truth. This is because the BERT output is generated based on probabilities, and the second-best candidate could also be close to the actual missing data, just with reduced likelihood.

- Compared with the direct *top-2* candidate selection method, iterative *top-2* candidate selection method has lower PoCP. However, since the points selected independently lack autocorrelation among adjacent points, we consider the results to be less valuable for time-series MDS.

- Among all iterative top-2 methods, when $e = 0.5$ (column highlighted in yellow), the PoCP is the highest. This demonstrates that the fork point selection influences the range of inclusion. When e becomes smaller, the iterative top candidate selection algorithm tends to identify a fork point later. Consequently, the top-2 candidate curve closely resembles the top-1 candidate, leading to identification of fewer "better" points.

- When we merge the top-1 and top-2 candidate by selecting the best value for each timestamp (column highlighted in gray), the errors are even smaller than the top-1 result. This demonstrates the benefit of incorporating both candidates when performing downstream tasks. This advantage is reflected in the wider range of data encompassed because, with both curves, we capture data points closer to the original missing data points.

Although the top-1 candidate delivers the highest accuracy, integrating either direct or iterative approaches with top-2 candidates broadens the inclusiveness of missing data estimation. Combining the top-1 and top-2 candidates enhances the robustness of our simulations beyond what's achievable with only the top-1 candidate. This approach also showcases BERT-PIN's capability to generate an ensemble of data segments for more comprehensive missing data restoration.

3.3.5 Restoration of Multiple MDSs

In this section, we showcase the performance of BERT-PIN for restoring multiple MDSs within a weekly load profile. This use case is essential for Conservation Voltage Reduction (CVR) baseline estimation. CVR is a frequently adopted strategy among utility companies to manage peak loads. During CVR events, the voltage on a distribution feeder is deliberately decreased by 2-4% over a period ranging from 1 to 4 hours. CVR can be executed on multiple days in a hot summer week or a cold winter week. As depicted in Figure 3.12, accurately assessing the actual load reduction achieved through CVR often involves estimating the baseline energy consumption (indicated by the red lines) that would occur in the absence of CVR implementation.

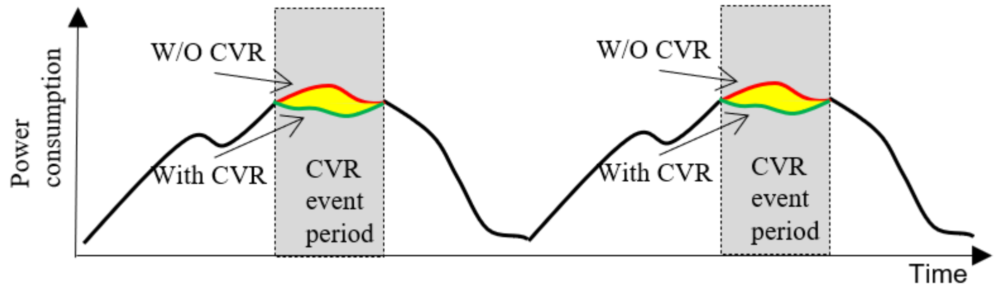


Figure 3.12 An illustration of the CVR baseline estimation.

To train BERT-PIN for CVR baseline estimation, we apply 4-hour masks targeting the peak load periods within weekly load profiles. The number of masks is randomly selected between 1 and 7, thereby representing different numbers of CVR event days. The hyperparameters of the BERT-PIN model remain consistent with those employed in the single-event scenario. As can be seen from Figure 3.13 and Table 3-4, BERT-PIN outperforms all existing methods by a large margin.

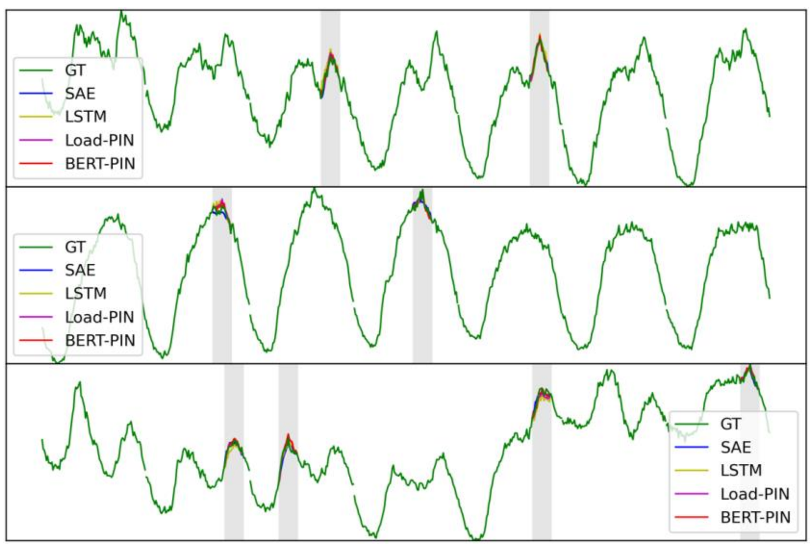


Figure 3.13 Missing data restoration examples: SAE (blue), LSTM (yellow), Load-PIN (magenta), BERT-PIN (red), and ground truth (green).

Table 3-4 Errors of Multiple MDSs Inpainting

	SAE (%)	LSTM (%)	Load-PIN (%)	**BERT-PIN (%)**	**Improvement**
MPE	7.615	6.389	5.168	**4.837**	**6.40%**
RMSE	3.310	2.848	2.301	**2.221**	**3.48%**
PKE	3.362	2.193	1.769	**1.746**	**1.30%**
VLE	2.697	2.407	1.691	**1.627**	**3.78%**
EGYE	4.874	3.822	3.257	**2.699**	**17.13%**
FCE	6.828	5.601	4.808	**4.501**	**6.39%**

This section demonstrates that BERT-PIN can effectively restore multiple missing data segments and accurately estimate the baseline of demand response programs with CVR baseline estimation serving as a practical case study.

3.4 Conclusion

In this study, we introduced a cutting-edge framework called BERT-PIN (Bidirectional Encoder Representations from Transformers-based Profile Inpainting Network), leveraging the advanced capabilities of the BERT model to restore multiple missing data segments within load profiles. Our contributions can be summarized in three main aspects. First, to the best of our knowledge, we are the first to introduce a BERT-based approach for power system load profile inpainting, leading to higher accuracy. Second, unlike existing methods, BERT-PIN can produce multiple data restoration candidates with varying levels of confidence, offering enhanced flexibility in data recovery. Third, BERT-PIN can restore multiple missing data segments (MDSs) within very long-time windows.

We tested our BERT-PIN model on different datasets from North Carolina, covering various scenarios like different levels of data aggregation and dataset sizes. The results show that BERT-PIN outperforming other methods by 5%-30% in accuracy for filling in both single and multiple gaps in the data. Moreover, using a combination of the top-1 and top-2 predictions allows us to predict missing data more comprehensively. Overall, BERT-PIN proved effective in specific tasks like filling in weekly data gaps and demand response baseline estimation, outperforming traditional methods in accuracy.

In our future research endeavors, we intend to apply the BERT model, already pre-trained, to a variety of downstream tasks such as Super Resolution, Load Disaggregation, Anomaly Detection, and Energy Forecasting. Our goal is to investigate how the attention mechanism and contextual capabilities of BERT can enhance both the precision and reliability of these specific tasks.

4. FINE-TUNING LLM FOR MISSING DATA RESTORATION

4.1 Background

Machine Learning has been extensively used in power system load profile analysis tasks, including missing data restoration, load forecasting, load disaggregation, and load profile generation. Training a large machine learning model from scratch typically requires substantial data and considerable training time. However, due to security and privacy concerns, utilities cannot share vast amounts of load data with the research community for these analyses. As an alternative, synthetic load profiles derived from real datasets have been used [60]. However, this approach also requires significant effort in training generative machine learning models.

In this chapter, we propose using fine-tuned Large Language Models (LLMs) for load profile analysis to address the challenges of data scarcity while reducing computational time and cost. Initially trained on vast, diverse datasets, LLMs develop an extensive understanding of patterns and relationships within data. By fine-tuning these models on specific tasks like load profile analysis, we leverage their pre-existing knowledge, reducing the need for large, task-specific datasets.

Fine-tuning is data-efficient, requiring only a small dataset to adapt the model to specific tasks, a significant advantage when data is scarce, sensitive, or expensive. This approach also demands fewer computational resources than training a model from scratch since it only adjusts a portion of the model's weight and is performed on smaller datasets [61].

Furthermore, fine-tuned LLMs offer the flexibility to adapt to various downstream tasks without extensive reprogramming, allowing the same base model to be used for different analysis tasks within the same field, such as load forecasting or missing data restoration. This flexibility ensures high levels of performance even when large, comprehensive datasets are unavailable due to privacy, security, or practicality concerns.

In this chapter, we will demonstrate how to fine-tune LLMs for conducting the missing data restoration task, illustrating both the process and the benefits of our approach.

4.1.1 Missing Data Restoration

Missing data is a common occurrence in load profile processing within power systems. These data gaps often stem from temporary equipment malfunctions or deliberate human interventions. Equipment malfunctions lead to periods without consumption measurements, resulting in missing data segments. Conversely, operations like demand response (DR) and conservation voltage reduction (CVR) do not directly interrupt consumption measurements but make the baseline consumption level unknown. This baseline, representing the consumption level if the operation were not implemented, constitutes another form of missing data. The absence of load data significantly impacts power distribution system analysis, affecting tasks such as modeling load behaviors, studying renewable integration, and designing demand response programs. Therefore, developing an algorithm to restore missing data has become a crucial solution to these challenges.

In the field of load profile inpainting, methods for restoring missing data can be broadly classified into two categories: *model-based* and *data-driven* approaches. Model-based methods utilize physical system models to simulate the missing data segments. For example, to estimate the CVR baseline, researchers use distribution system topology and load models to predict load variations in response to voltage changes [62]-[27]. These methods depend on accurate physics-based models that require numerous inputs and careful parameter tuning.

The data-driven approach provides a comprehensive solution for restoring missing data and has gained significant popularity due to its convenience. Researchers using this approach often categorize load profiles based on factors such as day type, weather conditions, and shape characteristics, then restore missing data segments by referencing load profiles with the closest similarity [28]-[31]. These methods are straightforward, easy to implement, and explainable. However, they often depend on subjective selections of similarity metrics and weights, typically defined by human analysts using weighted averages of various factors, making the accuracy contingent on the analyst's choices.

Recently, regression models have emerged as viable alternatives for load profile inpainting. Models such as Linear Regression [32], Long Short-term Memory networks (LSTM) [33], Stacked Autoencoder (SAE) [35], Gaussian Regression [37], Support Vector Regression (SVR) [38][39] have been increasingly employed to address the missing data restoration problem. These models generally achieve higher estimation accuracy compared to similarity-based methods due to their ability to capture nonlinear patterns in the data. However, it is important to note that deep-learning-based methods are less explainable and incur higher computational costs compared to similarity-based approaches. In recent years, hybrid solutions that combine multiple regression models have been proposed for baseline estimation and missing data restoration [40]-[42].

Additionally, GAN-based methods have also been utilized for restoring missing load data [44]-[50] and outperform model-based, similarity-based, regression-based, and other GAN-based methods in restoring missing data segments (MDSs) of fixed length. Moreover, a Bidirectional Encoder Representations from Transformers (BERT) based method is also employed to restore missing data in a daily load profile [87]. This method outperforms all the existing load inpainting methods in accuracy and can produce multiple data restoration candidates with varying confidence levels. Consequently, we select it as the benchmark method in this chapter.

4.1.2 Application of LLM in Different Areas of Research

However, all the aforementioned methods are trained from scratch, necessitating a large amount of training data and extended training time. To address this, we propose solving the load profile inpainting problem by fine-tuning a Large Language Model (LLM). Fine-tuning is a powerful technique that leverages the pre-trained knowledge and general language understanding already present in the LLM. This allows for efficient adaptation to new tasks or domains with relatively small amounts of additional data. Fine-tuning has been widely used to customize language models for various applications. Table 4-1 provides a list of the main applications of LLMs in different areas of research.

Although LLMs have been applied to various fields, their application in power systems is still relatively limited. LLMs have been utilized in smart grid cybersecurity studies [101][102], and J. Ruan et al. have discussed the potential security threats associated with applying LLMs to power system [103]. However, research on using LLMs to restore missing data in power systems and estimate DR baselines is lacking. Investigating how LLMs perform compared to specifically designed missing data restoration models is a promising area of research.

Table 4-1 Applications of Large Language Models in different areas of research

Field of application		Tasks	References	Main contribution
Engineering	Software engineering	Code generation	[88]	Investigating example selection for In-Context Learning to generate code effectively.
			[89]	Propose a novel prompting technique to improve the performance of LLMs in code generation.
		Self-Debug	[90]	Identify the mistakes in code by investigating the execution results and explaining the generated code in natural language and streamline the debugging process.
	Mechanical engineering	Calculations in mechanical engineering	[91]	Encountered incorrect procedures, formulas, or results. Showing that ChatGPT should not be relied upon solving practical mechanical problems.
	Mathematical engineering	Mathematical calculations	[92]	Investigate the mathematical capabilities of ChatGPT by testing it on publicly available datasets.
		Mathematical education	[93]	Use ChatGPT as a tool for teaching and learning mathematics
	Manufacturing	Support design, manufacturing, and engineering education	[94]	Propose a technology development roadmap to successfully integrate ChatGPT into the manufacturing industry.
		Manufacturing troubleshooting.	[95]	Evaluated ChatGPT's ability in technical matters, including printing parameters and bed detachment, warping, and stringing issues.
Finance	Risk assessment	Risk assessment, algorithmic trading, and low-code development.	[96]	Propose an end-to-end framework, FinGPT, to serve as a catalyst for stimulating innovation in the finance domain.
Medical	medical education	Assistant learning and problem solving.	[97]	Shows ChatGPT's performance in the United States Medical Licensing Exam (USMLE) was comparable to or exceeded the passing threshold, indicating its proficiency in medical knowledge without requiring specialized training or reinforcement.
	radiologic decision-making	AI-based clinical decision-making	[98]	Highlighting its feasibility and potential benefits in improving clinical workflow and ensuring responsible use of radiology services
	clinical genetics	Answer genetics-related questions	[99]	ChatGPT's performance did not significantly differ from humans when answering genetics-related questions. However, the model demonstrated better accuracy on memorization-type questions compared to questions requiring critical thinking.
	Patient care	Gathering patient data, administering surveys or questionnaires.	[100]	The recent popularity of LLMs can potentially not only improve patient confidence in interacting with such chatbots but also improve upon the services provided.

