# Data Efficiency Assessment of Generative Adversarial Networks in Energy Applications

Umme Mahbuba Nabila[a,*], Linyu Lin[b], Xingang Zhao[c], William L. Gurecky[d], Pradeep Ramuhalli[d], Majdi I. Radaideh[a,*]

[a] *Department of Nuclear Engineering and Radiological Sciences, University of Michigan, Ann Arbor, MI 48109, United States*

[b] *Nuclear Science & Technology Division, Idaho National Laboratory Idaho Falls, ID 83415, United States*

[c] *Department of Nuclear Engineering, University of Tennessee, Knoxville, TN 37996, United States*

[d] *Nuclear Energy and Fuel Cycle Division, Oak Ridge National Laboratory, Oak Ridge, TN 37830, United States*

**Abstract**

This study investigates the data requirements of generative artificial intelligence (AI), particularly generative adversarial networks (GANs), for reliable data augmentation in energy applications. Generative AI, though seen as a solution to data limitations, requires substantial data to learn meaningful distributions—a challenge often overlooked. This study addresses the challenge through synthetic data generation for critical heat flux (CHF) and power grid demand, focusing on renewable and nuclear energy. Two variants of GAN employed are conditional GAN (cGAN) and Wasserstein GAN (wGAN). Our findings include the strong dependency of GAN on data size, with performance declining on smaller datasets and varying performance when generalizing to unseen experiments. Mass flux and heated length significantly influence CHF predictions. wGAN is more robust to feature exclusion, making it suitable for constrained synthetic data generation. In energy demand forecasting, wGAN performed well for solar, wind, and load predictions. Longer lookback hours and larger datasets improved predictions, especially for load power. Seasonal variations posed challenges, with wGAN achieving a relatively high error of Root Mean Squared Error (RMSE) of 0.32 for load power prediction. Feature exclusions impacted cGAN the most, while wGAN showed greater robustness. This study concludes that, while generative AI is effective for data augmentation, it requires substantial data and careful training to generate realistic synthetic data and generalize to new experiments in engineering applications.

*Keywords:*

Generative AI, Generative Adversarial Networks, Critical Heat Flux, Data Augmentation, Power Grid Energy Forecasting

## 1. Introduction

Artificial intelligence and machine learning (AI/ML) are currently at the forefront of reshaping traditional practices in almost every sector, from healthcare to finance, from education to manufacturing, by unlocking new possibilities and promising efficiency, innovation, and unparalleled advancements [1]. It has a far-reaching impact across various domains of the engineering industry, including intelligent design and simulation for energy systems [2, 3, 4], autonomous systems and predictive maintenance [5, 6], data augmentation and documentation [7, 8],

*Corresponding Authors: Umme Mahbuba Nabila (unabila@umich.edu), Majdi I. Radaideh (radaideh@umich.edu)

quality control and inspection processes [9, 10], and environmental impact assessment [11]. However, in tandem with the myriad advantages, there are a multitude of challenges. One prominent challenge is data quality and availability. AI/ML algorithms depend heavily on large datasets for training and decision-making [12]. Ensuring the quality, relevance, and availability of diverse datasets poses a significant challenge, especially in domains where data may be scarce or sensitive, like in engineering domains [13, 14] or simulations are expensive that add additional challenges on data generation [15]. One potential solution that stands out most to the significant challenge posed by a shortage of training datasets is generative AI (GenAI). The GenAI domain is dedicated to creating algorithms and models that can produce synthetic data that closely mimic real-world data [16]. It has vastly influenced many industries, including media, marketing, education, medicine, architecture, software development, and nuclear [17]. It can be used to generate meaningful new content, which has paved the way for diverse applications, such as creating images and videos, generating text, augmenting data, composing poems and music, and developing chatbots with human-like characteristics [18, 19, 20]. In this study, we evaluate the performance of a GenAI-based model in generating synthetic data for both steady-state and time-series energy applications, by assessing their robustness to fewer training dataset, missing input features, their generalization to unseen experiments, and benchmarks their effectiveness against traditional regression models.

Popular variants of GenAI models include generative adversarial networks (GANs), variational autoencoders, normalizing flow, and transformers. GAN was introduced by Goodfellow in 2014, which is a novel generative model based on minimax game theory [21]. It has two networks: the generator and the discriminator. The generator creates as realistic data as possible to deceive the discriminator, which then attempts to distinguish fake samples from real ones. The training process continues until the generator succeeds in fooling the discriminator to no longer distinguish the fake data [22]. In 2013, Diederik P. Kingma and Max Welling introduced variational autoencoders, a special autoencoder based on stochastic variational inference and deep learning [23]. It is comprised of three parts: encoder, latent space, and decoder. The encoder compresses the input data into a lower dimensional latent space. The decoder then decompresses and tries to recreate the input data from the latent space. Although the framework of normalizing flow was introduced before, it was primarily popularized by Rezende and Mohamed in their 2015 article "Variational Inference with Normalizing Flows" [24]. Normalizing flows employ a series of invertible transformations to map a simple distribution to a more complex distribution, enabling the generation of realistic data samples [25]. The transformer architecture was introduced by Vaswani et al. [26]. Transformer architecture is based on the self-attention mechanism, which allows the model to weigh different parts of the input sequence differently when making predictions. Although all generative models can be targeted for assessing their data needs, this paper focuses on GAN models as one of the primary variants of GenAI.

Due to the wide range of applications, various types of GANs have been proposed with a specific purpose [27], including conditional GAN (cGAN), deep convolutional GAN, Laplacian GAN, progressive-growing GAN, cycle GAN, and Wasserstein GAN (wGAN) [28]. For example, Wang et al. used cGAN to generate artificial fault samples for a mechanical fault diagnosis in bearings and gearboxes [29]. Another variation, cycle GAN, was deployed by Branikas et al. to generate realistic images of cracks for visual defect detection in advanced gas-cooled reactor cores [30]. wGAN, introduced by Arjovsky et al. in 2017, addresses the instability of training traditional GANs by utilizing the Wasserstein distance (Earth mover's distance) as a metric for measuring the divergence between the real and generated data distributions [31]. This reformulation improves model convergence, mitigates issues like mode collapse, and makes it a worthy model for synthetic data generation.

Another significant application of GAN is forecasting and generating precise time-series data. For instance, Smith et al. used the concept cGAN to develop a time-series GAN model and showed that it effectively generates realistic synthetic one-dimensional signals across a diverse range of data types, including sensor readings, medical

data, simulated processes, and motion data [32]. Several studies have explored the use of cGANs and wGANs for forecasting applications across various domains, showcasing their potential in generating accurate time-series predictions [33, 34, 35, 36]. Continuing on this trajectory, our research examines the data needs of cGAN and conditional wGAN, leveraging their distinct loss functions and robust capability to handle complex static and transient datasets effectively. Both variants of GAN i.e., cGAN and wGAN utilize the concept of conditional GAN to better compare the synthetic data with real data at the specific input points.

To fully exploit the potential of deep learning and ML, a large training dataset is imperative. It has already been established that the larger the datasets are, the better results that can be obtained from deep learning models, including GenAI and GAN [37], which is not always the case for most engineering applications [38]. This situation is even worse for the energy industry, like the nuclear industry, where the experiments are costly, simulations are computationally expensive [14], and applications are sensitive to uncertainties and public interest such as spent nuclear fuel [39, 40]. To overcome this hurdle, researchers are seeking data augmentation to enlarge the real limited dataset and synthesize as realistic training data as possible. However, limited research has been conducted to determine the data needed for GenAI techniques to create new samples without losing accuracy.

In this paper, we implement GAN variants on two different datasets: a static dataset based on critical heat flux (CHF) measurements from nuclear reactor experiments and a time-dependent dataset based power grid demand data from renewable energy sources. The CHF data contains 21,453 CHF measurements in vertical water-cooled uniformly heated tubes collected from 59 different sources over a span of 60 years [41]. Our objective is to generate synthetic CHF data that is as realistic as possible using GAN models while assessing its data needs, that is, how much data and information GAN requires from real experiments to generate high-quality synthetic CHF data. The second dataset utilized in this study is the time-dependent dataset derived from the power-systems machine learning (PSML) dataset developed by Zheng et al. [42]. This vast dataset was collected over three years (2018–2020) from 66 regions across the United States featuring weather conditions and load demand. For this work, the minute-level data from Zone 1 of the California Independent System Operator (CAISO) was selected to explore the ability of GAN to generate realistic power grid demand with time while evaluating the dataset requirements for accurate forecasting. The major contributions of this work are:

1. Conduct a detailed evaluation of GAN performance, including cGAN and wGAN, in generating synthetic data for two energy-related applications: a steady-state static experimental setup and a time-series forecasting problem. In both cases, GAN models are exposed to varying forms of training data conditions to assess different aspects of their performance.
2. Evaluate the capability of GAN models to reliably generate data when certain features are excluded from the model inputs, reflecting typical challenges in energy and engineering applications where some signals may be unavailable or difficult to measure.
3. Analyze the generalization ability of GAN models to generate data for new, unseen experiments when trained on different experimental setups.
4. Benchmark the performance of GAN models with traditional regression models, utilizing feedforward and recurrent neural networks (FNNs and RNNs).

The structure of the remaining sections of this paper is organized as follows: Section 2 describes the training data collection and preprocessing methods conducted for this study. The deep generative models implemented in this paper, along with the assessment cases, are presented in Section 3. Section 4 provides the major findings obtained from GAN models and the performance assessment of their data needs based on the CHF and power grid datasets. Finally, the conclusions drawn from this study and the prospective approach of future research are highlighted in Section 5.

**Nomenclature**

| | |
|---|---|
| AI | Artificial Intelligence |
| CAISO | California Independent System Operator |
| cGAN | Conditional Generative Adversarial Network |
| CHF | Critical Heat Flux |
| DHI | Direct Horizontal Irradiance |
| DNI | Direct Normal Irradiance |
| FNN | Feedforward Neural Network |
| GAN | Generative Adversarial Network |
| GenAI | Generative Artificial Intelligence |
| GHI | Global Horizontal Irradiance |
| GRU | Gated Recurrent Unit |
| MAPE | Mean Absolute Percentage Error |
| ML | Machine Learning |
| RNN | Recurrent Neural Network |
| wGAN | Wasserstein Generative Adversarial Network |

## 2. Data Collection and Processing

In this section, we describe the two datasets used in this study: the CHF dataset and the power grid demand dataset. These datasets are selected to evaluate the effectiveness of GAN in generating realistic synthetic data for both static and time-dependent scenarios. Each dataset offers unique challenges and characteristics, which are detailed in the following subsections.

### *2.1. Critical Heat Flux*

This section reports the CHF dataset used in the analysis. For a nuclear reactor, one of the major safety limits is the local heat flux. Although higher heat flux is desirable for higher enrichment, it cannot be increased indefinitely. The limit is referred to as CHF. Beyond this value, the heat transfer coefficient deteriorates significantly. This leads to higher reactor temperature and fuel failure. Depending on the operating condition of the nuclear reactor, this phenomenon is also known as boiling crisis, departure from nucleate boiling, and dryout. Over the years, CHF for vertical water-cooled tubes has been measured in many experiments worldwide. In 2019, the U.S. Nuclear Regulatory Commission (NRC) published a compiled database which is widely known as 2006 Groeneveld CHF lookup table [43]. A cleaned-up version of the NRC database was provided to the Organisation for Economic Co-operation and Development (OECD) and Nuclear Energy Agency (NEA) for benchmarking purposes [41]. This dataset includes 21,453 CHF measurement points. All references and an overview of all test facilities that have contributed to developing the NRC CHF database are documented in [43]. The available data consist of **boundary conditions**: pressure ($P$), mass flux ($G$), inlet temperature ($T_{in}$); **geometrical parameters**: test section diameter ($D$) and heated length ($L$); and **parameters derived from measurements and water properties**: outlet equilibrium quality ($X$), inlet enthalpy ($H_{in}$), and critical heat flux ($CHF$).