In this chapter, we propose a two-stage strategy to fine-tune the ChatGPT to minimize data requirements in load profile analysis, demonstrated through the restoration of missing data in power system load profiles. The contributions are three-fold.

- **Pioneering Application:** To the best of our knowledge, this is the first application of a fine-tuned LLM to a specific power system data analytics task, specifically missing data restoration and baseline estimation. The fine-tuned model achieves acceptable accuracy in missing data restoration, comparable to state-of-the-art specifically designed models.
- **Innovative Fine-Tuning Strategy:** We introduce a two-stage fine-tuning strategy using few-shot transfer learning. By leveraging a very small amount of training data, we can transfer the knowledge from a general language model to the task of power system missing data restoration. This approach significantly reduces the data needs and training time for load profile analysis.
- **Empirical Evaluation:** We evaluate the fine-tuned model using real smart meter data collected from various households and conduct a comprehensive study on the effect of different fine-tuning strategies. This offers valuable insights, including the importance of prompt engineering, optimal utilization of fine-tuning samples, and the efficiency of few-shot learning. These findings enhance understanding of model behavior and inform other load profile analysis tasks.

The rest of the chapter is organized as follows. Section 4.2 introduces the fine-tuning methodology, Section 4.3 introduces the simulation results in different cases, and Section 4.4 concludes the chapter.

4.2 Fine-Tuning LLM for Missing Data Restoration Method Design Details

This section begins by formulating the missing load profile restoration problem. We then provide an overview of the proposed two-stage fine-tuning strategy, detailing its key components and functionalities. Additionally, we introduce performance evaluation metrics that will be used as benchmarks to assess the accuracy of the fine-tuned model.

4.2.1 Missing Data Restoration Problem Formulation

In this context an *event* refers to a missing data segment (MDS), visually depicted as the green segment in Figure 4.1. The event is characterized by the starting time T_{start} and the ending time T_{end}. Let $\boldsymbol{X} = [x_1, x_2, \dots, x_N]$ represent a historical time series of load, where N denotes the length of the time series. Consequently, $\boldsymbol{X}$ can be divided into three distinct parts:

$$\boldsymbol{X} = [\boldsymbol{X}_{pre}, \boldsymbol{X}_{event}, \boldsymbol{X}_{post}] \tag{4.1}$$

where $\boldsymbol{X}_{event} = [x_{T_{start}}, \dots, x_{T_{end}}]$.

The objective of missing data restoration problem is to recover the missing portion of the load, denoted as $\boldsymbol{X}_{event}$, utilizing the pre-event load $\boldsymbol{X}_{pre}$ and post-event load $\boldsymbol{X}_{post}$. In essence, the missing data imputation problem can be defined as follows:

$$\widehat{\boldsymbol{X}}_{event} = f(\boldsymbol{X}_{pre}, \boldsymbol{X}_{post}) \tag{4.2}$$

where f is the mapping function.

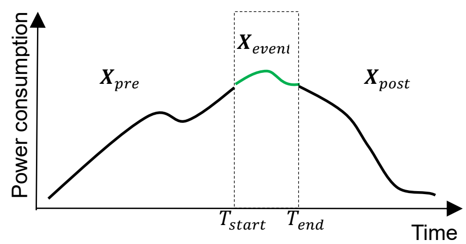


Figure 4.1 Illustration of missing data imputation problem.

Moreover, the *event* can also be an unknown Demand Response (DR) baseline or Conservation Voltage Reduction (CVR) baseline. Because the DR or CVR baseline estimation is to recover the would-have-been load profiles assuming the DR or CVR program had not been executed, which can be considered as a special case of the missing data imputation problem. Thus, the missing data imputation method proposed in this chapter can be applied to DR or CVR baseline estimation, which is valuable for load service providers to evaluate the performance of DR or CVR.

4.2.2 Fine-tuning Strategy Overview

Although many machine learning based methods can be used to restore missing data in load profiles, we propose this Large Language Model fine-tuning method for two key reasons: First, our method requires significantly less data and training time compared to models trained from scratch, yet it achieves comparable accuracy in missing data restoration. Second, this approach eliminates the need to design complex models and training processes. Instead, we only need to develop a fine-tuning strategy based on the well-knownChatGPT-3.5 model.

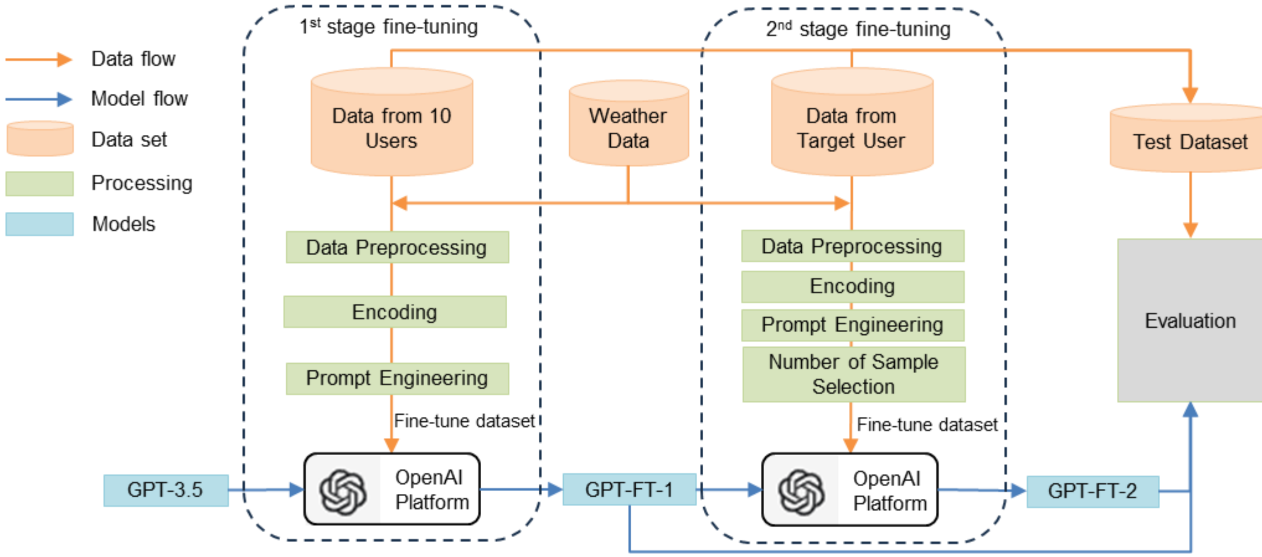


Figure 4.2 An overview of the two-stage fine-tuning strategy.

The overview of the two-stage fine-tuning strategy is illustrated in Figure 4.2. In the first stage, the standard GPT-3.5 model is fine-tuned using data collected from 10 users similar to the target user. An encoding method and prompt technique are employed to create the intermediate model, GPT-FT-1. In the second stage, we further fine-tune this intermediate model using data from the target user. Different sample sizes are tested in this stage to achieve the best fine-tuning results.

4.2.3 1st Stage Fine-tuning

In this section, we will outline the main steps involved in the first stage of fine-tuning. These steps include data preprocessing, smart meter data-to-words encoding, prompt engineering, and OpenAI fine-tuning.

1) Data Preprocessing

Unlike the BERT-PIN model [87] which utilizes data from 8000 users over a three-year period, our first stage fine-tuning uses data collected from only 10 users over three months. The load profiles are segmented into daily load profiles for this process. To generate the time series data with MDS, we create a mask vector $\boldsymbol{M}$ for each $\boldsymbol{X}$, so that

$$\boldsymbol{M} = [m_i \; for \; i = 1{:}N], m_i = \begin{cases} 0, \; missing \; data \\ 1, otherwise \end{cases} \tag{4.3}$$

Then, the masked load profile, $\boldsymbol{X}^m$, can be represented as

$$X^m = X \cdot M \tag{4.4}$$

Note that we set all missing data segments to be 0 kW because all the power values are within [210, 1751] kW, making “0” a unique value to be distinguishable.

To address the influence of temperature on load, we include the normalized ambient temperature profile data, designated as T, as an additional modality input to assist in the recovery of missing load data. X^m and T are subjected to normalization based on the highest and lowest annual power value and temperatures respectively, ensuring that the normalized power and temperature values range from 0 to 1. Then, X^m and T are rescaled to the [0, 200] range.

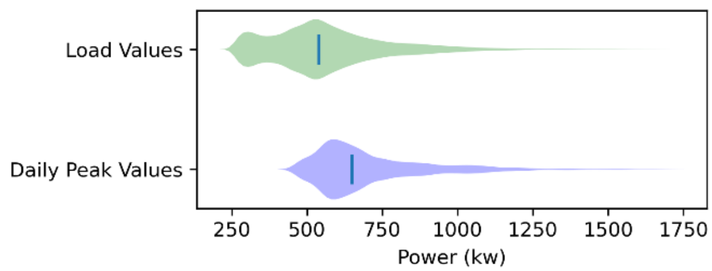


Figure 4.3 Distribution of the load values and daily peak values.

2) Encoding Smart Meter and Temperature data to Words

ChatGPT is primarily a large language model, so our approach involves converting load and temperature values into words that can be easily interpreted by ChatGPT. We propose an encoding method to transform smart meter data and temperature values into words.

After data preprocessing, both load and temperature data are represented by integers between 0 and 200. We use a five-digit ternary code to represent each load or temperature value, with each digit having three possible values: "0", "1", and "2". This five-digit ternary code can represent $3^5 = 243$ different numbers, which is sufficient for our range. We then convert the ternary code into a "word" by replacing "0", "1", and "2" with "L", "M", and "H", respectively. Missing values are encoded as "OOOOO" to distinguish them from actual data.

To further explore time alignment capabilities, we combine the load and temperature encodings at each timestamp into a single long "word", where the first five letters represent the load value, and the last five letters represent the temperature value.

An illustration of this load-temperature to "word" encoding is shown in Figure 4.4. The example includes a sequence of three time points with a missing value in the middle. The load and temperature sequences are encoded into a long "word" sequence. This sequence is then fed into ChatGPT using a proper prompting strategy (introduced in the next section) to obtain a restored load "word" sequence. Finally, the restored load value is generated by decoding the restored "word", which is the inverse process of encoding.

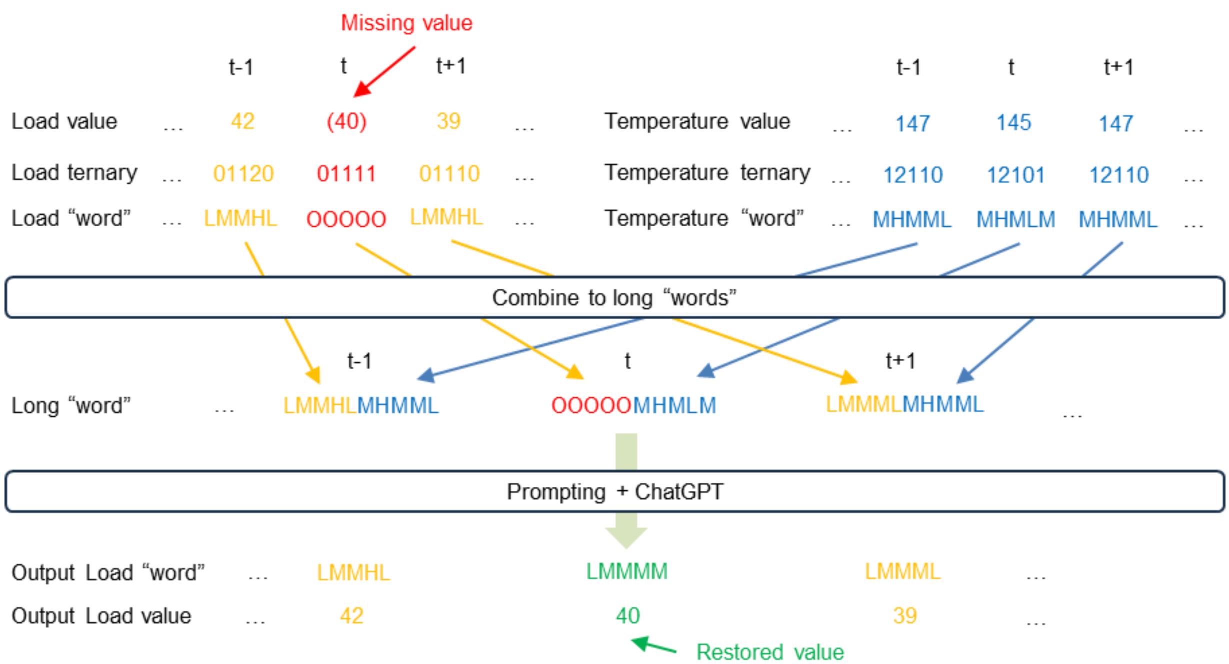


Figure 4.4 An illustration of load-temperature to "word" encoding method.

In the simulation section, we will discuss the performance across different scenarios: "encoded" versus "not-encoded", and "load-temperature-as-a-single-word" versus "load-temperature-as-separate-words".

3) Prompt Engineering

Prompt engineering is a methodology employed in the field of natural language processing (NLP) to design effective prompts for language models like GPT-3.5. It involves crafting specific instructions or queries that guide the model to generate desired outputs or responses. This process requires understanding the capabilities and limitations of the model, as well as the nuances of the task or domain for which prompts are being created. Effective prompt engineering can significantly improve the performance and applicability of language models across various tasks, including missing data restoration, demand response baseline estimation, and more.

In the power system demand response baseline estimation task, as shown in Figure 4.5 (a), we structure the prompt in a multi-turn chat format. The conversation follows the format defined by OpenAI [104], consisting of a list of messages where each message has a role and content. The "user" is the person interacting with ChatGPT, and the "assistant" represents ChatGPT. Initially, the user informs the assistant about the task and explains how the data is defined. Then, the user provides the encoded data as requested by the assistant. Finally, the assistant estimates the missing data and outputs the complete load encodings, which can be decoded to generate the complete load profile.

Combining load and temperature into a single long encoding is optional. We can also provide load and temperature encodings separately, as illustrated in Figure 4.5 (b). The comparison between these two prompting methods is discussed in the simulation results section.

A multi-turn chat, as shown in Figure 4.5, serves as a data sample for fine-tuning. A collection of such data samples constitutes a fine-tuning dataset, which is used to fine-tune ChatGPT. Once fine-tuning is complete, ChatGPT will be able to understand the inherent correlations between the missing parts and the rest of the load data, enabling it to perform missing data restoration and demand response baseline estimation. For testing purposes, the testing set is prepared in the same format, with the last message in each multi-turn chat removed.

```
{"messages":
    [{"role": "user",
      "content":"Given a load profile with missing segments and a complete"
                "daily temperature profile, estimate the missing portions of"
                "the load profile. The load and temperature data are provided"
                "jointly. Each value for load and temperature is encoded as a"
                "ten-digit word in ternary format. The first and last five"
                "digits represent load and temperature, respectively. Missing
                "load values are represented by 00000. Please provide the"
                "estimated load profile in the same format and length."},
     {"role": "assistant",
      "content": "What's the encoded data?"},
     {"role": "user",
      "content": long_encodings},
     {"role": "assistant",
      "content": completion}
    ]
}
```

(a)

```
{"messages":
    [{"role": "user",
      "content":"Given a load profile with missing segments and a complete"
                "daily temperature profile, estimate the missing portions of"
                "the load profile. The load and temperature data are provided"
                "separately and exhibit a correlation. Each value for load or"
                "temperature is encoded as a five-digit word in ternary format."
                "Missing load values are represented by 00000. Please provide"
                "the estimated load profile in the same format and length."},
     {"role": "assistant",
      "content": "What's the encoded load profile?"},
     {"role": "user",
      "content": load_encodings},
     {"role": "assistant",
      "content": "What's the encoded temperature?"},
     {"role": "user",
      "content": temperature_encodings},
     {"role": "assistant",
      "content": completion}
    ]
}
```

(b)

Figure 4.5 Prompting methods used in missing data restoration. (a) load-temperature-as-a-single-word. (b) load-temperature-as-separate-words.

4) OpenAI Fine Tuning

Fine-tuning is a powerful technique because it leverages the knowledge and general language understanding already present in the pre-trained model, allowing for efficient adaptation to new tasks or domains with relatively small amounts of additional data. It has been widely used to customize language models for a wide range of applications.

Fine-tuning in the context of OpenAI's language models involves adjusting a pre-trained model on a specific dataset or task to improve its performance for that particular use case. When fine-tuned, the model's parameters are updated based on new data, allowing it to adapt to the specifics of the task or domain.

For example, with GPT-3.5, fine-tuning involves providing additional training data related to a specific task, such as missing data restoration or demand response baseline estimation, and then adjusting the model's parameters through further training on this data. This process helps the model learn task-specific patterns and nuances, ultimately enhancing its ability to perform well on the given task.

To create a fine-tuning job with OpenAI, we first upload the fine-tuning dataset created in the previous section, then initiate the fine-tuning job through the Fine-tuning User Interface [105] or programmatically. The OpenAI platform will then fine-tune the specified model using the provided dataset and generate the fine-tuned model. In our fine-tuning job, we experimented with different numbers of samples in the dataset. These data sizes are relatively small compared to those used for training a traditional machine learning model from scratch. The performance evaluation of different fine-tuning data sizes will be presented in the simulation results section.

4.2.4 2nd Stage Fine-tuning

In the second stage, we use an even smaller amount of data from the target user, whose pattern is similar to the 10 users used in the first stage, to further fine-tune the model obtained from the first stage.

Fine-tuning is a powerful technique because it leverages the knowledge and general language understanding already present in the pre-trained model, enabling efficient adaptation to new tasks or domains with relatively small amounts of additional data. The first stage fine-tunes ChatGPT from a general language model to a specific missing data restoration model for a group of users. The second stage then adapts this model for the target user, effectively narrowing the gap between a general model and a highly specialized model, making the fine-tuning process smoother. If we were to fine-tune ChatGPT directly with data from the target user, the limited data availability would prevent the model from performing well.

The data preprocessing, encoding, and prompt engineering techniques used in this stage are the same as those in the first stage. We also experimented with different numbers of samples for the second stage fine-tuning, resulting in varying levels of missing data restoration accuracy. Additionally, we attempted to fine-tune ChatGPT directly with data from the target user. The results of these experiments will be presented in the simulation results section.

4.2.5 Evaluation Metrics

The performance metrics used for evaluating the accuracy of the restored data segments are calculated as

$$MPE = \frac{1}{K}\sum_{t=1}^{K}\frac{|\hat{x}_t^m - x_t^m|}{x_t^{event}} \tag{4.5}$$

$$RMSE = \sqrt{\frac{1}{K}\sum_{t=1}^{K}(\hat{x}_t^m - x_t^m)^2} \tag{4.6}$$

$$EGYE = \frac{|\sum_{t=1}^{K}\hat{x}_t^m - \sum_{t=1}^{K}x_t^m|}{\sum_{t=1}^{K}x_t^{event}} \tag{4.7}$$

where K is the total number of data points in the MDS, and $\hat{x}$ represents the restored data segment. These indices offer insights into different aspects of errors, including point-to-point errors and discrepancies in total energy consumption.

4.3 Simulation Results

In this section, we assess the ability of the fine-tuned large language model to recover missing data segments (MDS) under different scenarios and compare it with the benchmark method BERT-PIN [87]. The evaluation encompasses diverse fine-tuning scenarios and variations in training data volumes, providing valuable insights into fine-tuning Large Language Models for tasks in power system data analytics. Lastly, we discuss applying the proposed method to other power system data analysis tasks, such as load forecasting.

4.3.1 Data Preparation

The load profiles used in this study consists of 15-minute resolution smart meter data obtained from 11 users in North Carolina in summer 2018. The corresponding temperature data is downloaded from the National Oceanic and Atmospheric Administration (NOAA) [59] website and is used as a second modality input.

These load profiles are aligned with the temperature profile and then normalized based on the overall load and temperature peaks, respectively. Then, we partition the profiles into daily (96 data points) profiles. Each missing load data segment is consistently set at 4 hours (16 data points), a choice guided by the observation that around 70% of missing load data segments in actual utility data are less than 4 hours in duration. It's important to note that there are no missing data segments in the temperature profile. The dataset is divided into an 80% training set and a 20% testing set.

4.3.2 Evaluation of 1st Stage Fine-tuning

In the first stage of fine-tuning, the standard GPT-3.5 model is fine-tuned using data collected from 10 users. These users are evenly distributed between the fine-tuning and testing datasets. Subsequently, the fine-tuned GPT model is utilized to restore a 4-hour missing window in a daily load profile. BERT-PIN [87] is chosen as the benchmark method due to its superior performance in power system missing data restoration compared to other existing methods. Figure 4.6 presents several restored daily load profiles. Upon visual inspection, it is evident that the fine-tuned GPT model (represented by the red lines) achieves comparable performance with BERT-PIN (represented by the blue lines). The missing data segments restored by both methods closely resemble the ground truth.

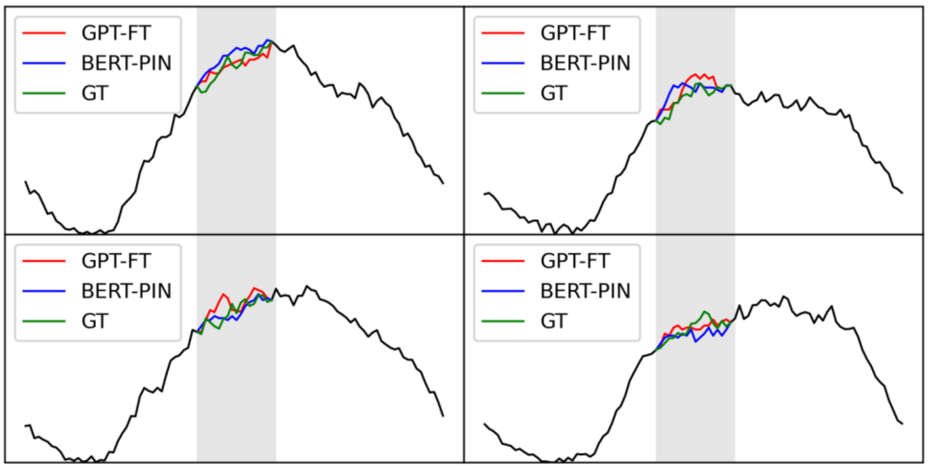


Figure 4.6 Examples of missing data restoration by fine-tuned model and benchmark model.

To quantitatively evaluate the accuracy of the fine-tuned model, we explored various scenarios. Before delving into these scenarios, it's important to clarify some concepts. A sample refers to a multi-turn chat message depicted in Figure 4.6, representing one daily load profile. "Advanced prompt" is a prompt method introduced in section 4.2.3, while "non-advanced prompt" implies the absence of detailed explanation information in the message, making it challenging for GPT to comprehend the encoding strategy and data features. "Discard encoding" refers to replacing load and temperature encodings with integers. "Abnormal days" are those when there is a sudden drop in temperature, which could affect the quality of the fine-tuning data, particularly during summer.

The experiment comprises seven scenarios:

Scenario 1: Fine-tune with 128 samples, using a non-advanced prompt, and combining load data and temperature data in the input. The encoding method is employed, and abnormal days are not removed.

Scenario 2: Similar to scenario 1, but with the number of samples increased to 256.

Scenario 3: Similar to scenario 2, but with the number of samples further increased to 512.

Scenario 4: Similar to scenario 3, but adopting the advanced prompt method introduced in section 4.2.3.

Scenario 5: Similar to scenario 4, but with load encoding and temperature encoding separated in the input, as shown in Figure 4.5 (b).

Scenario 6: Similar to scenario 5, but removing the encoding part, and directly using integers representing load and temperature values as input.

Scenario 7: Similar to scenario 6, but removing abnormal days from the fine-tuning dataset.

Based on the results presented in Table 4-2, the following observations were made:

- Across scenarios 1-3, using more samples in the first stage fine-tuning leads to higher accuracy. Increased data coverage enables the fine-tuned model to better capture various load patterns, thus enhancing its capability in missing data restoration. The selection of 512 samples was based on data availability, with each user contributing about 50 samples from the 86 summer days, leaving the remainder for testing purposes. Thus, 512 is selected as the number of samples in the following scenarios.

- Scenario 4, which employs an advanced prompt, yields lower errors. The detailed information about the task description, encoding strategy, and output constraints provided in the advanced prompt aids GPT in understanding the correlations between load and temperature as well as contextual information within load sequences, thereby improving the model's performance.

- Scenario 5, where the load encoding and temperature encoding are presented in a two-turn chat, instead of in one single long encoding, simplifies the encoding format. Because GPT model does not need to identify which part of the long encoding represents load and which part represents temperature. Therefore, scenario 5 further decreases the restoration error.

- Inspired by the success of scenario 5, scenario 6 removes the encoding mechanism entirely, replacing load and temperature encodings with normalized integers. This results in further performance improvement, indicating that ChatGPT can comprehend direct numbers better than encoded "words". Because these “words” have not appeared in ChatGPT’s training dataset, it’s hard to make ChatGPT understand it by fine-tuning with a small amount of data.

- In scenario 7, abnormal days with sudden temperature drops are removed from the fine-tuning dataset. This mitigates the adverse effect of temperature drops on power consumption, which makes the model's parameters be updated to another direction. Consequently, the removal of abnormal days leads to a further increase in missing data restoration accuracy.

- Despite the decreasing errors observed from scenario 1 to scenario 7, the performance remains slightly below that of the benchmark model BERT-PIN [87]. BERT-PIN is a BERT-based model that has been trained from scratch using a significantly larger dataset, comprising approximately 220,000 samples collected from 8,000 users over three years, with a training duration of 6-8 hours. In contrast, the proposed fine-tuned model only requires 512 samples collected from 10 users over three months, with a fine-tuning time less than 1 hour. Despite using fewer resources, the fine-tuned model achieves acceptable performance. Moreover, further performance improvements are expected in the second stage of fine-tuning.