Based on prior analysis of this dataset using traditional regression techniques, this study uses five input parameters, including boundary conditions and geometrical parameters, as input parameters to the generative model, while $CHF$ is the parameter to be generated by GAN. The parameter ranges of the dataset used in this study are demonstrated in Table 1.

### *2.2. PSML Dataset*

The second measured dataset is a time-series dataset from PSML dataset. This multiscale time-series dataset was developed by Zheng et al. by capturing real-world and synthetic time-series data across various spatiotemporal scales, including minute-level and millisecond-level measurements from the electric grid [42]. Minute-level data was collected over a period of 3 years, from 2018 to 2020, across 66 regions in the United States. Zone 1 of CAISO was used for this study. The dataset includes load power, wind power, and solar power, along with key

Table 1: Parameter spans of the database used in this study

| Variable | Diameter (mm) | Length (m) | Pressure (kPa) | Mass Flux (kg/m$^2$s) | Temperature (°C) | CHF (kW/m$^2$) |
|---|---|---|---|---|---|---|
| Minimum | 2.39 | 0.07 | 100.0 | 17.7 | 9.0 | 130.0 |
| Maximum | 16.0 | 15.0 | 20,000.0 | 7,712.0 | 353.62 | 13,345.0 |

meteorological parameters such as diffuse horizontal irradiance (DHI), direct normal irradiance (DNI), and global horizontal irradiance (GHI), dew point, solar zenith angle, wind speed, relative humidity, and temperature.

A visualization of feature trends over a time period selected for this forecasting study is given in Figure 1. In this study, the three power variables are generated by GAN (e.g., model outputs), while the weather-related data act as GAN inputs.

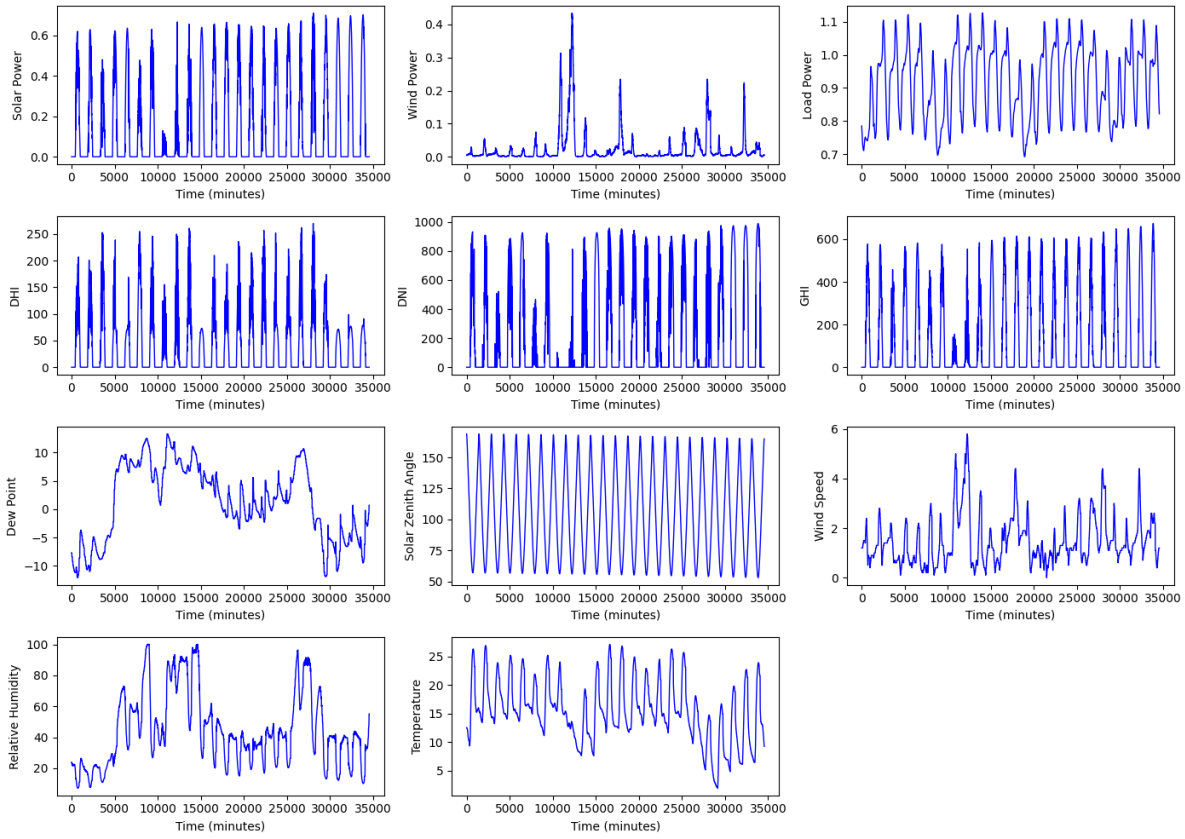



Figure 1: Plot of weather-related and load variables for a selected time period in CAISO Zone 1 dataset

Instead of using the original minute level dataset, we used a smoothed version where a 5 minute averaging technique was employed. This preprocessing step was implemented to reduce the computational complexity of model training while preserving the underlying patterns and trends. The comparison between the original and smoothed data power trends, as shown in Figure 2, indicates that the smoothing process did not affect the dataset's temporal dynamics. Additionally, an overall mean absolute percentage error (MAPE) of 3.4% between the original and smoothed datasets, L2 norm difference of 0.018, and maximum element-wise absolute difference of 0.00838 confirm that the smoothed data closely approximate the original dataset.

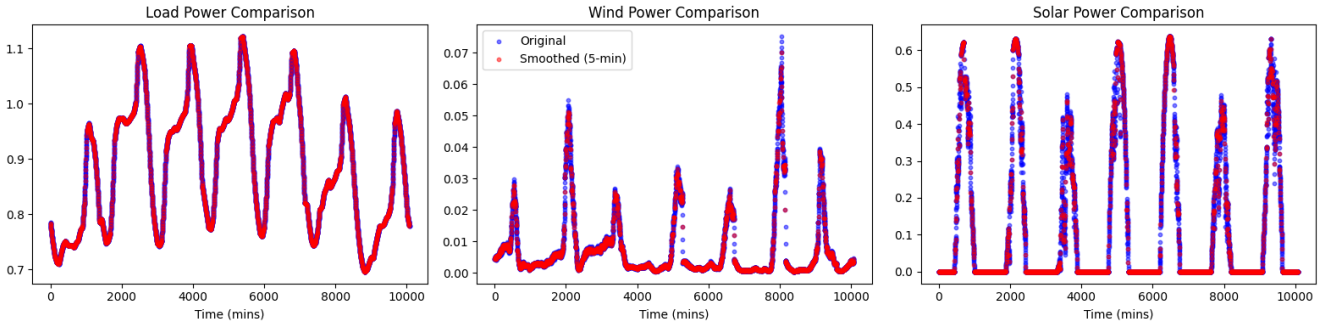



Figure 2: Comparison of original and smoothed dataset

## 3. Methodology

### *3.1. Generative Adversarial Network*

GANs are a class of ML models that consist of two neural networks: the generator $\mathcal{G}$ and the discriminator $\mathcal{D}$. They have gained significant attention due to their ability to generate synthetic data that closely resemble real data distributions. The generator takes as input a random noise vector $z$ sampled from a prior distribution $p_z(z)$ and produces synthetic data samples $\hat{x} = \mathcal{G}(z)$. The discriminator, on the other hand, aims to distinguish between real data samples $x$ from the true data distribution $p_{\text{data}}(x)$ and fake samples $\hat{x}$ produced by the generator [21]. The objective of the generator is to maximize the probability that the discriminator misclassifies its generated samples as real.

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D},\mathcal{G}) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log \mathcal{D}(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-\mathcal{D}(\mathcal{G}(z)))], \tag{1}$$

where $V(\mathcal{D},\mathcal{G})$ represents the value function or the adversarial loss. Here, the first term in the objective function represents the log probability that the discriminator correctly classifies real data as real, and the second term represents the log probability that the discriminator incorrectly classifies fake data as real.

#### *3.1.1. Conditional Generative Adversarial Network*

Unlike vanilla GAN, cGANs are not completely unsupervised in their training methods. Instead, cGAN architectures implement class labels or labeled data to execute specific tasks effectively. For creating a cGAN structure, a small adjustment is applied to the vanilla GAN architecture by adding a "$y$-label" to both the discriminator and generator networks. This adjustment transforms the previous probabilities into conditional probabilities [44]. With this modification, the training process guarantees that the generator produces outputs aligned with the specified labels, which are provided as conditions. Similarly, the discriminator scrutinizes the authenticity of the generated output, ensuring it matches the expected label. Consequently, after training, providing a specific input label will yield the desired output from the generative network. Figure 3 illustrates the typical cGAN structure.

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D},\mathcal{G}) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log \mathcal{D}(x|y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-\mathcal{D}(\mathcal{G}(z|y)))] \tag{2}$$

#### *3.1.2. Wasserstein Generative Adversarial Network*

Traditional cGANs use the binary cross-entropy (BCE) loss mentioned in the above formulas, equation 1 and equation 2, which is effective for distinguishing between real and fake data samples. However, BCE can lead to training instability and mode collapse, particularly when there is minimal overlap between the real and generated

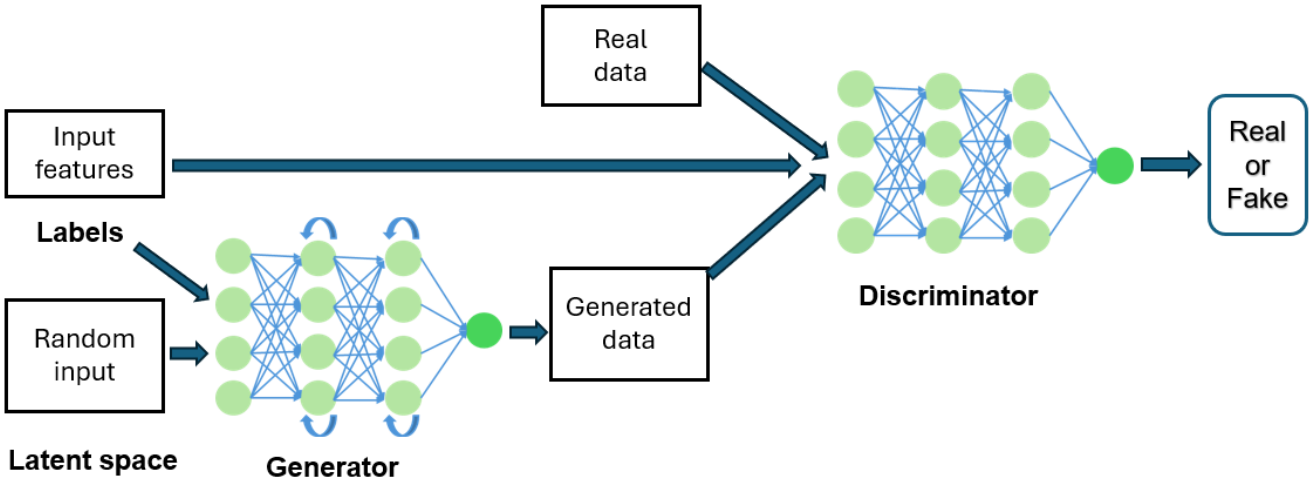



Figure 3: A typical cGAN structure

data distributions. The wGAN addresses the limitations of BCE loss by replacing it with the Wasserstein distance, or Earth mover's distance, as a measure of divergence between the real and generated data distributions [31]. In wGAN, instead of training the discriminator to classify samples as real or fake, the discriminator (also known as the "critic" in wGAN) is trained to provide a continuous score that estimates how "real" a sample is. The critic network is designed to assign higher scores to real samples and lower scores to generated samples, effectively learning to approximate the Wasserstein distance between real data and the generated data distribution. The generator network is trained to minimize this distance by producing samples that the critic scores similarly to real data, thereby reducing the gap between the two distributions. wGAN reformulates the cGAN loss function to use the expected values of the critic outputs for real samples and generated samples, as shown in the equation below:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\mathcal{D}(x|y)] - \mathbb{E}_{z \sim p_z(z)}[\mathcal{D}(\mathcal{G}(z|y))] \tag{3}$$

where $\mathcal{D}$ is a $k$-Lipschitz function, which is essential for accurately approximating the Wasserstein distance. wGAN enforces this property by applying a weight clipping constraint on the critic's parameters within a bounded interval $[-c, c]$, typically with c=0.01. This modification eliminates the need for logarithmic terms in the loss function, enhancing training stability.

Although weight clipping stabilizes training, it also restricts the critic's capacity, often pushing weights toward the limits of the clipping range. This constraint can result in issues such as exploding or vanishing gradients. To address these challenges, wGAN with gradient penalty replaces weight clipping with a gradient penalty to maintain the Lipschitz constraint. By penalizing deviations from a gradient norm of 1, this enables a more flexible and effective approximation of the Wasserstein distance, avoiding the drawbacks associated with strict weight clipping.

### *3.2. Gated Recurrent Unit*

Gated recurrent unit (GRU) is an advanced RNN designed to address the vanishing gradient problem in standard RNNs [45]. It incorporates update and reset gates to selectively retain or discard information, effectively preserving relevant past information across long sequences. The equations governing the GRU cell are:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{4}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{5}$$

$$\tilde{h}_t = \tanh(W_h x_t + r_t \odot (U_h h_{t-1}) + b_h) \tag{6}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{7}$$

where: $x_t$ is the input at time step $t$, $h_{t-1}$ is the hidden state from the previous time step, $z_t$ and $r_t$ are the update and reset gates, respectively, $\tilde{h}_t$ is the candidate hidden state, $\sigma$ denotes the sigmoid activation function, $W_z, W_r, W_h, U_z, U_r, U_h$ are the weight matrices, and $b_z, b_r, b_h$ are the bias terms.

The other neural network model used in this study to serve as either a generator or discriminator is the popular FNN, which is also known as a fully connected network, dense network, or multilayer perception [46].

GRU and FNN are included for benchmark purposes. Here, FNN acts as a baseline to measure the performance of cGAN and wGAN in generating synthetic data. FNNs are purely predictive and are not designed for data generation but rather for interpolating between the existing training points. Unlike cGAN and wGAN, which generate synthetic data from random noise by learning the underlying data distribution. Hence, comparing FNNs with GANs helps quantify the added value of GANs' ability to capture complex data distributions and produce new samples. Similarly, GRU helps to justify the requirement and purpose of complex GAN models to generate new time-series data where recurrent models are advantageous.

In this study, we explored the cGAN performance with traditional loss functions and Wasserstein loss, incorporating the "conditional" training framework in both models. *Conditional GAN is referred to as cGAN and conditional wGAN is referred to as wGAN for convenience.* Note that vanilla GAN is not used in this study due to its limitation in generating labeled data. Additionally, when GAN is applied to the static CHF dataset, FNN is selected as the generator. However, when GAN is applied to the time-series power grid dataset, GRU is chosen as the generator. The discriminator/critic model in both cases is FNNs.

### *3.3. Data Assessment Cases*

#### *3.3.1. Critical Heat Flux Dataset*

Table 2 illustrates three different cases investigated in this research work. In Case 1, we aim to assess the impact of training dataset size on model performance for generating new synthetic data in the test (unseen) CHF values. For example, whether GAN or wGAN is trained with 200 or 10,000 samples, it will be evaluated based on the same test set. Case 2 is dedicated to examining the possibility of generating CHF data of a specific experiment setup after training GAN with a dataset sourced from another experiment setup. Five experimental setups are chosen for this case study out of 59 setups due to the larger number of CHF measurements available in those setups. These experiments are Alekseev et al. (1,018 data points), Becker et al. (2,185 data points), Kirillov et al. (2,339 data points), Smolin et al. (2,755 data points), and Zenkevich et al. (4,639 data points) [43]. Case 2 includes two scenarios as indicated in Table 2. Finally, Case 3 explores the effect of the input parameters (conditions) on CHF generation quality.

To assess the accuracy of the generated CHF synthetic data, MAPE, $R^2$ (coefficient of determination) and Kullback-Leibler (KL) divergence are calculated as fitness parameters. A grid search was carried out to find the optimum number of layers, nodes, learning rate, and batch size that resulted in the best MAPE and $R^2$ values for cGAN and wGAN architectures.

Table 2: Description of the test cases used to evaluate GAN data needs for CHF dataset

| Case | Description |
|---|---|
| Case 1 | Test dataset size is 20% of total data. Training dataset size varies from 100 to 17,162 samples (80% of total data). |
| Case 2 | Scenario 1: Five selected experiment setups are used and combined (total 12,936 samples). The training dataset is 80% and the test dataset is the remaining 20%. |
| | Scenario 2: An experiment setup is taken as the training dataset, the remaining four experiment setups are used as a test dataset, and performance metrics are calculated by taking the average over the four test sets. |
| Case 3 | Scenario 1: Four out of five input parameters are used, and one parameter was excluded. Five models can be trained where GAN performance was measured by swapping the input parameter to be excluded. |
| | Scenario 2: Three input parameters are used, and two input parameters are excluded. Ten combinations are investigated to measure the effect of different pairs of excluded parameters. |

The MAPE is a metric used to evaluate model accuracy. It measures the average absolute percentage difference between the actual and predicted values. The formula for calculating MAPE for the CHF case is:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{CHF_i - \widehat{CHF}_i}{CHF_i}\right| \times 100\% \tag{8}$$

where $CHF_i$ represents the actual CHF values in a test set, $\widehat{CHF}_i$ represents the generated values, and $n$ is the total number of samples in the test set. Lower MAPE values indicate higher accuracy, meaning that the generated CHF values are closer to the actual values. $R^2$ is the popular statistical measure that represents the proportion of the variance in the dependent variable that is predictable from the independent variables in a regression model. It is often used as a measure of how well the regression model fits the data, with higher values indicating a better fit. In this case, the generated CHF data by cGAN or wGAN captures the actual values. The KL divergence is a probability-based metric that quantifies the difference between the distribution of generated CHF values and the actual CHF distribution. It measures how much extra information is required to represent one distribution using another. A lower KL divergence indicates that the synthetic CHF data closely matches the real data distribution, meaning the generative model captures the underlying statistical properties more effectively. Given the extensive results associated with this dataset, we present KL divergence findings only for the first case indicated in Table 4, as its trends were consistent with those observed using MAPE and $R^2$, leading to similar conclusions.

### *3.3.2. Power Grid Dataset*

Table 3 outlines the four evaluation cases used to assess the performance of GAN models in predicting power grid demand. These cases strategically explore the effects of lookback window, training dataset sizes, seasonal variations, and input parameter exclusions on model performance. The reference/standard case is chosen as follows: training dataset size is the first 2 weeks, and the test dataset size is the following 10 days with both lookback and lookforward time windows of 12 hours.

Case 1 investigates how varying the lookback window size influences the generative model's performance for a fixed lookforward window. The lookback window refers to the amount of past data (time steps) the model considers as input to learn patterns and make predictions. The lookforward window specifies how far into the future the model predicts or generates data based on the input from the lookback window. This analysis allows the evaluation of the temporal dependency of power grid data and determines the optimal lookback window for training GAN models. Next, Case 2 examines the impact of the size of the training dataset on the GAN's performance. This

case provides an insight into the minimum amount of data required for accurate generation performance. Case 3 evaluates the model's generation ability across seasonal variations by switching training and testing data from winter and summer months. This helps to assess how the model would perform when it is challenged with unseen data. Case 4 investigates the effect of excluding specific input parameters on the GAN model's performance.

Table 3: Description of the test cases used to evaluate GAN data needs in time-series data of power grid demand

| **Case** | **Description** |
|---|---|
| Case 1 | Lookforward is 12 hours and lookback varies from 2 hours, 4 hours, 6 hours, up to 12 hours with 2 hours increment |
| Case 2 | Training dataset size varies from 24 hours, 48 hours, 72 hours, up to 2 weeks hours with strategic increment. Test dataset size is fixed 10 days. |
| Case 3 | Scenario 1: Training on winter months, testing on summer months. |
| | Scenario 2: Training on summer months, testing on winter months. |
| Case 4 | Scenario 1: One parameter is excluded each time, and the rest of the input parameters are used for training. Eight models can be trained where GAN performance was measured by swapping the input parameter to be excluded. |
| | Scenario 2: Different pairs of input parameters are excluded and the rest of the input parameters are used. 28 combinations are investigated to measure the effect of different pairs of excluded parameters. |

The performance of the models for power prediction can be evaluated using mean absolute error or root mean square error (RMSE), both of which provide an indication of the time-average magnitude of the differences between GAN-generated time-series data and the actual data. Due to the volume of the results associated with this dataset, we only report RMSE results, given the authors did not find any noticeable differences in the conclusions when mean absolute error is also reported.

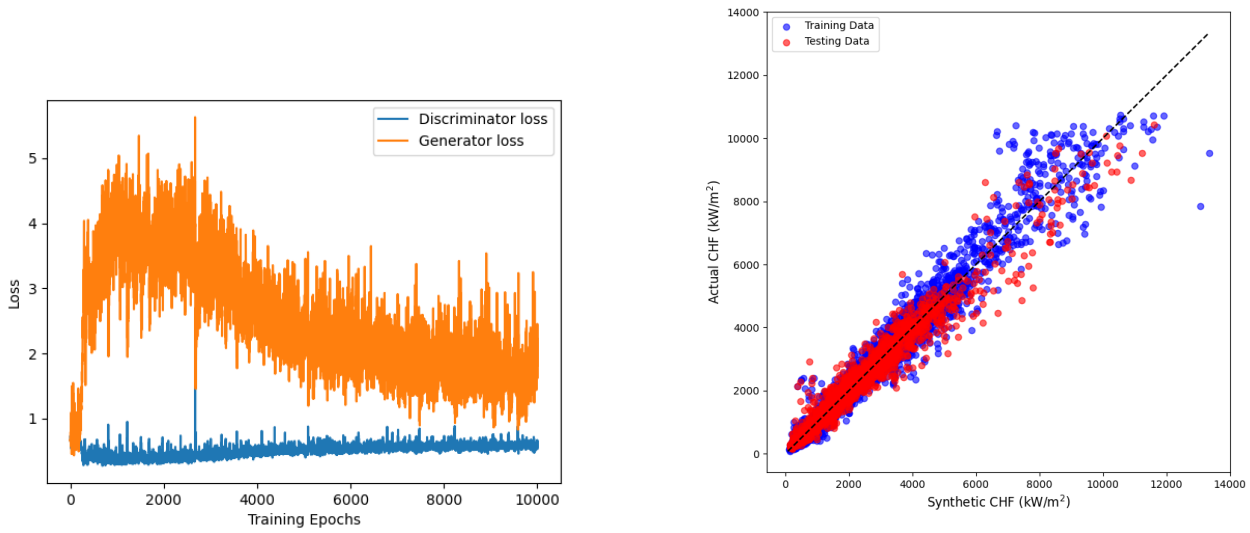
## 4. Results and Discussions

As a sample result, we report the generator and discriminator loss curves for a sample case from the CHF dataset, these loss curves demonstrate how GAN is trained. The cGAN results for the maximum data points in Case 1 for CHF (see Table 2) are displayed in Figure 4a. The generator loss reaches a high value at the beginning, gradually decreases, and finally stabilizes to a lower value as training progresses, indicating its improvement in producing synthetic data close to real data. On the other hand, the discriminator loss begins with a very low value and slowly increases with higher epochs, which indicates its diminishing ability to correctly distinguish real from fake data. The stability of these loss curves demonstrates the effective convergence of this adversarial training.