Table 4-2 Missing data restoration in different scenarios

	# of samples	Advanced prompt	Separate Load & temperature	Discard Encoding	Remove abnormal days	Errors (%)		
						MPE	RMSE	EGYE
Scenario 1	128	N	N	N	N	5.609	4.513	4.303
Scenario 2	256	N	N	N	N	4.609	3.816	3.652
Scenario 3	512	N	N	N	N	4.59	3.795	3.547
Scenario 4	512	Y	N	N	N	3.806	3.027	2.987
Scenario 5	512	Y	Y	N	N	3.266	2.656	2.469
Scenario 6	512	Y	Y	Y	N	2.48	2.029	1.639
Scenario 7	512	Y	Y	Y	Y	**2.221**	**1.977**	**1.443**
BERT-PIN	~220K	-	-	-	-	1.612	0.699	0.887

4.3.3 Evaluation of 2nd Stage Fine-tuning

In the first stage, GPT-3.5 undergoes fine-tuning with data collected from 10 users (user0 – user9), resulting in the generation of the fine-tuned model GPT-FT-1. Subsequently, in the second stage, GPT-FT-1 undergoes further fine-tuning with data collected from a target user, distinct from the users utilized in the first stage (user10 - user19). It's important to note that only one target user is employed in the second stage fine-tuning, with the experiment being repeated using 10 different target users to assess the reliability of the fine-tuning strategy.

In the second stage fine-tuning, varying numbers of samples ranging from 10 to 50 are utilized to investigate the requisite sample size. This exploration is particularly advantageous in scenarios where suitable training data is scarce. Additionally, the performance of the GPT-FT-1 model without the second stage fine-tuning is evaluated on the target user to demonstrate the transferability of the first stage fine-tuned model. Furthermore, attempts are made to fine-tune the original GPT-3.5 model solely with data from the target user. In this scenario, 68 samples from the target user are utilized for fine-tuning, with the remaining 18 samples reserved for testing purposes. The setup of this scenario, depicted in Figure 4.7, serves to illustrate the necessity of the proposed two-stage fine-tuning strategy, particularly in cases where suitable training data is scarce.

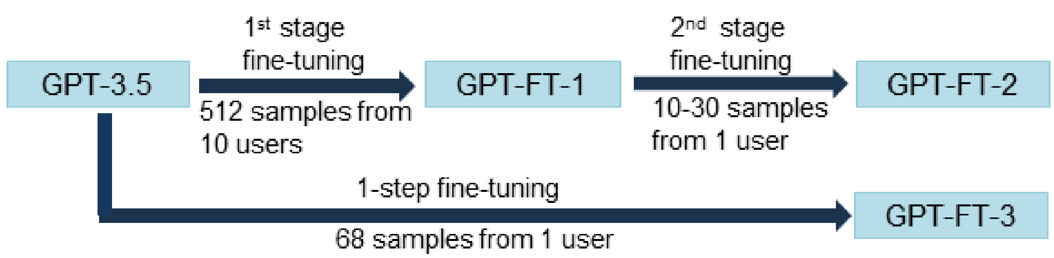


Figure 4.7 An illustration of fine-tuning the original GPT-3.5 model solely with data from the target user.

Table 4-3 Errors of Second Stage Fine-Tuning

Target Users	Errors (%)	GPT-FT-2 fine-tuned with different numbers of samples					GPT-FT-1 (W/O 2nd stage)		GPT-FT-3 (W/O 1st stage)
		10	20	30	40	50	Test on target user	Test on user0-9	
User10	MPE	2.245	2.277	2.428	2.395	2.361	2.497	2.221	2.983
	RMSE	1.988	1.969	2.014	2.094	2.140	2.182	1.977	2.772
	EGYE	1.501	1.513	1.599	1.604	1.637	1.843	1.443	2.461
User11	MPE	2.206	2.168	2.416	2.352	2.549	2.77	2.221	3.216
	RMSE	1.911	1.887	1.957	2.008	2.164	2.438	1.977	2.837
	EGYE	1.499	1.453	1.534	1.649	1.692	2.103	1.443	2.524

User12	MPE	2.276	2.453	2.483	2.533	2.676	2.441	2.221	3.068
	RMSE	1.962	1.997	2.016	2.147	2.163	2.206	1.977	2.649
	EGYE	1.524	1.517	1.567	1.599	1.643	1.784	1.443	2.581
User13	MPE	2.382	2.448	2.565	2.604	2.412	2.503	2.221	2.979
	RMSE	2.009	2.018	2.097	2.105	2.166	2.263	1.977	2.729
	EGYE	1.587	1.589	1.535	1.673	1.749	1.958	1.443	2.441
User14	MPE	2.318	2.248	2.558	2.48	2.336	2.673	2.221	3.110
	RMSE	1.997	1.948	2.021	2.134	2.196	2.273	1.977	2.811
	EGYE	1.472	1.499	1.514	1.731	1.887	2.094	1.443	2.687
User15	MPE	2.314	2.273	1.999	2.088	2.116	2.528	2.221	2.914
	RMSE	1.964	1.897	1.846	1.943	2.155	2.364	1.977	2.761
	EGYE	1.507	1.414	1.634	1.697	1.805	1.833	1.443	2.416
User16	MPE	2.186	2.302	2.418	2.336	2.237	2.767	2.221	3.131
	RMSE	1.885	1.934	2.149	2.261	2.381	2.419	1.977	2.828
	EGYE	1.455	1.393	1.620	1.678	1.803	1.972	1.443	2.338
User17	MPE	2.453	2.323	2.681	2.698	2.536	2.625	2.221	3.401
	RMSE	2.012	2.244	2.198	2.265	2.337	2.397	1.977	2.945
	EGYE	1.572	1.461	1.628	1.724	1.796	2.007	1.443	2.367
User18	MPE	2.466	2.449	3.019	2.466	2.777	2.92	2.221	3.274
	RMSE	2.194	2.130	2.248	2.555	2.730	2.615	1.977	2.716
	EGYE	1.419	1.527	1.699	1.647	1.933	2.118	1.443	2.592
User19	MPE	2.693	2.598	2.474	2.662	2.856	2.644	2.221	3.392
	RMSE	2.248	2.267	2.101	2.377	2.409	2.529	1.977	2.779
	EGYE	1.623	1.584	1.770	1.764	2.008	2.177	1.443	2.486

Based on the results presented in Table 4-3, the following observations were made:

- Contrary to common intuition, the performance of the second stage fine-tuned model does not improve with an increase in training data. Instead, optimal accuracy is achieved with 10 to 30 fine-tuning samples, while adding more samples (40 or 50) leads to decreased performance. This suggests that the GPT-FT-1 model, following the first stage fine-tuning, has grasped the underlying principles of the missing data restoration task in general user cases. Consequently, only a small number of samples are required for effective transfer of knowledge from general users to a specific target user, with additional data being redundant and detrimental to accuracy.
- Testing GPT-FT-1 on the target user yields increased errors compared to testing it on users0-9. This discrepancy arises because GPT-FT-1 is fine-tuned with data from users0-9, resulting in

better performance for these users. However, even when tested on data from a new user, GPT-FT-1 still achieves acceptable accuracy, indicating that the first stage fine-tuning endows GPT-FT-1 with a general understanding of the missing data restoration problem, thereby possessing potential for further fine-tuning to enhance performance.

- Compared to testing GPT-FT-1 without the second stage fine-tuning on target user (second column from the right of Table 4-3), the second stage fine-tuned model GPT-FT-2 exhibits improved accuracy for all target users. This underscores the effectiveness of the second stage fine-tuning in tailoring the model to the target data, successfully transferring the general understanding of missing data restoration to the specific user context with minimal labeled examples.

- Direct fine-tuning of the original GPT-3.5 model with 68 samples from the target user results in inferior performance compared to all other cases. While increased training data would likely improve accuracy significantly, such an approach is impractical in scenarios of data scarcity. In contrast, the two-stage fine-tuning strategy addresses this challenge by leveraging data from other users over a shorter timeframe, circumventing the need for extensive data collection from the target user over a prolonged period.

4.3.4 Cost Analysis

The fine-tuning strategy proposed in this chapter is executed through the OpenAI API. We upload the prepared fine-tuning data onto the OpenAI platform and initiate a fine-tuning task. The platform then proceeds to fine-tune the specified model with the uploaded data, generating the fine-tuned model. The cost associated with this process varies depending on the number of samples utilized in the fine-tuning process.

As depicted in Figure 4.8 (a), employing more samples results in a higher number of tokens trained by the platform. According to pricing information sourced from OpenAI [107], the cost amounts to $8 per 1M tokens trained. Consequently, the training cost for different sample sizes is illustrated in Figure 4.8 (b). For instance, fine-tuning with a dataset comprising 512 samples would incur approximately $13 in training costs. It's important to note that this cost solely pertains to the fine-tuning process; testing costs are calculated separately but are relatively lower compared to training expenses, hence not depicted in Figure 4.8.

One of the key advantages of utilizing fine-tuning is its time efficiency. Fine-tuning 512 samples takes merely 1 hour to achieve acceptable results. This stands in stark contrast to the hours required to train a specific model from scratch. Therefore, fine-tuning represents a trade-off between marginal performance improvements and computational resource consumption.

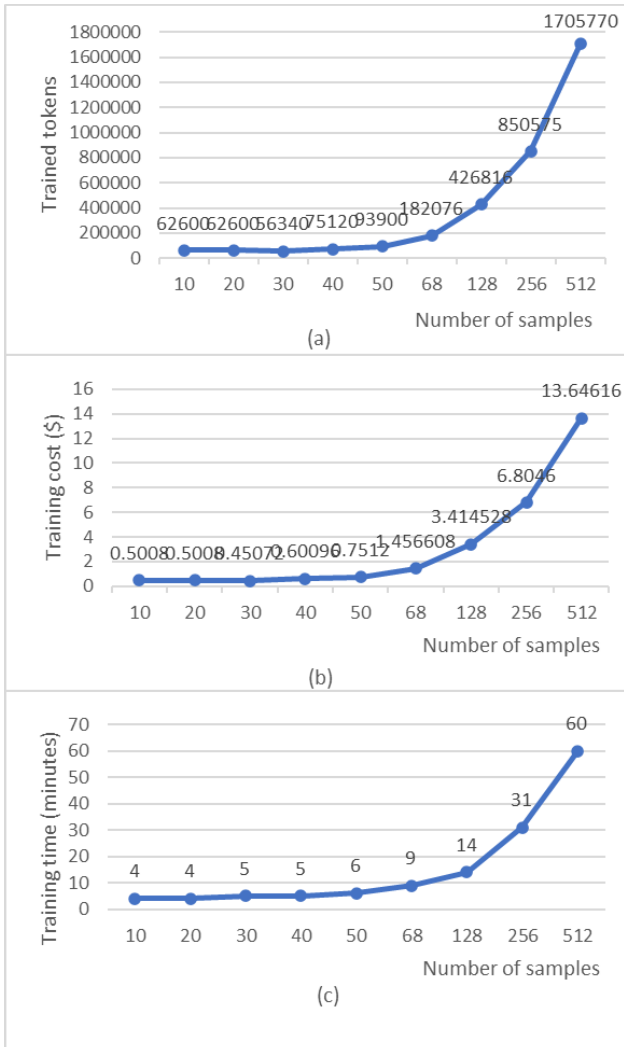


Figure 4.8 The cost of fine-tuning versus number of samples.

4.4 Conclusion

In conclusion, this chapter introduces a novel approach using fine-tuned LLMs for load profile analysis, exemplified by the missing data restoration task. We adapted the pre-trained GPT-3.5 model for power system load analytics through a two-stage fine-tuning process. Our empirical results demonstrate that the fine-tuned model restores missing data with accuracy comparable to advanced models like BERT-PIN.

Key insights include the effectiveness of using a minimal number of fine-tuning samples, which underscores the efficiency of few-shot learning. Additionally, advanced prompt engineering and separate encoding of load and temperature data significantly enhance model performance. The fine-tuning strategy proved to be cost-effective and time-efficient, presenting a viable alternative to training models from scratch, especially in data-limited and resource-scarce settings.

Future work from this research will focus on extending the application of LLMs to additional power system load analysis tasks, including load forecasting, load disaggregation, customer segmentation, and synthetic data generation.

5. GENERATIVE AI-BASED POWER SYSTEM EVENT DETECTION USING SYNCHROPHASOR DATA

5.1 Background

With the growing penetration of renewable energy, increasing system complexity, and tighter operating margins, the need for real-time, accurate, and robust detection of power system disturbances has become critical. Synchrophasor data, collected via Phasor Measurement Units (PMUs), offers high-resolution, time-synchronized measurements that are ideal for monitoring system dynamics and enabling early warning of abnormal events. However, effectively analyzing such data at scale remains a significant challenge.

Existing approaches to PMU event detection can be broadly categorized into signal processing-based and data-driven methods. Traditional signal processing techniques, such as wavelet transforms [108], short-time Fourier transforms [109], and principal component analysis (PCA) [110], detect events by examining frequency-domain components or low-rank subspace deviations in raw PMU data. Other methods employ correlation coefficient matrices [111] or singular value decomposition (SVD) [112] to capture spatial correlations and structural changes. While effective in some settings, these rule-based methods typically depend on manually defined features, thresholds, and fixed-time windows, which limit their adaptability and accuracy for detecting diverse event types.

To improve generalization, supervised learning models have been introduced, including decision trees [113], convolutional neural networks (CNNs) [114][115][116][117], and hybrid models such as autoencoders with LSTMs [118][119]. These approaches require both positive and negative labeled samples, which can be difficult to obtain in realistic grid environments, particularly for rare or emerging event types. Furthermore, many supervised models rely on extracted features rather than directly utilizing raw PMU signals, potentially losing critical temporal and spatial resolution. The recent use of GANs for PMU event detection [120] marks a shift toward unsupervised generative modeling, but conventional GANs face limitations in training stability and latent space interpretability.

To address these limitations, this chapter proposes a novel event detection framework based on Variational Autoencoder–Generative Adversarial Networks (VAE-GAN) using raw PMU data. Our method provides several key advantages:

- No manual feature definition: The model learns directly from raw multivariate PMU waveforms.
- Unsupervised learning: Only normal (non-event) data is needed for training, improving scalability.
- Detection of unknown event types: The model generalizes to event types not seen during training.
- Flexible temporal-spatial modeling: It can detect long-duration and spatially distributed events across multiple PMUs, overcoming the rigidity of fixed-window models.