The adjacent plot, Figure 4b, compares the synthetic CHF values against the actual CHF values for both the training (blue points) and testing datasets (red points) for the same CHF case study. The tight clustering of points around the dashed line for both datasets indicates good agreement between the synthetic and actual CHF values. This result demonstrates the ability of GAN (in this case cGAN) to generate synthetic CHF data that closely resembles real experimental data, even for unseen test cases.

### *4.1. Generative Adversarial Network Assessment on the Critical Heat Flux Dataset*

Table 4 presents the performance metrics MAPE, $R^2$ and KL divergence values found for Case 1 using cGAN, wGAN, and FNN where different training dataset sizes were employed. For better comprehension, the comparison is also demonstrated in Figure 5. The findings reveal that, as the training dataset size decreases, there was a significant decline in the overall performance of every model. For cGAN, MAPE increased nearly twofold from 9.08% with the maximum training dataset to 19.53% when the training dataset size was reduced to 100 samples. wGAN exhibited




(a) (b)

Figure 4: (a) Evolution of generator and discriminator loss during cGAN training and (b) comparison of synthetic and actual CHF for training and testing data. Case 1 from Table 2 with maximum training data points is shown in these figures.

a similar trend, where MAPE increased from 7.30% with the largest dataset to 57.20% for the smallest dataset. Overall, wGAN consistently outperformed cGAN for larger datasets, but its performance deteriorated rapidly for smaller datasets. Additionally, While KL divergence remains relatively low and stable for cGAN and FNN, it increases significantly for wGAN for fewer training size.

These findings suggest that smaller dataset sizes result in less accurate predictions by GenAI models. The metric values after using 2,000 training samples started to saturate as the changes were marginal. The same conclusion can also be seen for $R^2$ values between 0.71 and 0.83 for the smallest training sizes and 0.98 and 0.96 for the largest training sizes for wGAN and cGAN, respectively. With more data available for training, the model can learn more patterns and relationships in the data, leading to more accurate predictions and a better ability to generate more accurate synthetic CHF data.

If we compare GANs with FNN, GANs have comparable accuracy when larger datasets are used. However, FNN's consistent accuracy across dataset sizes suggests it is a reliable option for predictive tasks, especially when the primary goal is not synthetic data generation but accurate regression.

Table 4: Performance metrics for GAN (cGAN, wGAN), and FNN for declining training dataset sizes for the CHF dataset.

| Training Dataset Size | MAPE (%) | | | $R^2$ | | | KL Divergence | | |
|---|---|---|---|---|---|---|---|---|---|
| | cGAN | wGAN | FNN | cGAN | wGAN | FNN | cGAN | wGAN | FNN |
| 17,162 | 9.08 | 7.30 | **3.48** | 0.96 | 0.98 | **0.99** | 0.0137 | 0.011 | 0.0088 |
| 10,000 | 9.94 | 8.40 | **3.62** | 0.96 | 0.96 | **0.99** | 0.015 | 0.013 | 0.0109 |
| 2,000 | 9.38 | 8.60 | **5.03** | 0.96 | 0.96 | **0.99** | 0.016 | 0.014 | 0.0064 |
| 1,000 | 9.66 | 17.60 | **5.78** | 0.96 | 0.91 | **0.98** | 0.017 | 0.017 | 0.0214 |
| 500 | 13.68 | 22.90 | **6.89** | 0.93 | 0.86 | **0.98** | 0.026 | 0.018 | 0.014 |
| 200 | 17.24 | 23.06 | **9.30** | 0.91 | 0.86 | **0.97** | 0.028 | 0.018 | 0.017 |
| 100 | 19.53 | 57.20 | **12.56** | 0.83 | 0.71 | **0.93** | 0.068 | 0.286 | 0.025 |

Next, the fitness metrics for Case 2 are shown in Table 5 and Figure 6, where one experiment setup was a training set and the other four were test sets. Due to a large difference in MAPE values, Alekseev's metrics are not

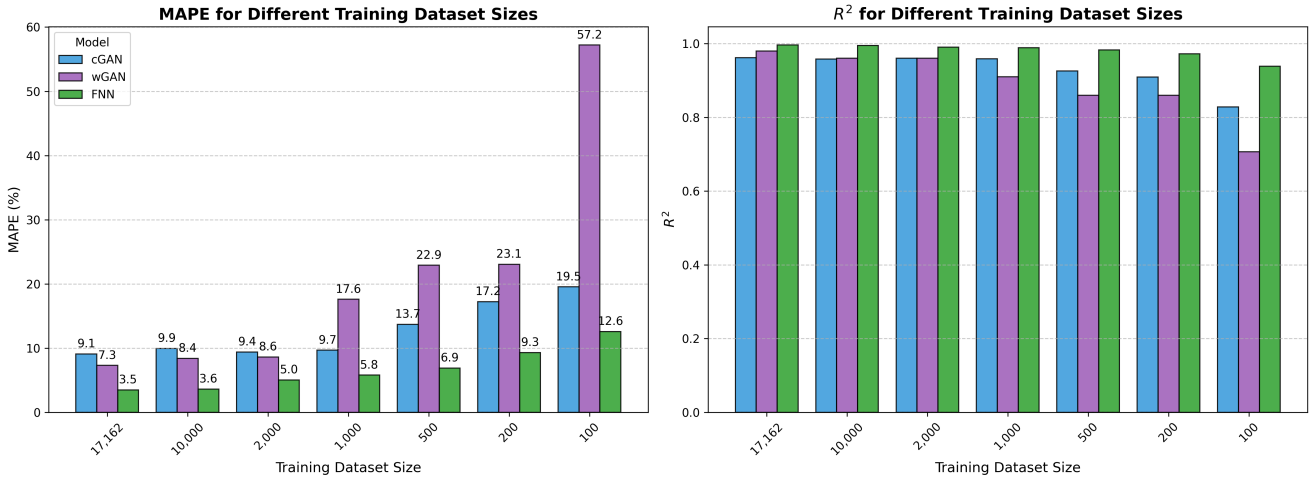



Figure 5: Comparison of MAPE and $R^2$ for Case 1 with different training sizes for the CHF dataset

Table 5: Performance metrics for GAN (cGAN, wGAN), and FNN methods across different experiments for the CHF dataset

| Experiment | MAPE (%) | | | $R^2$ | | |
|---|---|---|---|---|---|---|
| | cGAN | wGAN | FNN | cGAN | wGAN | FNN |
| Subset of Five experiments | 7.04 | 10.03 | **2.63** | 0.98 | 0.96 | **0.99** |
| Alekseev et al. | 271.33 | **182.69** | 400.13 | **-5.11** | -7.12 | -32.64 |
| Becker et al. | 27.91 | 29.28 | **17.96** | 0.79 | 0.69 | **0.90** |
| Kirillov et al. | **28.51** | 38.80 | 36.30 | **0.78** | 0.62 | 0.60 |
| Smolin et al. | 17.72 | 43.25 | **10.68** | 0.81 | 0.52 | **0.93** |
| Zenkevich et al. | **16.77** | 24.01 | 19.17 | **0.82** | 0.76 | 0.79 |

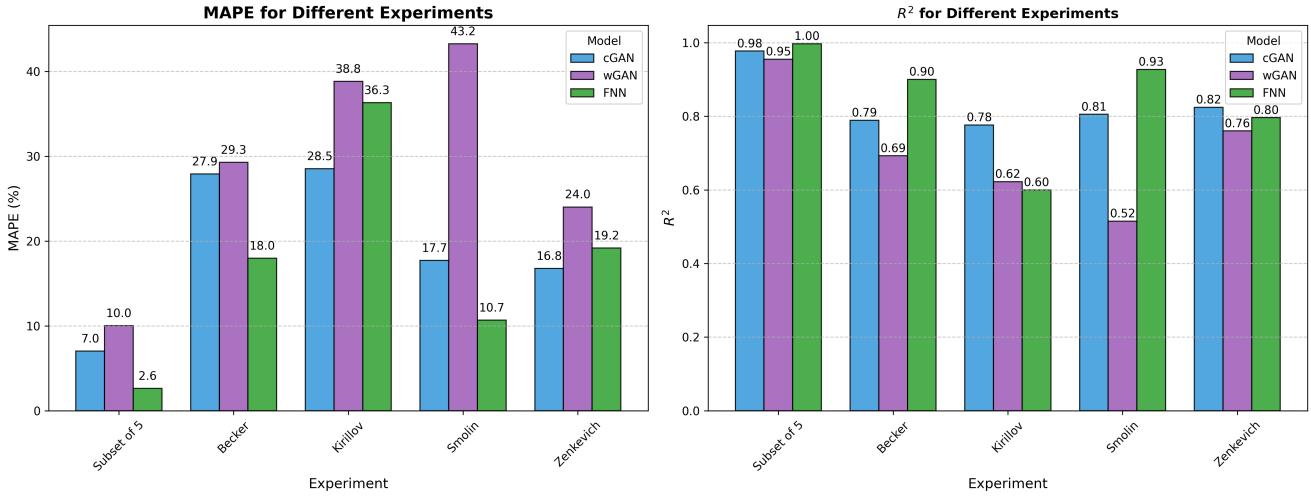



Figure 6: Comparison of MAPE and $R^2$ for Case 2 on different experiments for the CHF dataset

included in Figure 6. Similar to what was observed in Case 1, cGAN performs better than wGAN with a smaller dataset. For cGAN, the dataset from Smolin et al. and Zenkevich et al. [43] have a relatively fair MAPE of 17.72% and 16.77% and $R^2$ of 0.805 and 0.824, respectively, which indicate a fair accuracy of generation when Smolin et al. and Zenkevich et al. measurements are used to generate the other four experiments. Two other experiment setups

(i.e., Becker and Kirillov [43]) show lower accuracy when being used as training data. For example, Kirillov et al. metrics are a MAPE of 28.51% and $R^2 = 0.776$. cGAN with Alekseev et al. data completely failed to reproduce any of the four other experiments. This is because their experiments do not adequately encompass the experimental conditions in the other setups, causing both cGAN and wGAN to heavily extrapolate in their generated data.

A noteworthy finding from Case 2 highlights a clear limitation of generative models, particularly GAN in our context, which often rely on exposure to a large number of samples from various configurations. Although Scenario 1 yielded commendable metrics when cGAN or wGAN has seen a lot of data from all five experiments, their performance declined by at least 10% when tasked with replicating experiment setups it had not encountered previously. This finding highlights that we need to be very careful about how we use GAN or GenAI in synthetic data generation and how much margin we have in extrapolation.

Table 6 showcases the fitness metrics of GAN models for Case 3 Scenario 1 when four input parameters out of five were used during training, and one was excluded each time. A visual depiction of the comparison is provided in Figure 7. The bar graph shows that, in the case of cGAN, a higher $R^2$ value was achieved when the diameter and pressure were not used as inputs, implying less impact on the cGAN generation ability of CHF. However, the results highlight the importance of length and mass flux in CHF data generation, as their exclusion leads to substantial performance degradation across all models, particularly for cGAN and FNN. The inlet temperature has a moderate impact on CHF and comes in the middle of the five parameters. wGAN has displayed better performance than cGAN for most of the cases, showcasing its ability to handle the challenge of fewer input features. These feature importance result follows similar trends previously shown by various studies conducted on feature importance analysis in CHF prediction [47]. Moreover, the strong influence of mass flux and pressure on GAN-predicted CHF values is consistent with established empirical trends, reinforcing the physical significance of these parameters in CHF modeling. Among the well-known correlations, the Barnett correlation is widely applied to uniformly heated annular tubes and rod bundles, incorporating mass flux and hydraulic diameter as key predictors. The W-3 correlation, developed for Pressurized Water Reactors (PWRs), relies on local quality and mass flux, while the WSC-2 correlation, designed for subchannel applications, refines CHF predictions by incorporating the effects of pressure and flow rate [48].

Table 6: Performance metrics for Case 3 Scenario 1 on a single excluded parameter for the CHF dataset

| **Excluded Parameter** | **MAPE (%)** | | | $R^2$ | | |
|---|---|---|---|---|---|---|
| | **cGAN** | **wGAN** | **FNN** | **cGAN** | **wGAN** | **FNN** |
| Diameter | 16.84 | 17.40 | **10.96** | 0.91 | 0.90 | **0.98** |
| Length | 56.69 | **27.70** | 36.74 | 0.36 | **0.79** | 0.77 |
| Pressure | 14.68 | 16.80 | **9.83** | 0.94 | 0.93 | **0.97** |
| Mass Flux | 40.90 | **17.56** | 31.92 | 0.49 | **0.87** | 0.81 |
| Temperature | 27.33 | **18.18** | 18.77 | 0.79 | 0.85 | **0.90** |

To study the effect of parameter combinations on GAN, the findings of Case 3 Scenario 2 are documented in Table 7 and Figure 8. First, GAN performance in all cases of Scenario 2 is worse than in Scenario 1, implying that any feature exclusion could have a detrimental impact on GAN generation accuracy, let alone multiple feature exclusions. Matching our conclusions from Case 3 Scenario 1, the combination of diameter and pressure did not cause a large reduction in performance metrics, while the length–mass flux pair caused a huge drop in $R^2$. This indicates that length and mass flux strongly influence the generation of accurate CHF values. wGAN consistently demonstrates robustness in handling the absence of critical parameters, making it a suitable choice for synthetic data generation under constrained feature availability. The results also indicate that the FNN model consistently struggles with feature exclusion, indicating its limitation for interpolation when limited features are available.

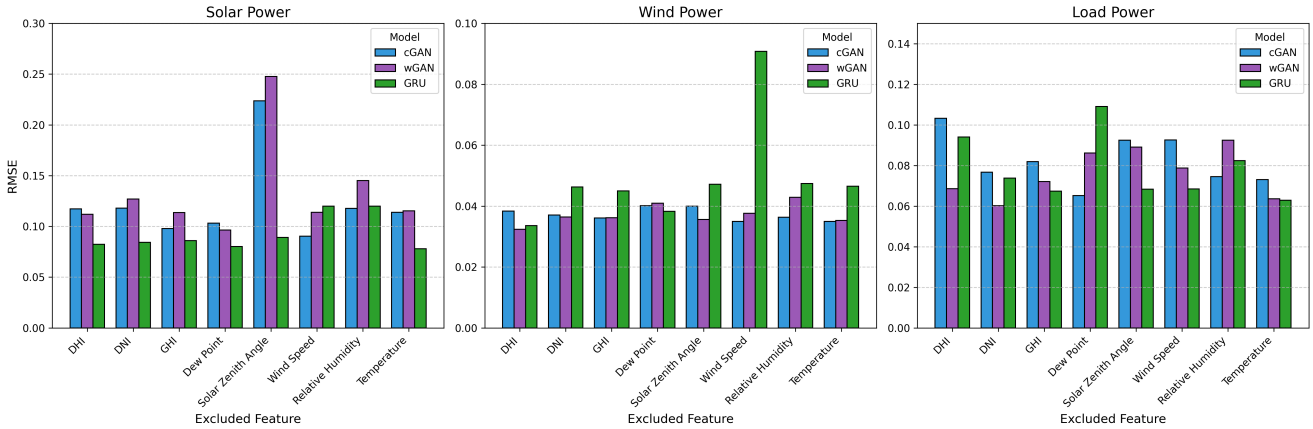



Figure 7: Comparison of metrics for Case 3 Scenario 1 for the CHF dataset

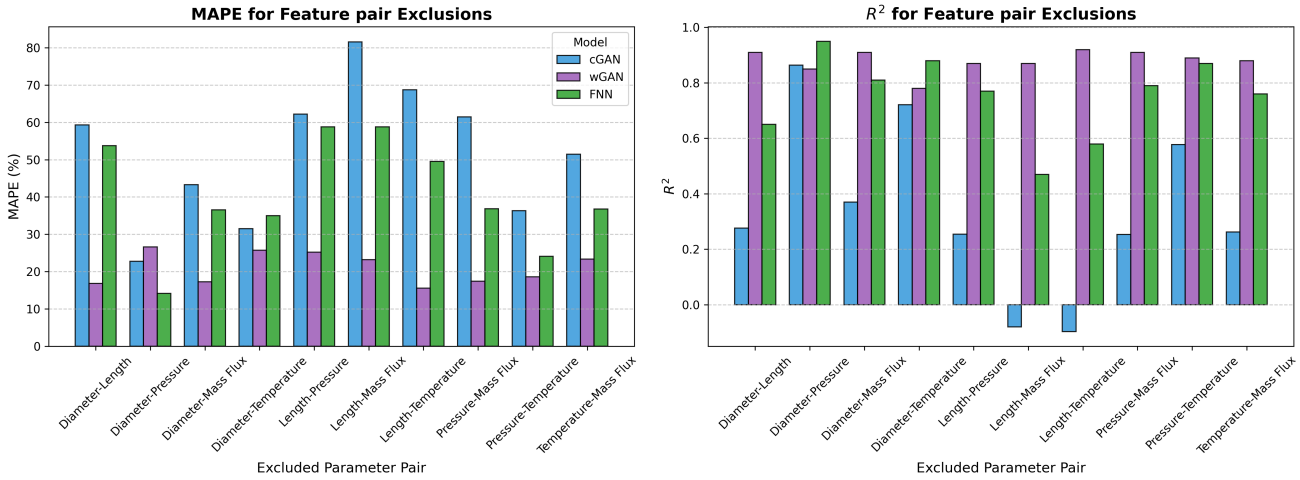



Figure 8: Comparison of metrics for Case 3 Scenario 2 for the CHF dataset

### *4.2. Generative Adversarial Network Assessment on the Power Grid Dataset*

Figure 9 provides a visual comparison of the predictive performance of cGAN, wGAN, and GRU models for solar, wind, and load power under the standard/reference condition of 12 hours lookback and 12 hours lookforward time windows. wGAN performed the best for load power, by closely following the power trend with time, while all the models performed well for solar power prediction, but failed to capture the peaks in the wind power trend. Overall, solar power prediction is the most accurate, which can be explained by the plethora of available input features dedicated to it, such as DHI, DNI, GHI, solar zenith angle, and temperature. There are a few input features directly related to wind power, particularly wind speed, relative humidity, and dew point. However, load power lacks any input feature that directly accounts for its transient. This may explain the poor model performance in predicting the load power.

Table 8 compares the RMSE values for solar, wind, and load power predictions across different lookback hours (2, 4, 6, 8, 10, and 12) using GRU, cGAN, and wGAN. Additionally, a visual depiction of the comparison is provided in Figure 10. The results highlight the strengths and weaknesses of each model across power types and time horizons. For solar power, GRU achieved the lowest RMSE for all the lookback periods. For wind power,

Table 7: Performance metrics (MAPE and $R^2$) for Case 3 Scenario 2 on excluded parameter pairs for the CHF dataset

| **Excluded Parameter Pair** | **MAPE (%)** | | | $R^2$ | | |
|---|---|---|---|---|---|---|
| | **cGAN** | **wGAN** | **FNN** | **cGAN** | **wGAN** | **FNN** |
| Diameter–Length | 59.31 | **16.82** | 53.78 | 0.276 | **0.91** | 0.65 |
| Diameter–Pressure | 22.75 | 26.6 | **14.16** | 0.864 | 0.85 | **0.95** |
| Diameter–Mass Flux | 43.3 | **17.29** | 36.52 | 0.37 | **0.91** | 0.81 |
| Diameter–Temperature | 31.48 | 29.69 | **23.36** | 0.721 | 0.78 | **0.88** |
| Length–Pressure | 62.18 | **25.89** | 35.01 | 0.254 | **0.84** | 0.77 |
| Length–Mass Flux | 81.56 | **23.19** | 58.77 | -0.08 | **0.87** | 0.47 |
| Length–Temperature | 68.71 | **15.53** | 49.54 | 0.097 | **0.92** | 0.58 |
| Pressure–Mass Flux | 61.46 | **17.43** | 36.86 | 0.253 | **0.91** | 0.79 |
| Pressure–Temperature | 36.32 | **18.6** | 24.06 | 0.578 | **0.9** | 0.87 |
| Temperature–Mass Flux | 51.44 | **23.36** | 36.75 | 0.262 | **0.88** | 0.76 |

wGAN demonstrated superior performance, achieving the lowest RMSE for most lookbacks (2, 4, 10, and 12 hours). Finally, for load power, wGAN achieved the lowest RMSE values for 4, 8, 10, and 12 hour lookbacks, demonstrating its advantage in modeling load power dynamics over longer time horizons. cGAN showed moderate performance but did not outperform GRU or wGAN.

Table 8: RMSE values for solar power, wind power, and load power predictions with varying lookback hours for the power grid dataset.

| Lookback Hours | Solar RMSE | | | Wind RMSE | | | Load RMSE | | |
|---|---|---|---|---|---|---|---|---|---|
| | cGAN | wGAN | GRU | cGAN | wGAN | GRU | cGAN | wGAN | GRU |
| 2 | 0.130 | 0.106 | **0.093** | 0.034 | **0.032** | 0.037 | 0.093 | 0.100 | **0.074** |
| 4 | 0.123 | 0.100 | **0.073** | 0.035 | **0.035** | 0.038 | 0.095 | **0.068** | 0.072 |
| 6 | 0.121 | 0.095 | **0.079** | 0.036 | 0.033 | **0.032** | 0.091 | 0.073 | **0.061** |
| 8 | 0.097 | 0.111 | **0.088** | 0.037 | 0.035 | **0.034** | 0.090 | **0.074** | 0.088 |
| 10 | 0.132 | 0.107 | **0.077** | 0.039 | **0.034** | 0.047 | 0.098 | **0.087** | 0.106 |
| 12 | 0.097 | 0.108 | **0.083** | 0.035 | 0.036 | **0.033** | 0.080 | **0.070** | 0.084 |

Benchmarking GAN-based forecasting performance against industrial standards is challenging in this context due to the complexity and variability of influencing factors—such as geographic location (e.g., state), weather conditions, and the availability of additional features to inform the forecasted variables. For instance, forecasting solar load proved to be easier than forecasting total load power, primarily because more relevant features were available for the solar load predictions. Despite these challenges, the GAN performance metrics reported in this study align well with those found in related work using other methods that are not generative by nature for energy load forecasting. For example, Nowotarski et al. [49] employed an ensemble of forecasting techniques on ISO New England data, while Butt et al. [50] applied traditional LSTM methods for power forecasting in Islamabad. Hasanat et al. [51] used a hybrid GRU and convolutional neural network for forecasting energy demand in the ISO New England dataset, and Agrawal et al. [52] also used LSTM-based models. Our evaluation metrics, including RMSE, are comparable to the MAE and RMSE results reported across these studies.