The proposed architecture is evaluated using real-world synchrophasor data across a wide range of event types, including line trips, faults, and oscillations. Experimental results demonstrate the superiority of our model over baselines in terms of the capability to identify a wide range of power system events, including those that are often missed by conventional methods.

The remainder of the chapter is organized as follows: Section 5.2 provides a detailed overview of the proposed method. Section 5.3 illustrates the simulation results on event detection. Finally, section 5.4 concludes the chapter.

5.2 VAE-GAN Based Event Detection Method

This chapter leverages the representational power of VAEs to learn a compact latent space of normal operating conditions in PMU time series and utilizes a GAN-style discriminator to enhance the model’s ability to distinguish between real and synthetic (reconstructed) PMU data. Additionally, the inclusion of a sliding window–based temporal segmentation mechanism enables the framework to monitor extended periods of time and detect evolving or long-lasting events that may not be evident from instantaneous or short-window analyses.

5.2.1 Signal Selection and Data Pre-processing

In the synchrophasor data, there are many signals that are available for event detection. After extensive testing with field recorded synchrophasor data, inventors found that the following signals provide the best detection performance:

- Voltage magnitude at phase A
- Voltage magnitude at phase B
- Voltage magnitude at phase C
- Positive sequence current magnitude
- Frequency
- Rate-Of-Change-Of-Frequency (ROCOF)

Then, the selected signals for the n^{th} PMU are normalized by

$$v_{pu}^{n} = \frac{v^{n}}{v_{rate}^{n}} \tag{5.1}$$

$$v_{norm}^{n} = v_{pu}^{n} - mean\left(v_{pu}^{n}\right) \tag{5.2}$$

$$i_{norm}^{n} = i_{m}^{n} - mean(i_{m}^{n}) \tag{5.3}$$

$$f_{norm}^{n} = f^{n} - mean(f^{n}) \tag{5.4}$$

Where v^{n} is the phase voltage magnitude, v_{rate}^{n} is the rated voltage at the measurement point, i_{m}^{n} is the positive sequence current magnitude, and f^{n} is the frequency. In this way, the normalized signals values are distributed around zero no matter the original voltage or current levels.

5.2.2 Event Detection via VAE-GAN

This study utilizes a deep learning–based approach to detect power system events by modeling normal system behavior with a Variational Autoencoder–Generative Adversarial Network (VAE-GAN). As illustrated in Figure 5.1, the system consists of three major components: Encoder, Decoder/Generator, and Discriminator.

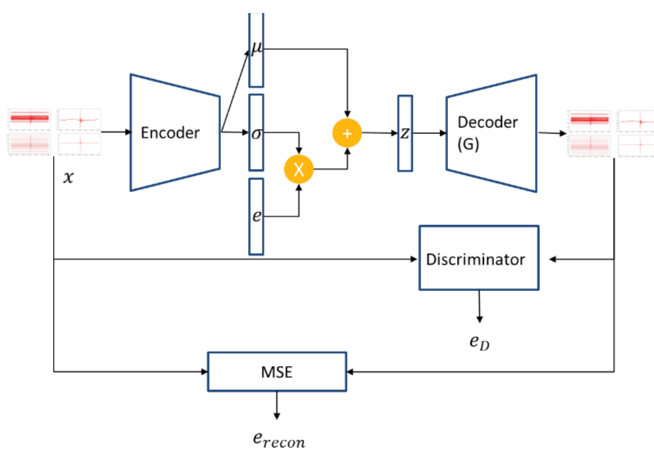
The encoder network receives the normalized PMU measurements at a 5-second length. It outputs two vectors: mean vector μ, and standard deviation vector σ. A latent vector z is then sampled using the reparameterization trick:

$$z = \mu + \sigma \cdot e, \ where\ e \sim \mathcal{N}(0,1) \tag{5.5}$$

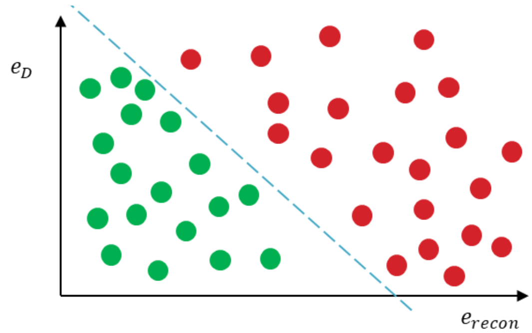
The Decoder/Generator reconstructs a signal $G(z)$ from the latent representation. This reconstruction is compared to the original input to detect anomalies.

The Discriminator network receives both the original PMU signal and the reconstructed signal. It is trained to distinguish the difference between real and reconstructed samples, thereby enhancing the generator's realism.

Under normal operating conditions, the model learns to accurately reconstruct time-series data from phasor measurement units (PMUs), resulting in consistently low reconstruction and discrimination errors. However, when an abnormal event occurs—such as a fault, oscillation, or disturbance—the input pattern deviates from the learned distribution of normal behavior. As a result, the model's reconstruction becomes less accurate, and the discriminator identifies the difference more sharply, leading to a significant increase in error metrics. These deviations are used to flag and detect power system events in real time, providing a data-driven, unsupervised framework for enhancing grid situational awareness and resiliency.



(a) VAE-GAN Architecture.



(a) 2D feature space illustration.

Figure 5.1 VAE-GAN for PMU event detection

5.2.3 Loss Function Designation

Reconstruction Loss that measures the difference between original and reconstructed signals using a hybrid of MSE and a max-penalty term:

$$\mathcal{L}_{recon} = \frac{1}{N}\sum_{i=1}^{N}\|x_i - G(z)\|^2 + \max\left(\|x_i - G(z)\|\right) \tag{5.6}$$

KL Divergence Loss that regularizes the latent space:

$$\mathcal{L}_{KL} = -\frac{1}{2}\sum(1 + \log(\sigma^2) - \mu^2 - \sigma^2) \tag{5.7}$$

Adversarial Loss that encourages realism in the reconstructed output:

$$\mathcal{L}_{adv} = -\mathbb{E}[\log D(G(z))] \tag{5.8}$$

The total loss for training the VAE-GAN model is:

$$\mathcal{L}_{VAE-GAN} = \mathcal{L}_{recon} + \lambda_1\mathcal{L}_{KL} + \lambda_2\mathcal{L}_{adv} \tag{5.9}$$

Where λ_1 and λ_2 are scaling coefficients.

The discriminator D itself is trained using:

$$\mathcal{L}_D = -\mathbb{E}[log\, D(x)] - \mathbb{E}[log(1 - D(G(z))] \tag{10}$$

5.2.4 Decision Metrics

Each sample is evaluated by two errors: reconstruction error e_{recon} and discriminator output error e_D. Those two values are jointly used to score the time window. As shown in Figure 5.1, events are expected to deviate from the normal data cluster in a 2D feature space defined by e_{recon} and e_D, enabling both threshold-based and classifier-based detection mechanisms.

An event is detected if the metrics meet the following criteria:

$$\lambda_{recon} \cdot e_{recon} + \lambda_D \cdot e_D > \eta_1 \tag{5.11}$$

Where η_1 is the threshold for event detection, λ_{recon} and λ_D are the weights for the two errors, respectively.

5.2.5 Sliding Window-based Processing

A hierarchical sliding window mechanism is incorporated to enable continuous and long-term monitoring of power system events using phasor measurement unit (PMU) data streams. This mechanism is essential to the temporal structure of the method, allowing it to detect both short-duration transient and slow-developing system events over extended periods.

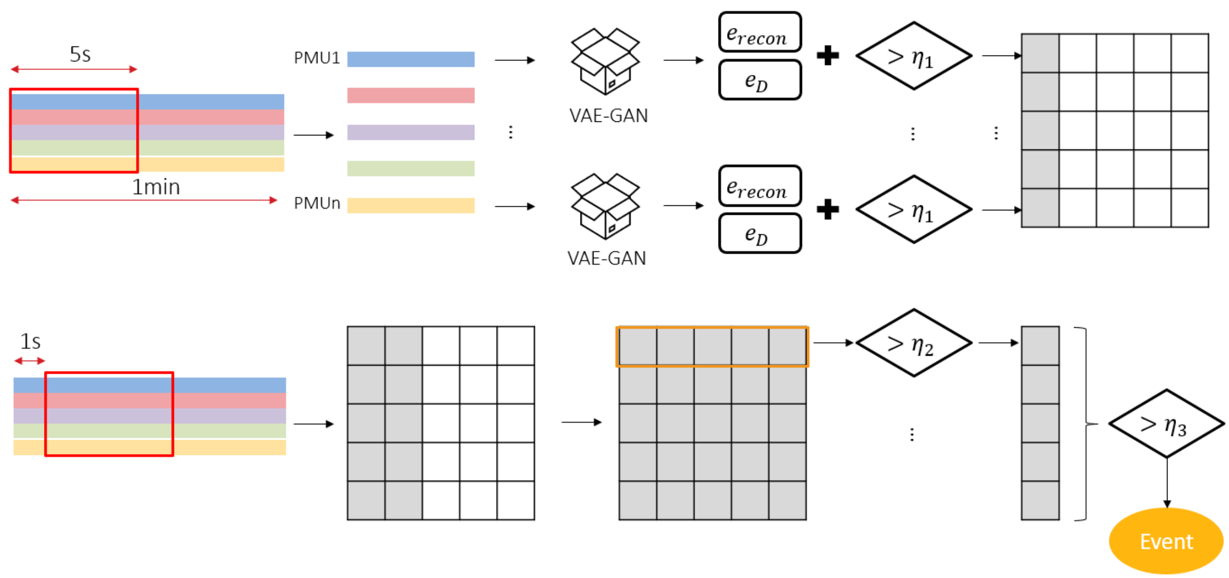


Figure 5.2 A two-level sliding window mechanism

As shown in Figure 5.2, the mechanism operates on two levels:

Primary Windowing – VAE-GAN Inference per PMU: A larger time window (e.g. 1 minute) of PMU data is segmented into smaller overlapping windows (e.g. 5-second segments with 1-second steps). Then, each short segment is independently processed by the VAE-GAN model to generate reconstruction error and discriminator error. Third, the combined error is compared against a threshold η_1. If it exceeds this threshold, the segment is flagged as abnormal for that PMU stream. This process is repeated across multiple PMU channels, forming a 2D binary anomaly matrix representing spatial-temporal detection flags for each segment and PMU.

Secondary Windowing – Spatiotemporal Aggregation: A second level sliding window (e.g., 1-second steps across the matrix) scans the binary detection matrix generated above. Within each time-aligned slice (i.e., row), the maximum number of consecutive abnormal flags is calculated. If the result exceeds the secondary threshold η_2, that time slice is flagged as suspicious. This results in a sequence of binary indicators across PMUs. A final event detection decision is made by applying a third threshold η_3 on the accumulated event flags, allowing for confirmation of persistent or impactful events.

This multi-resolution sliding window framework is not simply a pre-processing step – it is an integral part of the framework's ability to detect events that unfold gradually across PMUs, capture temporal continuity in disturbances, and reduce false positives by incorporating spatiotemporal correlations. By structuring the input this way, the method ensured robust detection of both localized and system-wide anomalies in real-time PMU streams.

5.3 Simulation Results

This section presents the simulation results evaluating the performance of the proposed VAE-GAN-based event detection framework using real-world synchrophasor data. The model is tested against a diverse set of power system disturbances and benchmarked against the traditional Max Envelope method.

5.3.1 Model Training

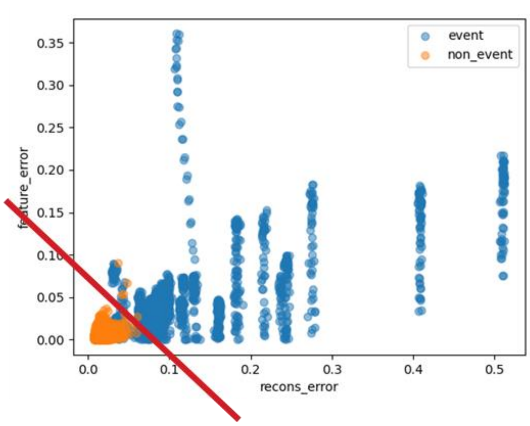
The VAE-GAN model was trained to capture the normal operating behavior of power system dynamics using historical phasor measurement unit (PMU) data. Training was conducted in an unsupervised fashion, where only normal (non-event) time-series segments were used to establish the latent distribution and reconstruction capabilities of the model. This approach enables the model to identify deviations associated with abnormal conditions without relying on labeled event data. Each PMU signal was first segmented into overlapping windows of fixed length (e.g., 5 seconds) with a defined stride (e.g., 1 second). These windowed segments served as input samples for training. The model was implemented using PyTorch and trained using the Adam optimizer with a learning rate of 2×10^{-4}. The batch size was set to 128, and early stopping was employed based on reconstruction loss on a validation set. To improve generalization, batch normalization was applied during training. Upon convergence, the model demonstrated the ability to reconstruct normal PMU patterns with low error while producing higher reconstruction and adversarial scores on eventful or abnormal sequences. This discrepancy forms the basis for effective anomaly detection in downstream applications.

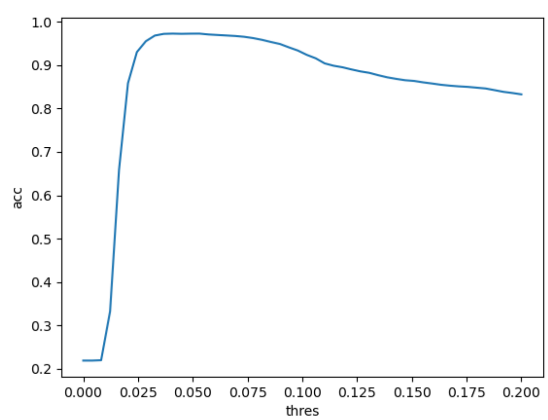
High speed event detection is the key when dealing with massive amount of synchrophasor data. The scalability of the algorithm is tested on a laptop PC. It only takes around 5 seconds for the proposed detection to scan through a 1-minute synchrophasor dataset and create a list of events. It shows that the proposed detection method is able to detect events in a real-time manner.

5.3.2 Event Detection on 5-second Window for Single PMU

To validate the effectiveness of the proposed VAE-GAN-based event detection framework, a series of detection experiments were conducted using labeled phasor measurement unit (PMU) time-series data containing both normal and abnormal system conditions. Each data segment was processed using a 5-second sliding window, and the resulting outputs were used to evaluate the reconstruction error e_{recon} and discriminator error e_D, which were then combined to compute an overall event detection score.

Figure 5.3 (a) illustrates the distribution of individual PMU data segments in the 2D feature space defined by the reconstruction error and discriminator error. Blue dots correspond to data segments associated with known power system events, while orange dots represent non-event (normal) segments. As shown in the plot, event-related segments tend to occupy regions with higher values of either or both error metrics. In contrast, non-event samples cluster tightly in the lower-left quadrant of the error space, indicating high-fidelity reconstruction and low discriminator rejection.





(a) 2D feature space and threshold. (b) Accuracy over a range of scoring thresholds.
Figure 5.3Event detection accuracy over a range of scoring thresholds.

A conceptual decision boundary is drawn to separate these two distributions, illustrating the discriminative capability of the learned scoring space. This visualization confirms that the joint error space provides a meaningful separation between normal and abnormal behavior, forming the basis for effective threshold-based detection.

To determine an optimal detection threshold, a range of values was evaluated for their impact on binary classification accuracy. Figure 5.3 (b) shows the variation of detection accuracy as a function of the event score threshold η_1. The accuracy increases rapidly as the threshold increases from 0 to approximately 0.025, reaching a peak of 97.3%. Beyond this point, accuracy gradually declines due to the increase in false negatives (i.e., undetected events).

These results demonstrate that the proposed VAE-GAN model, when paired with an appropriately tuned threshold, can effectively distinguish between event and non-event PMU behavior using only unsupervised training on normal data. The joint use of reconstruction and adversarial features enhances the model's sensitivity to subtle system changes while maintaining a low false-positive rate.

5.3.3 Event Detection on 1-minute Window for Multiple PMUs

Figure 5.4 demonstrates the ability of the proposed VAE-GAN framework to detect and localize a power system event using 1-minute phasor measurement unit (PMU) data. The left subplot shows normalized voltage magnitude signals from multiple PMU channels over a 60-second interval. While most signals remain steady, a sharp drop occurs around timestamp 01:09:36, clearly indicating the presence of an event. A smaller disturbance is also visible just before the main deviation, suggesting early anomalous behavior.

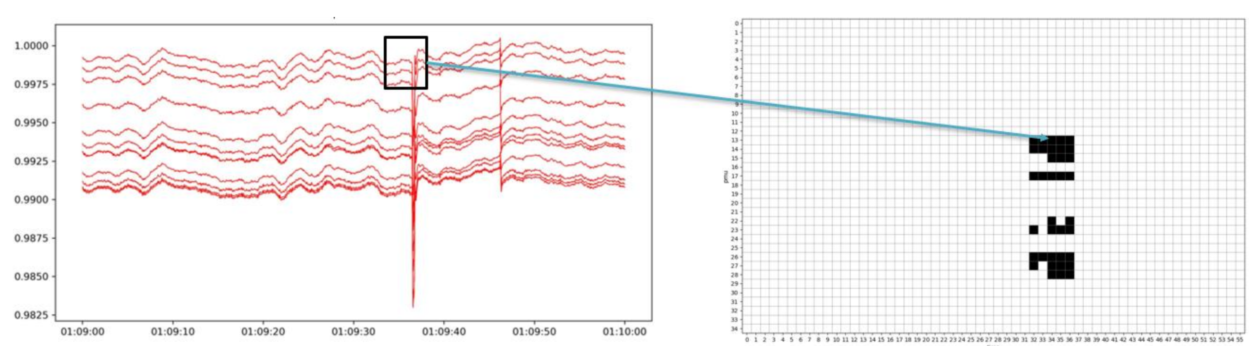


Figure 5.4 Illustration of binary anomaly matrix.

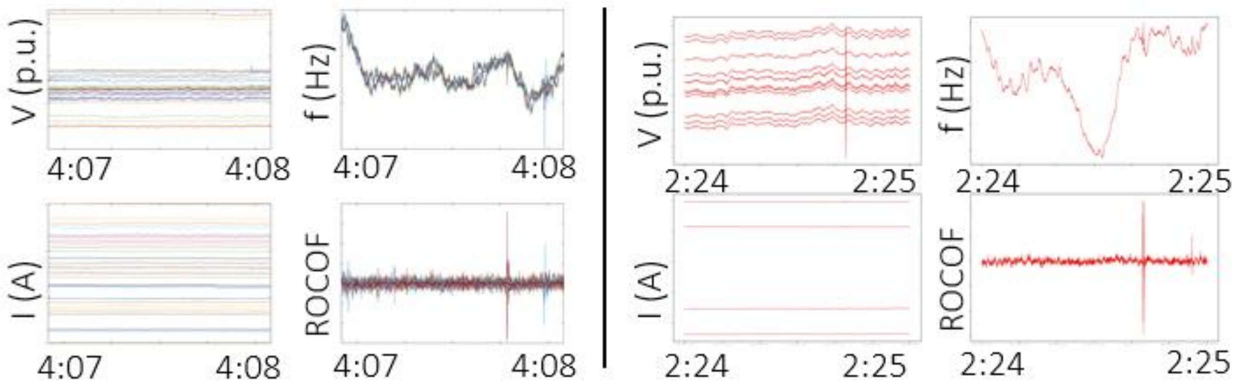
The right subplot in Figure 5.4 presents a spatiotemporal anomaly matrix generated by the detection framework. Each row corresponds to a specific PMU channel, and each column represents a time step derived from overlapping sliding windows. Black squares mark the segments where the joint event score—based on reconstruction and adversarial errors—exceeds a predefined threshold. The resulting matrix highlights a well-localized cluster of detections aligned with the event timing and the affected channels. The framework successfully identifies not only the core event but also the precursor variations, offering both spatial and temporal precision in event characterization. This confirms the effectiveness of the proposed method in delivering interpretable, data-driven insights for real-time power system monitoring.

5.3.4 Performance Comparison with Max Envelope Method

As a benchmark, the Max Envelope method [121] identifies disturbances by calculating the area between moving minimum and maximum bounds of power system signals. A larger envelope area typically indicates abnormal behavior. The envelope size is a critical hyperparameter; empirical studies suggest it should be approximately one order of magnitude greater than the event duration for optimal detection sensitivity.

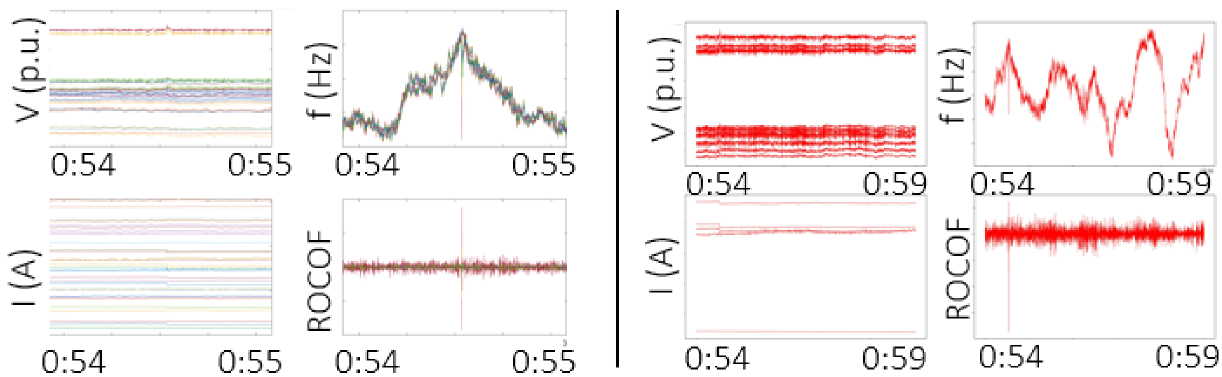
To illustrate the practical differences between the two methods, several representative figures are included. Figures 5.5 and 5.6 illustrate qualitative differences between the two methods. Figure 5.5 showcases events detected exclusively by one method: (a) depicts events identified solely by the Max Envelope method, typically characterized by short, spike-like changes in single channels, while (b) highlights events detected only by VAE-GAN, which tend to involve subtle, multichannel disturbances that evolve over a longer duration. These differences underscore the limitations of envelope-based techniques when encountering spatially distributed or low-amplitude anomalies.

Figure 5.6 shows examples of events detected by both methods but with varying durations. In these cases, the VAE-GAN model generally captures a broader temporal scope of the disturbance, while the Max Envelope method often restricts its detection to the peak or central portion. This difference demonstrates VAE-GAN's ability to recognize prolonged or slowly progressing events, which is crucial for accurately assessing power system stability in real time.



(a) Max Envelop (b)VAE-GAN

Figure 5.5 Example events detected by only one method.



(a) Max Envelop (b)VAE-GAN

Figure 5.6 Example event detected by both methods with different durations.

Table 5-1 summarizes a comparative evaluation between the proposed VAE-GAN-based event detection approach and the Max Envelope method across multiple criteria. The VAE-GAN model detected 420 events, significantly outperforming the Max Envelope approach, which detected 272 events. While the Max Envelope technique is effective for identifying spike-like events and single-channel patterns, it struggles with more complex event dynamics. Specifically, it often misses long-term, low-magnitude, or multi-channel patterns due to its dependence on fixed time windows and threshold heuristics.

In contrast, the VAE-GAN framework demonstrates broader capabilities. It successfully captures not only sharp disturbances but also gradual, spatially distributed events across multiple PMUs. The model's unsupervised architecture and ability to learn directly from raw PMU waveforms contribute to its generalization power and sensitivity to subtle anomalies. Although the Max Envelope method remains appealing for its simplicity and interpretability, its effectiveness is limited in more diverse and data-rich operational scenarios.

Table 5-1 Performance Comparison

Criteria	**Max Envelope**	**VAE-GAN**
Spike-like events	√	√
Long-term events		√
Pattern shown in one channel	√	
Simplicity and interpretability	√	
Subtle events		√
Generalization		√

5.4 Conclusion

This chapter presents a novel generative AI-based framework for power system event detection using a Variational Autoencoder–Generative Adversarial Network (VAE-GAN). Unlike traditional threshold-based or supervised approaches, the proposed method requires only normal data for training and effectively detects both known and previously unseen events directly from raw PMU waveforms. By modeling spatial-temporal dynamics without fixed time window assumptions, the architecture exhibits strong generalization to diverse event types, including long-duration disturbances.

Evaluations on real-world synchrophasor datasets demonstrate that the proposed VAE-GAN model outperforms the benchmark method, particularly in detecting a wider range of event types and showing enhanced robustness to subtle and long-duration disturbances. Additionally, the model's scalability makes it suitable for real-time monitoring applications in wide-area power systems. These findings indicate the potential of the VAE-GAN framework for real-time anomaly detection and to serve as a foundation for future research, including event localization and classification.

6. SUMMARY AND FUTURE WORK

This dissertation presented a suite of Generative AI-powered methods to address fundamental data challenges in power systems, namely, the scarcity, incompleteness, and event detection using synchrophasor data. These limitations have long hindered the adoption of advanced machine learning and data-driven tools across utility applications. Through the integration of generative adversarial networks (GANs), transformer-based models like BERT, fine-tuned large language models (LLMs), and deep learning classifiers, this research offers scalable and flexible solutions to enhance data quality and availability in the modern grid.

6.1 Summary of Contributions

The key contributions of this research are summarized as follows:

- **GAN-Based Synthetic Data Generation**

  The development of Generative Adversarial Networks (GANs) capable of generating synthetic load profiles that accurately capture spatial-temporal correlations across households and time periods. This contribution addresses the issue of limited data availability, providing high-quality synthetic datasets that mirror real-world data characteristics. The generated synthetic datasets can be safely shared across the community without privacy concerns.

- **BERT-Based Missing Load Data Restoration**

  The application of BERT-based approach for restoring missing load data in power systems. By leveraging the contextual understanding of BERT, the model successfully predicted missing values in load profiles with high accuracy. This work highlighted the importance of using language models for non-textual data and provided evidence that BERT can effectively restore missing information, improving data quality for downstream applications. This method can be used to demand response baseline estimation, which is critical for demand-side management.

- **Fine-Tuning Large Language Models (LLMs) for Missing Data Restoration**

  The exploration of fine-tuned LLMs for missing data restoration, showing that these models can achieve comparable accuracy to BERT-based methods while requiring significantly less training data and computational resources. This provides a flexible and efficient solution for data restoration in power systems.

- **Power System Event Detection Using PMU Data**

  A generative AI-based framework was developed to identify known and novel power system events with high accuracy and resilience. This work supports wide-area situational awareness and real-time monitoring in inverter-based, data-rich environments.

Overall, this research has contributed to the development of AI-powered solutions that enhance the reliability, efficiency, and resilience of power systems.

6.3 Future Work

While this dissertation has made significant strides in solving the problem of data analysis in power systems, several areas warrant further exploration and development. The following are suggestions for future research:

1. **Expansion of Synthetic Data Generation Techniques:**

   Future work could explore the integration of additional AI models, such as Variational Autoencoders (VAEs) or Diffusion Models, with GANs and Transformers to enhance the quality and diversity of synthetic load profiles. Investigating the applicability of these models to other types of power system data, such as voltage or frequency data, could also be beneficial.