The RMSE values for solar, wind, and load power predictions across different training dataset sizes (24, 48, 96, 120, 168, 240, and 336 hours) are documented in Table 9. This comparison is shown in Figure 11. Increasing training size improves the accuracy of all models, but load power prediction benefits the most from larger datasets. The lowest RMSE values of wind power (0.033 from GRU, 0.035 from cGAN, wGAN) are obtained for the largest dataset (336 hours). Similarly, the lowest RMSE (0.070 for 336 hours) for load power was obtained using wGAN.

The impact of seasonal variation on model performance was investigated in this case. The training data consisted of 2 weeks from either summer or winter, while the test data comprised 10 days from the opposite season. The best RMSE achieved for load power was 0.32 (compared to 0.07) with wGAN, indicating that none

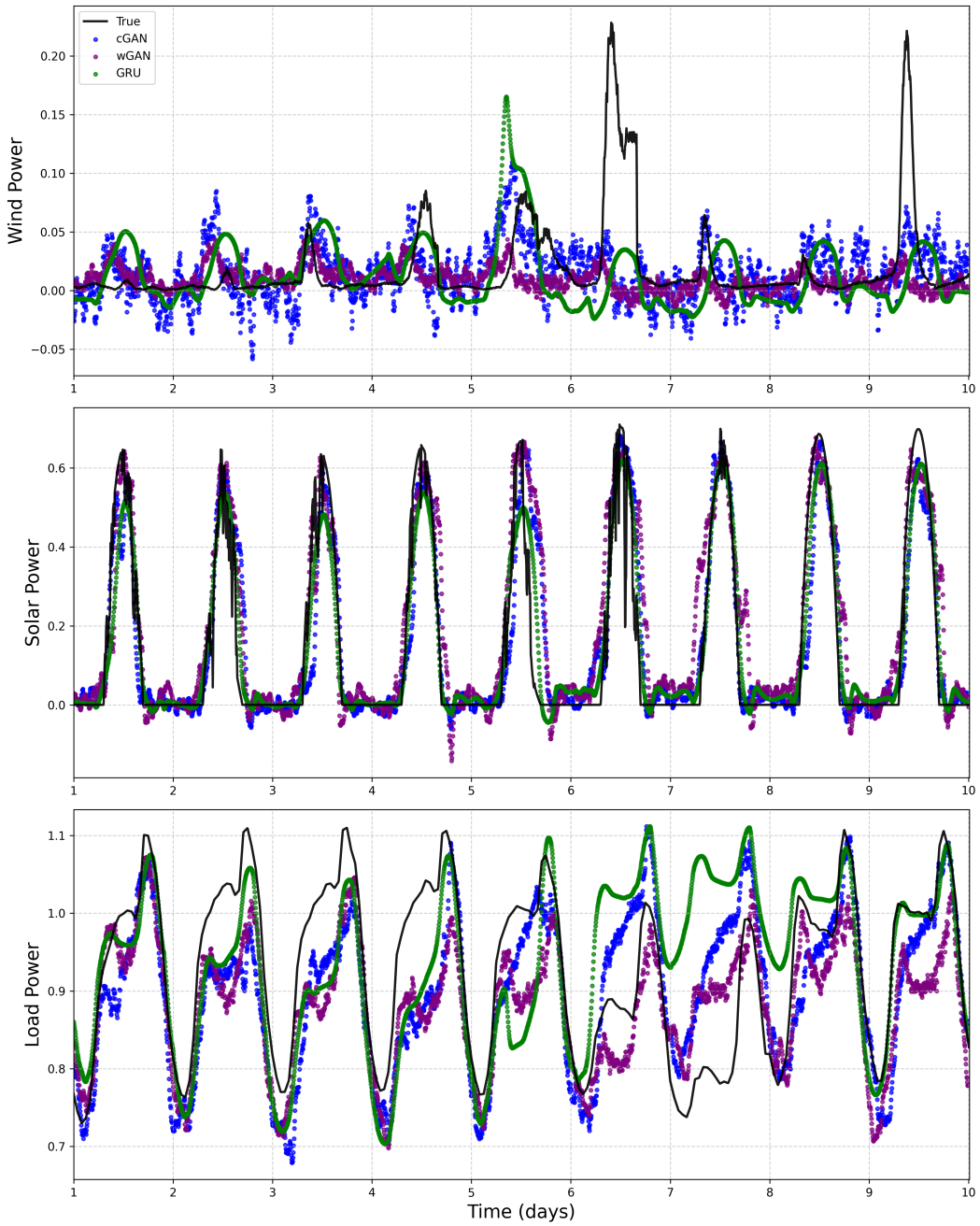



Figure 9: Comparison of power grid prediction using cGAN, wGAN, and GRU

of the models performed well under these conditions. Since there are only two scenarios in this case, results are neither tabulated nor plotted in figure. Similar to the CHF dataset, the generative models struggle to generalize to new conditions beyond their training conditions. Future work may explore transfer learning to fine-tune GAN on limited recent data, as well as hybrid architectures such as GAN-Transformers for long-range dependencies, physics-informed GANs to enforce seasonal constraints, or ensemble GANs trained across different seasons. These approaches can provide a more adaptable framework for power demand forecasting in scenarios with limited or seasonally inconsistent data availability.

For Case 4 on feature exclusion, the results are indicated in Table 10 and Figure 12, which highlight the critical

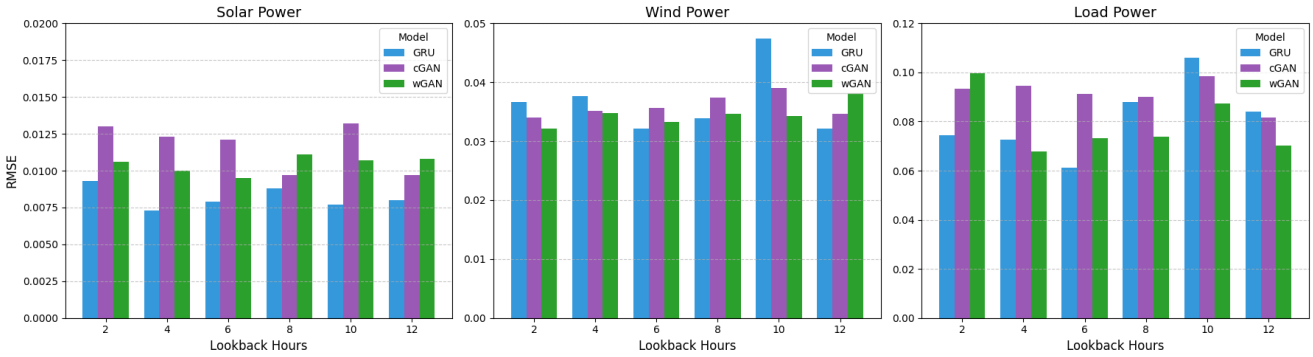



Figure 10: Comparison of RMSE for Case 1 with different lookback hours for the power grid dataset

Table 9: RMSE values for solar power, wind power, and load power predictions with varying training dataset sizes for the power grid dataset

| Training Datasize (hours) | Solar RMSE | | | Wind RMSE | | | Load RMSE | | |
|---|---|---|---|---|---|---|---|---|---|
| | cGAN | wGAN | GRU | cGAN | wGAN | GRU | cGAN | wGAN | GRU |
| 24 | **0.206** | 0.226 | 0.209 | 0.086 | 0.097 | **0.081** | 0.241 | 0.297 | **0.199** |
| 48 | **0.175** | 0.208 | 0.180 | 0.078 | 0.084 | **0.075** | 0.191 | 0.158 | **0.083** |
| 96 | 0.165 | **0.162** | 0.232 | **0.074** | 0.083 | 0.085 | 0.223 | **0.191** | 0.255 |
| 120 | 0.138 | **0.135** | 0.157 | 0.085 | **0.080** | 0.081 | 0.227 | 0.201 | **0.116** |
| 168 | 0.121 | 0.118 | **0.103** | 0.075 | **0.069** | 0.072 | 0.110 | 0.119 | **0.098** |
| 240 | 0.185 | **0.155** | 0.208 | 0.040 | **0.038** | 0.068 | 0.092 | **0.080** | 0.107 |
| 336 | 0.097 | 0.108 | **0.083** | 0.035 | 0.036 | **0.033** | 0.080 | **0.070** | 0.084 |

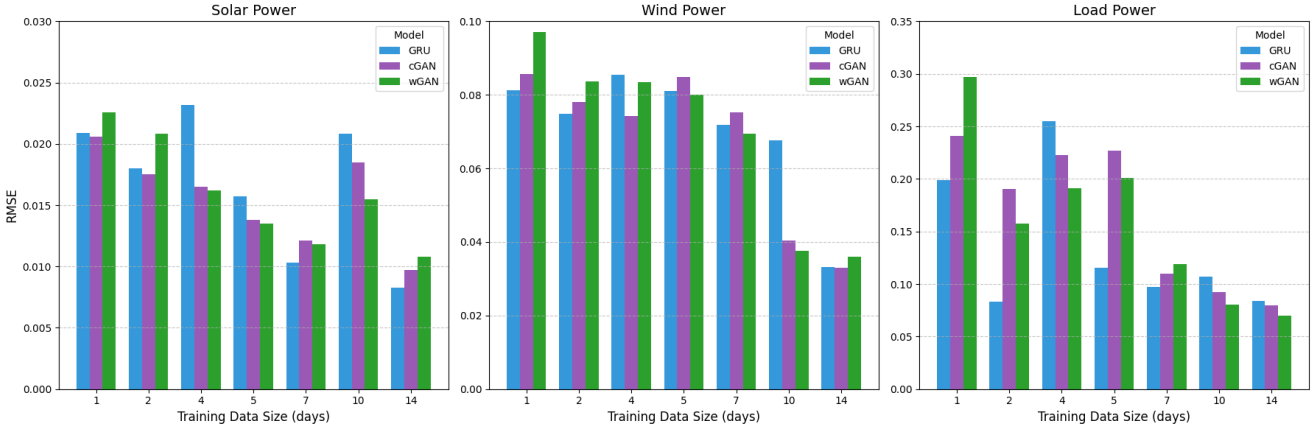



Figure 11: Comparison of RMSE for Case 2 with different training data sizes for the power grid dataset

importance of specific features like solar zenith angle for solar power predictions and wind speed and dew point for wind power predictions. GRU achieves the lowest RMSE for solar power prediction. In contrast, GANs perform better than GRU in handling feature exclusions for wind power. Load power predictions are more resilient to feature exclusions compared to solar and wind power predictions.

The results found from Case 4 Scenario 2 follow a similar pattern as Scenario 1; Cases where a feature was paired with wind speed worsen the RMSE of wind power. Due to the large volume of the results of this case, the complete results for 28 different combinations of feature pairwise exclusion are reported in a spreadsheet in the **Supplementary Materials**.

Table 10: RMSE values for solar power, wind power, and load power predictions with excluded input feature for the power grid dataset

| Excluded Feature | Solar RMSE | | | Wind RMSE | | | Load RMSE | | |
|---|---|---|---|---|---|---|---|---|---|
| | cGAN | wGAN | GRU | cGAN | wGAN | GRU | cGAN | wGAN | GRU |
| DHI | 0.117 | 0.112 | **0.082** | 0.038 | **0.032** | 0.034 | 0.103 | **0.068** | 0.094 |
| DNI | 0.118 | 0.127 | **0.084** | 0.037 | **0.036** | 0.046 | 0.077 | **0.060** | 0.074 |
| GHI | 0.098 | 0.114 | **0.086** | **0.036** | **0.036** | 0.045 | 0.082 | 0.072 | **0.067** |
| Dew Point | 0.103 | 0.096 | **0.080** | 0.040 | 0.041 | **0.038** | **0.065** | 0.086 | 0.109 |
| Solar Zenith Angle | 0.224 | 0.248 | **0.089** | 0.040 | **0.036** | 0.047 | 0.092 | 0.089 | **0.068** |
| Wind Speed | **0.090** | 0.114 | 0.120 | **0.035** | 0.038 | 0.091 | 0.093 | 0.079 | **0.068** |
| Relative Humidity | **0.118** | 0.145 | 0.120 | **0.036** | 0.043 | 0.047 | **0.075** | 0.092 | 0.082 |
| Temperature | 0.114 | 0.115 | **0.078** | **0.035** | **0.035** | 0.046 | 0.073 | 0.064 | **0.063** |

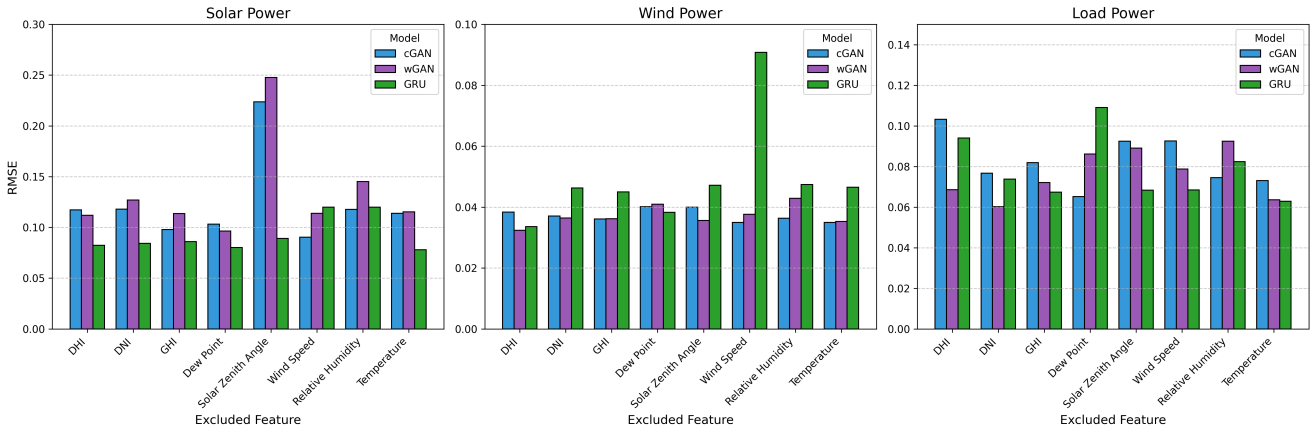



Figure 12: Comparison of RMSE in case of excluded features

One might question why advanced feature importance methods, such as SHAP (SHapley Additive exPlanations) [53, 54] or Sobol indices [55, 56], were not utilized in this analysis, given their popularity and use in previous related studies. While we have considered these approaches, we note that they are typically post hoc methods that assess feature importance based on a "well-trained" model. In contrast, our study directly evaluates the impact of feature removal on model performance, offering a more application-driven perspective. Using SHAP with an unreliable GAN or machine learning model may yield misleading feature importance rankings. Pre-training methods like linear correlation analysis, though possible, are limited by their linear assumptions and may fail to capture nonlinear relationships, leading to inaccurate conclusions. Given these considerations, we have opted for direct exclusion experiments to provide a more intuitive and reliable assessment of feature contributions.

This study demonstrates the potential and limitations of GenAI techniques for synthetic data generation and predictive modeling. While cGAN and wGAN exhibit robust performance under optimal conditions, their reliability diminishes with smaller or less representative datasets. These findings provide valuable insights for applying GenAI in engineering contexts and emphasize the need for careful dataset curation and feature selection to maximize model efficacy.

cGAN and wGAN, while not as accurate as FNN for predictive modeling as observed in the CHF case, offer the unique advantage of capturing the underlying data distribution and generating new, realistic samples. Both models excel at generating synthetic data points when a large and diverse dataset is available. wGAN, in particular, is a reliable choice for generating accurate synthetic data, even with limited feature availability or challenging conditions. wGAN consistently outperformed cGAN and GRU across most scenarios, particularly for larger datasets and longer

lookback periods. This can be attributed to WGAN's use of the Wasserstein loss, which improves the stability of the training process and better captures complex distributions in the data. cGANs exhibited moderate performance, often underperforming compared to wGAN and GRU for time-series data. This can be linked to cGAN's sensitivity to data size and training stability. The adversarial nature of cGAN training makes it prone to mode collapse and unstable gradients, especially with limited or unbalanced datasets.

The goal of this study was to evaluate the strengths, limitations, and reliability of GANs, as one of the popular generative models, in engineering and energy applications. A major challenge with GANs is their high data requirement, which can be restrictive in real-world scenarios with limited data availability. Rather than assuming GANs (or any other generative AI model) are always the best approach, we followed a systematic approach to assess their feasibility by considering factors such as available training data, the need for synthetic data, and whether using a GAN is justified under these constraints. By examining how GAN performance declines with smaller datasets and how it generalizes to new unseen scenarios, we identify the threshold for data adequacy, guiding practitioners in determining whether GANs (or generative AI) are a viable solution or if alternative methods should be considered. This evaluation helps engineering decision-makers make informed choices about the suitability of GANs and generative AI for their specific needs.

This study aimed to conduct a comprehensive investigation of GAN-based architectures for data augmentation and time-series forecasting. Rather than an extensive comparison of generative models, our focus was on evaluating different study scenarios, such as dataset size and feature availability, to better understand the applicability of GANs in engineering contexts. Given the complexity and computational demands of generative models, we centered our analysis on GANs to ensure a rigorous assessment of their performance, limitations, and practical relevance. While Variational Autoencoders (VAEs) and diffusion models present promising alternatives, each comes with distinct challenges and hyperparameter considerations that require dedicated analysis. A fair and comprehensive evaluation of these models falls beyond the scope of this study. However, we recognize their potential and encourage future research to extend our methodology to alternative generative approaches. This study provides a systematic framework and case studies that can be easily adapted to other generative models, including VAEs and diffusion models, for a broader evaluation of generative techniques.

## 5. Conclusions

In this study, we examined how well GenAI techniques, particularly GAN, perform in real-world engineering applications with a focus on nuclear energy and renewable energy applications. We focused on the data requirements of GANs when applied to CHF synthetic data generation and energy demand prediction. In addition to cGAN and wGAN, more conventional models like FNN and GRU were studied to compare the performance of GAN models. We utilized two datasets, one comprising over 20,000 real experimental CHF measurements and another comprising 3 years of minute-level power grid data.

Through three designed experiments, we explored the behavior of GANs when varying the size of the training dataset, training GANs on data from different experimental sources to generate new data from unseen experimental sources, and excluding various combinations of input parameters from the training to assess their impact on GAN's accuracy in data generation. Our findings revealed that GAN's performance decreases with smaller training dataset sizes, indicating its significant data dependency for reliable performance. Both cGAN and wGAN achieved strong performance metrics (e.g., MAPE  9% for cGAN and  7% for wGAN) with large datasets, but their accuracy deteriorated sharply with smaller datasets. For instance, MAPE for cGAN and wGAN increased to 19.53% and 57.20%, respectively, when trained with only 100 samples. When trained on data from certain experiments, cGAN performed well in predicting similar data but poorly when trained on different experiment sources. For example,

when trained on data from Smolin et al.'s experiments, cGAN performed fairly well in predicting the data from Becker et al., Kirillov et al., and Alekseev et al. but provided very poor performance when trained with Alekseev et al.'s data and tasked to predict other experiment setups. In either case, cGAN performance was lower compared to when it was trained on samples from all experiments at the same time. In analyzing feature importance, we observed that cGAN relies heavily on input parameters such as mass flux and heated length for accurate CHF generation, while parameters like diameter and pressure have less influence and inlet temperature showed a moderating effect. wGAN consistently exhibited resilience to feature exclusion, making it a robust choice for synthetic data generation under constrained feature availability. On the other hand, FNN offers consistent and reliable performance, making it a preferred choice for regression tasks where the primary goal is predictive accuracy rather than synthetic data generation.

Furthermore, we evaluated the performance of GRU, cGAN, and wGAN models for predicting solar, wind, and load power under four different conditions, including different lookback hours, training dataset sizes, seasonal variation and feature exclusion scenarios. We found that, for solar power, GRU demonstrated superior performance for most of the lookback variations. However, wGAN excelled in wind and load power predictions for most time horizons, indicating its strength in modeling long-term dynamics. While cGAN showed moderate performance, it was generally outperformed by GRU and wGAN across all power types and time horizons. While increasing the training dataset size improved the performance of all models, we observed that load power benefited the most, particularly for larger datasets. None of the tested models were successful at predicting power demands with seasonal variation. wGAN achieved around 0.32 RMSE when it was trained on a winter month and asked to predict the demand during a summer month and similar performance for the opposite. Feature exclusions considerably affected model performance, particularly for solar and wind power predictions, whereas load power was the least affected by feature exclusion.

Future research could consider probabilistic GANs by incorporating Bayesian techniques into the generator and discriminator. This adaptation aims to enable GANs to account for measured uncertainty in the data and generate data within acceptable measurement uncertainty bounds provided in the dataset, which may alleviate some of their generalization limitations observed in this work. In addition, other generative models not explored in this study, such as diffusion models or variational autoencoders, might be subjected to a similar assessment on similar or other datasets to determine their effectiveness in real-world energy applications.

### Data Availability

Data will be available upon request from the corresponding author.

## ACKNOWLEDGMENTS

The first author would like to thank Farah Alsafadi from North Carolina State University for the valuable discussions about generative models at the beginning of this project. This work was supported through Idaho National Laboratory's Laboratory Directed Research and Development (LDRD) Program Award Number (24A1081-116FP) under Department of Energy Idaho Operations Office contract no. DE-AC07-05ID14517. The authors also acknowledge the use of Idaho National Laboratory's high performance computing (HPC), for providing computational resources, which significantly contributed to the modeling and analysis presented in this work. Additionally, it was partially funded by the AI Initiative as part of the Laboratory Directed Research and Development Program of ORNL, which is managed by UT-Battelle LLC for the US Department of Energy under contract DE-AC05-00OR22725.