2. **Contribute to Open-source Community and Standardize Synthetic Data Generation Performance Evaluation:**

   As the field of power system data analysis continues to grow, contributing to the open-source community and establishing standardized benchmarks for performance evaluation will be crucial for advancing research and ensuring consistency across studies. Future work in this area could focus on the following:

   a. **Open-source Tool Development:**

      Sharing the tools, models, and datasets developed during this research with the broader AI and power systems communities could accelerate progress and facilitate collaboration. By releasing code and documentation through open-source platforms like GitHub, other researchers and practitioners can replicate results, build upon existing models, and apply these techniques to their own datasets. Open-sourcing these resources will also promote transparency and encourage innovation, allowing the community to explore new approaches and improve upon current methods.

b. **Establishing Standardized Evaluation Metrics:**

The performance of synthetic data generation models is often evaluated using a variety of metrics, making it difficult to compare results across different studies. Future research could focus on developing and promoting standardized evaluation metrics that are widely accepted by the research community. These metrics should assess not only the accuracy and fidelity of the synthetic data but also its utility in downstream tasks such as load forecasting or demand response. Standardized benchmarks would enable more consistent comparisons between models and help identify the most effective approaches for specific applications.

c. **Benchmark Datasets for Synthetic Data Generation:**

Creating and maintaining publicly available benchmark datasets specifically designed for synthetic data generation in power systems would provide a valuable resource for researchers. These datasets could include real-world load profiles, voltage readings, and other relevant power system data, along with corresponding synthetic datasets generated by various models. By comparing the performance of different models on these standardized datasets, researchers can gain insights into the strengths and limitations of each approach, fostering the development of more robust and generalizable models.

d. **Community-driven Competitions and Challenges:**

Organizing competitions or challenges within the research community, focused on AI in power system, can stimulate innovation and drive the adoption of standardized practices. These events could encourage participants to develop new models, improve existing ones, and apply them to benchmark datasets, with results being openly shared and compared. Such initiatives would not only push the boundaries of current research but also help establish best practices for AI in power systems.

e. **Collaborative Research and Knowledge Sharing:**

Engaging with the broader AI and power systems communities through workshops, conferences, and collaborative research projects can facilitate the exchange of ideas and promote the adoption of standardized evaluation practices. By actively participating in these forums, researchers can contribute to the collective understanding of synthetic data generation, influence the development of industry standards, and ensure that future advancements are guided by a shared vision.

By contributing to the open-source community and working towards the standardization of synthetic data generation performance evaluation, future research can ensure that developments in this field are transparent, reproducible, and widely accessible. This approach will not only advance the state of the art but also foster a collaborative environment where innovations can be rapidly disseminated and adopted across the power systems domain and beyond.

REFERENCES

[1] J. Dickert and P. Schegner, "A time series probabilistic synthetic load curve model for residential customers," *2011 IEEE Trondheim PowerTech*, Trondheim, Norway, pp. 1-6, 2011, doi: 10.1109/PTC.2011.6019365.

[2] J. K. Gruber and M. Prodanovic, "Residential Energy Load Profile Generation Using a Probabilistic Approach," *2012 Sixth UKSim/AMSS European Symposium on Computer Modeling and Simulation*, Malta, pp. 317-322, 2012, doi: 10.1109/EMS.2012.30.

[3] Y. -I. Kim, S. -J. Kang, J. -M. Ko and S. -H. Choi, "A study for clustering method to generate Typical Load Profiles for Smart Grid," *8th International Conference on Power Electronics - ECCE Asia*, Jeju, Korea (South), pp. 1102-1109, 2011, doi: 10.1109/ICPE.2011.5944675.

[4] A. Pinceti, O. Kosut and L. Sankar, "Data-Driven Generation of Synthetic Load Datasets Preserving Spatio-Temporal Features," *2019 IEEE Power & Energy Society General Meeting (PESGM)*, Atlanta, GA, USA, 2019, pp. 1-5, doi: 10.1109/PESGM40551.2019.8973532.

[5] X. Liu, N. Iftikhar, H. Huo, R. Li, and P. Sieverts Nielsen, "Two approaches for synthesizing scalable residential energy consumption data." *Future Generation Computer Systems,* vol. 95, pp. 586-600, 2019.

[6] N. Iftikhar, X. Liu, F. E. Nordbjerg and S. Danalachi, "A Prediction-Based Smart Meter Data Generator," *2016 19th International Conference on Network-Based Information Systems (NBiS)*, Ostrava, Czech Republic, 2016, pp. 173-180, doi: 10.1109/NBiS.2016.15.

[7] J. Sarochar, I. Acharya, H. Riggs, A. Sundararajan, L. Wei, T. Olowu, A. I. Sarwat, "Synthesizing Energy Consumption Data Using a Mixture Density Network Integrated with Long Short Term Memory," *2019 IEEE Green Technologies Conference(GreenTech)*, Lafayette, LA, USA, 2019, pp. 1-4, doi: 10.1109/GreenTech.2019.8767148.

[8] G. -G. Pillai, G. -A. Putrus, N. -M. Pearsall. "Generation of synthetic benchmark electrical load profiles using publicly available load and weather data." *International Journal of Electrical Power & Energy Systems*, vol. 61, pp. 1-10, 2014.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative adversarial nets." *Communications of the ACM*, vol. 63, no. 11, pp. 139-144, 2020.

[10] Y. Gu, Q. Chen, K. Liu, L. Xie and C. Kang, "GAN-based Model for Residential Load Generation Considering Typical Consumption Patterns," *2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA, 2019, pp. 1-5, doi: 10.1109/ISGT.2019.8791575.

[11] Z. Wang and T. Hong, "Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN)," *Energy and Buildings*, vol. 224, pp. 0378-7788, Oct. 2020, doi: 10.1016/j.enbuild.2020.110299.

[12] C. Zhang, S. R. Kuppannagari, R. Kannan and V. K. Prasanna, "Generative Adversarial Network for Synthetic Time Series Data Generation in Smart Grids," *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Aalborg, Denmark, 2018, pp. 1-6, doi: 10.1109/SmartGridComm.2018.8587464.

[13] J. Zheng, L. Zhang, J. Chen, G. Wu, S. Ni, Z. Hu, C. Weng and Z. Chen, “Multiple-load forecasting for integrated energy system based on copula-DBiLSTM”, *Energies*, vol. 14, no. 8, p. 2188, 2021.

[14] H. Ren, Q. Li, Q. Wu, C. Zhang, Z. Dou and J. Chen, 2022. “Joint forecasting of multi-energy loads for a university based on copula theory and improved LSTM network”, *Energy Reports*, 8, pp.605-612, 2022.

[15] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” Jan. 2017, [Online]. Available: http://arxiv.org/abs/1701.07875

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved Training of Wasserstein GANs,” *International conference on machine learning*, pp. 214-223. PMLR, 2017.

[17] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift.” *International conference on machine learning*, pp. 448-456. pmlr, 2015.

[18] S. Haben, G. Giasemidis, F. Ziel, and S. Arora, “Short term load forecasting and the effect of temperature at the low voltage level,” *International Journal of Forecasting*, vol. 35, no. 4, pp. 1469–1484, Oct. 2019, doi: 10.1016/j.ijforecast.2018.10.007.

[19] L. Hu, L. Zhang, T. Wang and K. Li, "Short-term load forecasting based on support vector regression considering cooling load in summer," *2020 Chinese Control And Decision Conference (CCDC)*, Hefei, China, 2020, pp. 5495-5498, doi: 10.1109/CCDC49329.2020.9164387.

[20] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium.” *Advances in neural information processing systems*, vol. 30, 2017.

[21] Pytorch-Ignite Contributors, “Fréchet inception distance, PyTorch-Ignite v0.4.9 Documentation,” Accessed: Feb. 06, 2023, [Online]. Available: https://pytorch.org/ignite/generated/ignite.metrics.FID.html (accessed Feb. 06, 2023).

[22] “National Centers for Environmental Information,” Accessed: Feb. 06, 2023. [Online]. Available: https://www.ncei.noaa.gov/cdo-web/datasets/LCD/stations/WBAN:63819/detail

[23] I. Visconti, D. A. Lima and J. V . Milanović. "Comprehensive analysis of conservation voltage reduction: A real casestudy." 2019 IEEE Milan PowerTech. IEEE, 2019.

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

[25] 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.

[26] M. Diaz-Aguiló, J. Sandraz, R. Macwan, F. Leon, D. Czarkowski, C. Comack, and D. Wang, "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.

[27] K.P. Schneider, F.K. Tuffner, J.C. Fuller, and R. Singh, “Evaluation of conservation voltage reduction (CVR) on a national level.” Pacific Northwest National Lab (PNNL), Richland, WA (United States), no. PNNL-19596, 2010.

[28] K. Coughlin, M. 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), no. LBNL-63728, 2008.
[29] 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." 2019 IEEE Industry Applications Society Annual Meeting. IEEE, 2019.
[30] 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.
[31] H. P. Lee, L. Song, Y. Li, N. Lu, D. Wu, PJ Rehm, M. Makdad, E. Miller, "An Iterative Bidirectional Gradient Boosting Algorithm for CVR Baseline Estimation" 2023 IEEE Power & Energy Society General Meeting (PESGM), 2023.
[32] 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." 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia). IEEE, 2019.
[33] 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." 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe). IEEE, 2019.
[34] C, Bülte, M. Kleinebrahm, H.Ü. Yilmaz, and J. Gómez-Romero, “Multivariate time series imputation for energy data using neural networks.” Energy and AI, vol. 13, pp.100239, July 2023.
[35] Y. Chen, C. Chen, X. Zhang, M. Cui, F. Li, X. Wang, and S. Yin, "Privacy-Preserving Baseline Load Reconstruction for Residential Demand Response Considering Distributed Energy Resources." IEEE Transactions on Industrial Informatics, vol. 18, no. 5, pp. 3541-3550, 2022.
[36] A. Liguori, R. Markovic, M. Ferrando, J. Frisch, R. Causone, and C. Treeck, “Augmenting energy time-series for data-efficient imputation of missing values.” Applied Energy, vol. 334, pp. 120701, March 2023.
[37] Y. Weng, Y. Jiafan, and R. Rajagopal. "Probabilistic baseline estimation based on load patterns for better residential customer rewards." International Journal of Electrical Power & Energy Systems, no. 100, pp. 508-516, 2018.
[38] Y. Chen, P. Xu, Y. Chu, W. Li, L. Ni, Y. Bao, and K. Wang, "Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings." Applied Energy, no. 195, pp. 659-670, 2017.
[39] 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.
[40] M. Sun, Y. Wang, F. Teng, Y. Ye, G. Strbac, and C. Kang, et al. "Clustering-based residential baseline estimation: A probabilistic perspective." IEEE Transactions on Smart Grid, vol. 10, no. 6, pp. 6014-6028, 2019.
[41] Y. Zhang, Q. Ai, and Z. Li. "Improving aggregated baseline load estimation by Gaussian mixture model." Energy Reports, vol. 6, pp. 1221-1225, 2020.

[42] X. Ge, F. Xu, Y. Wang, H. Li, F. Wang, J. Hu, X. Lu, and B. Chen, "Spatio-Temporal Two-Dimensions Data Based Customer Baseline Load Estimation Approach Using LASSO Regression." IEEE Transactions on Industry Applications, vol. 58, no. 3, pp. 3112-3122, 2022.
[43] J. Chen, J. Yuan, W. Chen, A. Zeb, M. Suzauddola, and Y. A. Nanehkaran. "Research on Interpolation Method for Missing Electricity Consumption Data." Computers, Materials & Continua, vol, 78, no.2, 2024.
[44] J. Yoon, J. Jordon, and M. Schaar. "Gain: Missing data imputation using generative adversarial nets." International conference on machine learning. PMLR, 2018.
[45] Y. Luo, X. Cai, Y. Zhang, and J. Xu. "Multivariate time series imputation with generative adversarial networks." Advances in neural information processing systems 31, 2018.
[46] K. Zhang, X. Dou, and X. Xiao. "Grid Defect Data Completion Based on Generative Adversarial Imputation Nets." 2021 IEEE Sustainable Power and Energy Conference (iSPEC). IEEE, 2021.
[47] 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, June 2020.
[48] X. Hu, Z. Zhan, D. Ma and S. Zhang, "Spatiotemporal generative adversarial imputation networks: An approach to address missing data for wind turbines", IEEE Trans. Instrum. Meas., vol. 72, pp. 1-8, 2023.
[49] S. Ma, Z.-S. Xu and T. Sun, "Parallel generative adversarial imputation network for multivariate missing time-series reconstruction and its application to aeroengines", IEEE Trans. Instrum. Meas., vol. 72, pp. 1-16, 2023.
[50] Y. Li, L. Song, Y. Hu, H.P. Lee, D. Wu, PJ Rehm, and N. Lu, "Load Profile Inpainting for Missing Load Data Restoration and Baseline Estimation.", IEEE Transactions on Smart Grid, vol. 15, no. 2, pp. 2251-2260, Mar. 2024, doi: 10.1109/TSG.2023.3293188.
[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need.” In NIPS, 2017.
[52] J. Devlin, M. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of deep bidirectional transformers for language understanding.” In NAACL, 2019.
[53] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).
[54] K. Ye, H. Kim, Y. Hu, N. Lu, D. Wu, and P. Rehm, "A Modified Sequence-to-point HVAC Load Disaggregation Algorithm," 2023 IEEE Power & Energy Society General Meeting (PESGM), Orlando, FL, USA, 2023, pp. 1-5, doi: 10.1109/PESGM52003.2023.10252553.
[55] L. Song, Y. Li and N. Lu, "ProfileSR-GAN: A GAN Based Super-Resolution Method for Generating High-Resolution Load Profiles," in IEEE Transactions on Smart Grid, vol. 13, no. 4, pp. 3278-3289, July 2022, doi: 10.1109/TSG.2022.3158235.
[56] S. Haben, G. Giasemidis, F. Ziel, and S. Arora, “Short term load forecasting and the effect of temperature at the low voltage level,” International Journal of Forecasting, vol. 35, no. 4, pp. 1469–1484, Oct. 2019, doi: 10.1016/j.ijforecast.2018.10.007.
[57] D. Bahdanau, K. Cho, and Y. Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
[58] T. M. Cover, J. A. Thomas, “Elements of Information Theory, 2nd Edition”, Wiley, pp. 80.

[59] “National Centers for Environmental Information,” Accessed: Feb. 06, 2023. [Online]. Available: https://www.ncei.noaa.gov/cdo- web/datasets/LCD/stations/WBAN:63819/detail

[60] Yi Hu, Yiyan Li, Lidong Song, Han Pyo Lee, PJ Rehm, Mattew Makdad, Edmond Miller, and Ning Lu, "MultiLoad-GAN: A GAN-Based Synthetic Load Group Generation Method Considering Spatial-Temporal Correlations," in IEEE Transactions on Smart Grid, vol. 15, no. 2, pp. 2309-2320, Mar. 2024, doi: 10.1109/TSG.2023.3302192.

[61] X. Lin, W. Wang, Y. Li, S. Yang, F. Feng, Y. Wei, and T. Chua. "Data-efficient Fine-tuning for LLM-based Recommendation." arXiv preprint arXiv:2401.17197 (2024).

[62] I. Visconti, D. A. Lima and J. V . Milanović. "Comprehensive analysis of conservation voltage reduction: A real casestudy." 2019 IEEE Milan PowerTech. IEEE, 2019.

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

[64] 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.

[65] M. Diaz-Aguiló, J. Sandraz, R. Macwan, F. Leon, D. Czarkowski, C. Comack, and D. Wang, "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.

[66] K.P. Schneider, F.K. Tuffner, J.C. Fuller, and R. Singh, “Evaluation of conservation voltage reduction (CVR) on a national level.” Pacific Northwest National Lab.(PNNL), Richland, WA (United States), no. PNNL-19596, 2010.

[67] K. Coughlin, M. 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), no. LBNL-63728, 2008.

[68] 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." 2019 IEEE Industry Applications Society Annual Meeting. IEEE, 2019.

[69] 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.

[70] H. P. Lee, L. Song, Y. Li, N. Lu, D. Wu, PJ Rehm, M. Makdad, E. Miller, "An Iterative Bidirectional Gradient Boosting Algorithm for CVR Baseline Estimation" 23PESGM0022, submitted to 2023 IEEE PES General Meeting, Available online at: http://arxiv.org/abs/2211.03733.

[71] 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." 2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia). IEEE, 2019.

[72] 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." 2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe). IEEE, 2019.

[73] Y. Chen, C. Chen, X. Zhang, M. Cui, F. Li, X. Wang, and S. Yin, "Privacy-Preserving Baseline Load Reconstruction for Residential Demand Response Considering Distributed Energy Resources." IEEE Transactions on Industrial Informatics, vol. 18, no. 5, pp. 3541-3550, 2022.
[74] Y. Weng, Y. Jiafan, and R. Rajagopal. "Probabilistic baseline estimation based on load patterns for better residential customer rewards." International Journal of Electrical Power & Energy Systems, no. 100, pp. 508-516, 2018.
[75] Y. Chen, P. Xu, Y. Chu, W. Li, L. Ni, Y. Bao, and K. Wang, "Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings." Applied Energy, no. 195, pp. 659-670, 2017.
[76] 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.
[77] M. Sun, Y. Wang, F. Teng, Y. Ye, G. Strbac, and C. Kang, et al. "Clustering-based residential baseline estimation: A probabilistic perspective." IEEE Transactions on Smart Grid, vol. 10, no. 6, pp. 6014-6028, 2019.
[78] Y. Zhang, Q. Ai, and Z. Li. "Improving aggregated baseline load estimation by Gaussian mixture model." Energy Reports, vol. 6, pp. 1221-1225, 2020.
[79] X. Ge, F. Xu, Y. Wang, H. Li, F. Wang, J. Hu, X. Lu, and B. Chen, "Spatio-Temporal Two-Dimensions Data Based Customer Baseline Load Estimation Approach Using LASSO Regression." IEEE Transactions on Industry Applications, vol. 58, no. 3, pp. 3112-3122, 2022.
[80] J. Yoon, J. Jordon, and M. Schaar. "Gain: Missing data imputation using generative adversarial nets." International conference on machine learning. PMLR, 2018.
[81] Y. Luo, X. Cai, Y. Zhang, and J. Xu. "Multivariate time series imputation with generative adversarial networks." Advances in neural information processing systems 31, 2018.
[82] K. Zhang, X. Dou, and X. Xiao. "Grid Defect Data Completion Based on Generative Adversarial Imputation Nets." 2021 IEEE Sustainable Power and Energy Conference (iSPEC). IEEE, 2021.
[83] 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, June 2020.
[84] X. Hu, Z. Zhan, D. Ma and S. Zhang, "Spatiotemporal generative adversarial imputation networks: An approach to address missing data for wind turbines", IEEE Trans. Instrum. Meas., vol. 72, pp. 1-8, 2023.
[85] S. Ma, Z.-S. Xu and T. Sun, "Parallel generative adversarial imputation network for multivariate missing time-series reconstruction and its application to aeroengines", IEEE Trans. Instrum. Meas., vol. 72, pp. 1-16, 2023.
[86] Y. Li, L. Song, Y. Hu, H.P. Lee, D. Wu, PJ Rehm, and N. Lu, "Load Profile Inpainting for Missing Load Data Restoration and Baseline Estimation." arXiv preprint arXiv:2211.16332 (2022).
[87] Y. Hu, K. Ye, H. Kim, and N. Lu, “BERT-PIN: A BERT-based Framework for Recovering Missing Data Segments in Time-series Load Profiles”, 2023, arXiv: 2310.17742. Available: http://arxiv.org/abs/2310.17742
[88] J. Li, G. Li, C. Tao, H. Zhang, F. Liu, and Z. Jin. "Large Language Model-Aware In-Context Learning for Code Generation.", 2023, *arXiv preprint arXiv:2310.09748*.

[89] J. Li, Y. Zhao, Y. Li, G. Li, and Z. Jin. "Acecoder: Utilizing existing code to enhance code generation.", 2023, *arXiv preprint arXiv:2303.17780*.

[90] X. Chen, M. Lin, N. Schärli, and D. Zhou. "Teaching large language models to self-debug.", 2023, *arXiv* preprint arXiv:2304.05128.

[91] D. Tiro, "The Possibility of Applying ChatGPT (AI) for Calculations in Mechanical Engineering." *In International Conference "New Technologies, Development and Applications"*, pp. 313-320. Cham: Springer Nature Switzerland, 2023.

[92] S. Frieder, L. Pinchetti, R.-R. Griffiths, T. Salvatori, T. Lukasiewicz, P. C. Petersen, A. Chevalier, and J. Berner, "Mathematical capabilities of chatgpt,", 2023, *arXiv preprint arXiv:2301.13867*.

[93] Y. Wardat, M. A. Tashtoush, R. AlAli, and A. M. Jarrah, "Chatgpt: A revolutionary tool for teaching and learning mathematics," *Eurasia Journal of Mathematics, Science and Technology Education*, vol. 19, no. 7, p. em2286, 2023.

[94] X. Wang, N. Anwer, Y. Dai, and A. Liu. "ChatGPT for design, manufacturing, and education." *Procedia CIRP*, vol. 119, pp. 7-14, 2023.

[95] S. Badini, S. Regondi, E. Frontoni, and R. Pugliese. "Assessing the capabilities of ChatGPT to improve additive manufacturing troubleshooting." *Advanced Industrial and Engineering Polymer Research*, vol. 6, pp. 278-287, July 2023.

[96] Yang, Hongyang, Xiao-Yang Liu, and Christina Dan Wang. "FinGPT: Open-Source Financial Large Language Models.", 2023, *arXiv preprint arXiv:2306.06031*.

[97] A. Gilson, C. W. Safranek, T. Huang, V. Socrates, L. Chi, R. A. Taylor, D. Chartash, et al., "How does chatgpt perform on the United States medical licensing examination? the implications of large language models for medical education and knowledge assessment," JMIR Medical Education, vol. 9, no. 1, p. e45312, 2023.

[98] A. Rao, J. Kim, M. Kamineni, M. Pang, W. Lie, and M. D. Succi, "Evaluating chatgpt as an adjunct for radiologic decision-making," medRxiv, pp. 2023–02, 2023.

[99] D. Duong and B. D. Solomon, "Analysis of large-language model versus human performance for genetics questions," medRxiv, pp. 2023–01, 2023.

[100] R. S. D'Amico, T. G. White, H. A. Shah, and D. J. Langer, "I asked a chatgpt to write an editorial about how we can incorporate chatbots into neurosurgical research and patient care. . . ,", Neurosurgery, vol. 92, no. 4, pp. 663-664, 2023.

[101] J. Li, Y. Yang, and J. Sun. "Risks of Practicing Large Language Models in Smart Grid: Threat Modeling and Validation." arXiv preprint arXiv:2405.06237 (2024).

[102] A. Zaboli, S. Choi, T. Song, and J. Hong. "ChatGPT and other Large Language Models for Cybersecurity of Smart Grid Applications." arXiv preprint arXiv:2311.05462 (2023).

[103] J. Ruan, G. Liang, H. Zhao, G. Liu, X. Sun, J. Qiu, Z. Xu, F. Wen, and Z. Dong. "Applying Large Language Models to Power Systems: Potential Security Threats." IEEE Transactions on Smart Grid, vol. 15, no. 3, pp. 3333-3336, May, 2024.

[104] "Create chat completion", Accessed: May 24, 2024. [Online]. Available: https://platform.openai.com/docs/api-reference/chat/create

[105] "Fine-tune UI", Accessed: May 24, 2024. [Online]. Available: https://platform.openai.com/finetune

[106] "National Centers for Environmental Information," Accessed: Feb. 06, 2023. [Online]. Available: https://www.ncei.noaa.gov/cdo- web/datasets/LCD/stations/WBAN:63819/detail

[107] "Pricing information", Accessed: May 24, 2024. [Online]. Available: https://openai.com/api/pricing/

[108] D.-I. Kim, T. Y. Chun, S.-H. Yoon, G. Lee, and Y.-J. Shin, “Wavelet-based event detection method using PMU data,” IEEE Transactions on Smart Grid, vol. 8, no. 3, pp. 1154–1162, Oct. 2015.

[109] S.-W. Sohn, A. J. Allen, S. Kulkarni, W. M. Grady, and S. Santoso, “Event detection method for the PMUs synchrophasor data,” in 2012 IEEE Power Electronics and Machines in Wind Applications, Jul. 2012, pp. 1–7.

[110] L. Xie, Y. Chen, and P. Kumar, “Dimensionality reduction of synchrophasor data for early event detection: Linearized analysis,” IEEE Transactions on Power Systems, vol. 29, no. 6, pp. 2784–2794, Nov. 2014.

[111] J. Wu, J. Xiong, P. Shil, and Y. Shi, “Real time anomaly detection in wide area monitoring of smart grids,” in 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), Nov. 2014, pp. 197– 204.

[112] W. Li, M. Wang, and J. H. Chow, “Real-time event identification through low-dimensional subspace characterization of high-dimensional synchrophasor data,” IEEE Trans. Power Syst., vol. 33, no. 5, pp. 4937–4947, Sep. 2018.

[113] O. P. Dahal and S. M. Brahma, “Preliminary work to classify the disturbance events recorded by phasor measurement units,” in Proc. IEEE Power Energy Soc. Gen. Meeting, 2012, pp. 1–8.

[114] W. Wang et al., “Frequency disturbance event detection based on synchrophasors and deep learning,” IEEE Trans. Smart Grid, vol. 11, no. 4, pp. 3593–3605, Jul. 2020.

[115] I. Niazazari et al., “PMU-data-driven event classification in power transmission grids,” in Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT), 2021, pp. 1–5.

[116] M. Pavlovski, M. Alqudah, T. Dokic, A. A. Hai, M. Kezunovic, and Z. Obradovic, “Hierarchical convolutional neural networks for event classification on PMU measurements,” IEEE Trans. Instrum. Meas., vol. 70, pp. 1–13, 2021.

[117] Y. Yuan, Y. Guo, K. Dehghanpour, Z. Wang, and Y. Wang, “Learningbased real-time event identification using rich real PMU data,” IEEE Trans. Power Syst., vol. 36, no. 6, pp. 5044–5055, Nov. 2021.

[118] B. P. Leao, D. Fradkin, Y. Wang, and S. Suresh, “Big data processing for power grid event detection,” in Proc. IEEE Int. Conf. Big Data (Big Data), 2020, pp. 4128–4136.

[119] R. Chandrakar, R. Dubey, and B. Panigrahi. "Deep-Learning based Multiple Class Events Detection and Classification using Micro-PMU Data." In 2024 8th International Conference on Green Energy and Applications (ICGEA), pp. 137-142. IEEE, 2024.

[120] Y. Cheng, N. Yu, B. Foggo, and K. Yamashita. "Online power system event detection via bidirectional generative adversarial networks." IEEE Transactions on Power Systems 37, no. 6 (2022): 4807-4818.

[121] Z. Cheng, J. Schmidt, Y. Hu, D. Hart, and J. Holbach. "Automated High-Speed Power System Event Detection and Classification Using Synchrophasor Data." U.S. Patent Application 18/131,155, filed October 10, 2024.