**CRediT Author Statement**

- **Umme Mahbuba Nabila**: Conceptualization, Methodology, Software, Validation, Formal Analysis, Visualization, Investigation, Data Curation, Writing - Original Draft.
- **Linyu Lin**: Conceptualization, Methodology, Data Curation, Funding Acquisition, Supervision, Project Administration, Writing - Review and Edit.
- **Xingang Zhao**: Conceptualization, Methodology, Data Curation, Funding Acquisition, Supervision, Project Administration, Writing - Review and Edit.
- **William L. Gurecky**: Conceptualization, Methodology, Data Curation, Funding Acquisition, Supervision, Project Administration, Writing - Review and Edit.
- **Pradeep Ramuhalli**: Conceptualization, Methodology, Data Curation, Funding Acquisition, Supervision, Project Administration, Writing - Review and Edit.
- **Majdi I. Radaideh**: Conceptualization, Methodology, Data Curation, Funding Acquisition, Supervision, Project Administration, Writing - Review and Edit.

**References**


[1] A. Pannu, Artificial intelligence and its application in different areas, Artificial Intelligence 4 (10) (2015) 79–84.

[2] M. I. Radaideh, K. Du, P. Seurin, D. Seyler, X. Gu, H. Wang, K. Shirvan, Neorl: Neuroevolution optimization with reinforcement learning—applications to carbon-free energy systems, Nuclear Engineering and Design 412 (2023) 112423.

[3] D. Kochkov, J. A. Smith, A. Alieva, Q. Wang, M. P. Brenner, S. Hoyer, Machine learning–accelerated computational fluid dynamics, Proceedings of the National Academy of Sciences 118 (21) (2021) e2101784118.

[4] M. I. Radaideh, M. I. Radaideh, T. Kozlowski, Design optimization under uncertainty of hybrid fuel cell energy systems for power generation and cooling purposes, International Journal of Hydrogen Energy 45 (3) (2020) 2224–2243.

[5] T. P. Carvalho, F. A. Soares, R. Vita, R. d. P. Francisco, J. P. Basto, S. G. Alcalá, A systematic literature review of machine learning methods applied to predictive maintenance, Computers & Industrial Engineering 137 (2019) 106024.

[6] M. I. Radaideh, C. Pappas, S. Cousineau, Real electronic signal data from particle accelerator power systems for machine learning anomaly detection, Data in Brief 43 (2022) 108473.

[7] C. Shorten, T. M. Khoshgoftaar, B. Furht, Text data augmentation for deep learning, Journal of big Data 8 (1) (2021) 101.

[8] K. M. Rashid, J. Louis, Times-series data augmentation and deep learning for construction equipment activity recognition, Advanced Engineering Informatics 42 (2019) 100944.

[9] C. A. Escobar, R. Morales-Menendez, Machine learning techniques for quality control in high conformance manufacturing environment, Advances in Mechanical Engineering 10 (2) (2018) 1687814018755519.

[10] M. Radaideh, C. Pappas, P. Ramuhalli, S. Cousineau, Application of convolutional and feedforward neural networks for fault detection in particle accelerator power systems, Tech. rep., Oak Ridge National Laboratory (ORNL), Oak Ridge, TN (United States) (2022).

[11] M. F. I. Sumon, M. Osiujjaman, M. A. Khan, A. Rahman, M. K. Uddin, L. Pant, P. Debnath, Environmental and socio-economic impact assessment of renewable energy using machine learning models, Journal of Economics, Finance and Accounting Studies 6 (5) (2024) 112–122.

[12] L. Alzubaidi, J. Bai, A. Al-Sabaawi, J. Santamaría, A. Albahri, B. S. N. Al-dabbagh, M. A. Fadhel, M. Manoufali, J. Zhang, A. H. Al-Timemy, et al., A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications, Journal of Big Data 10 (1) (2023) 46.

[13] Z. Xu, J. H. Saleh, Machine learning for reliability engineering and safety applications: Review of current status and future opportunities, Reliability Engineering & System Safety 211 (2021) 107530.

[14] M. I. Radaideh, T. Kozlowski, Combining simulations and data with deep learning and uncertainty quantification for advanced energy modeling, International Journal of Energy Research 43 (14) (2019) 7866–7890.

[15] M. I. Radaideh, T. Kozlowski, Surrogate modeling of advanced computer simulations using deep gaussian processes, Reliability Engineering & System Safety 195 (2020) 106731.

[16] A. Bandi, P. V. S. R. Adapa, Y. E. V. P. K. Kuchi, The power of generative ai: A review of requirements, models, input–output formats, evaluation metrics, and challenges, Future Internet 15 (8) (2023) 260.

[17] C. Tang, C. Yu, Y. Gao, J. Chen, J. Yang, J. Lang, C. Liu, L. Zhong, Z. He, J. Lv, Deep learning in nuclear industry: A survey, Big Data Mining and Analytics 5 (2) (2022) 140–160.

[18] V. Joynt, J. Cooper, N. Bhargava, K. Vu, O. H. Kwon, T. R. Allen, A. Verma, M. I. Radaideh, A comparative analysis of text-to-image generative ai models in scientific contexts: A case study on nuclear power, arXiv preprint arXiv:2312.01180.

[19] V. L. T. De Souza, B. A. D. Marques, H. C. Batagelo, J. P. Gois, A review on generative adversarial networks for image generation, Computers & Graphics 114 (2023) 13–25.

[20] S. R. Gayam, Enhancing creative industries with generative ai: Techniques for music composition, art generation, and interactive media, Journal of Machine Learning in Pharmaceutical Research 3 (1) (2023) 54–88.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27.

[22] J. Gui, Z. Sun, Y. Wen, D. Tao, J. Ye, A review on generative adversarial networks: Algorithms, theory, and applications, IEEE transactions on knowledge and data engineering 35 (4) (2021) 3313–3332.

[23] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114.

[24] D. Rezende, S. Mohamed, Variational inference with normalizing flows, in: International conference on machine learning, PMLR, 2015, pp. 1530–1538.

[25] I. Kobyzev, S. J. Prince, M. A. Brubaker, Normalizing flows: An introduction and review of current methods, IEEE transactions on pattern analysis and machine intelligence 43 (11) (2020) 3964–3979.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30.

[27] D. Saxena, J. Cao, Generative adversarial networks (gans) challenges, solutions, and future directions, ACM Computing Surveys (CSUR) 54 (3) (2021) 1–42.

[28] A. Jabbar, X. Li, B. Omar, A survey on generative adversarial networks: Variants, applications, and training, ACM Computing Surveys (CSUR) 54 (8) (2021) 1–49.

[29] J. Wang, B. Han, H. Bao, M. Wang, Z. Chu, Y. Shen, Data augment method for machine fault diagnosis using conditional generative adversarial networks, Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering 234 (12) (2020) 2719–2727.

[30] E. Branikas, P. Murray, G. West, A novel data augmentation method for improved visual crack detection using generative adversarial networks, IEEE Access 11 (2023) 22051–22059.

[31] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: International conference on machine learning, PMLR, 2017, pp. 214–223.

[32] K. E. Smith, A. O. Smith, Conditional gan for timeseries generation, arXiv preprint arXiv:2006.16477.

[33] X. Gu, K. See, Y. Liu, B. Arshad, L. Zhao, Y. Wang, A time-series wasserstein gan method for state-of-charge estimation of lithium-ion batteries, Journal of Power Sources 581 (2023) 233472.

[34] S. Festag, J. Denzler, C. Spreckelsen, Generative adversarial networks for biomedical time series forecasting and imputation, Journal of Biomedical Informatics 129 (2022) 104058.

[35] S. Qi, J. Chen, P. Chen, P. Wen, W. Shan, L. Xiong, An effective wgan-based anomaly detection model for iot multivariate time series, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2023, pp. 80–91.

[36] H. Ni, L. Szpruch, M. Sabate-Vidales, B. Xiao, M. Wiese, S. Liao, Sig-wasserstein gans for time series generation, in: Proceedings of the Second ACM International Conference on AI in Finance, 2021, pp. 1–8.

[37] Y. Gorishniy, I. Rubachev, V. Khrulkov, A. Babenko, Revisiting deep learning models for tabular data, Advances in Neural Information Processing Systems 34 (2021) 18932–18943.

[38] M. A. Bansal, D. R. Sharma, D. M. Kathuria, A systematic review on data scarcity problem in deep learning: solution and applications, ACM Computing Surveys (CSUR) 54 (10s) (2022) 1–29.

[39] M. I. Radaideh, D. Price, T. Kozlowski, Criticality and uncertainty assessment of assembly misloading in bwr transportation cask, Annals of Nuclear Energy 113 (2018) 1–14.

[40] D. Price, M. I. Radaideh, D. O'Grady, T. Kozlowski, Advanced bwr criticality safety part ii: Cask criticality, burnup credit, sensitivity, and uncertainty analyses, Progress in Nuclear Energy 115 (2019) 126–139.

[41] J. M. Le Corre, G. Delipei, X. Wu, X. Zhao, Benchmark on artificial intelligence and machine learning for scientific computing in nuclear engineering. phase 1: Critical heat flux exercise specifications, NEA Working Papers.

[42] X. Zheng, N. Xu, L. Trinh, D. Wu, T. Huang, S. Sivaranjani, Y. Liu, L. Xie, A multi-scale time-series dataset with benchmark for machine learning in decarbonized energy grids, Scientific Data 9 (1) (2022) 359.

[43] D. Groeneveld, Critical heat flux data used to generate the 2006 groeneveld lookup tables, Tech. rep., tech. rep., United States Nuclear Regulatory Commission (2019).

[44] M. Mirza, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784.

[45] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555.

[46] D. Svozil, V. Kvasnicka, J. Pospichal, Introduction to multi-layer feed-forward neural networks, Chemometrics and intelligent laboratory systems 39 (1) (1997) 43–62.

[47] R. Z. Khalid, A. Ullah, A. Khan, M. H. Al-Dahhan, M. H. Inayat, Dependence of critical heat flux in vertical flow systems on dimensional and dimensionless parameters using machine learning, International Journal of Heat and Mass Transfer 225 (2024) 125441.

[48] G. Zhu, Z. Wang, H. Xie, Prediction of critical heat flux in non-uniform heated rod bundle based on modified chf empirical correlations, Nuclear Engineering and Design 417 (2024) 112875.

[49] J. Nowotarski, B. Liu, R. Weron, T. Hong, Improving short term load forecast accuracy via combining sister forecasts, Energy 98 (2016) 40–49.

[50] F. M. Butt, L. Hussain, S. H. M. Jafri, H. M. Alshahrani, F. N. Al-Wesabi, K. J. Lone, E. M. T. E. Din, M. A. Duhayyim, Intelligence based accurate medium and long term load forecasting system, Applied Artificial Intelligence 36 (1) (2022) 2088452.

[51] S. M. Hasanat, K. Ullah, H. Yousaf, K. Munir, S. Abid, S. A. S. Bokhari, M. M. Aziz, S. F. M. Naqvi, Z. Ullah, Enhancing short-term load forecasting with a cnn-gru hybrid model: A comparative analysis, IEEE Access.

[52] R. K. Agrawal, F. Muchahary, M. M. Tripathi, Long term load forecasting with hourly predictions based on long-short-term-memory networks, in: 2018 IEEE Texas power and energy conference (TPEC), IEEE, 2018, pp. 1–6.

[53] F. Prendin, J. Pavan, G. Cappon, S. Del Favero, G. Sparacino, A. Facchinetti, The importance of interpreting machine learning models for blood glucose prediction in diabetes: an analysis using shap, Scientific reports 13 (1) (2023) 16865.

[54] M. I. Radaideh, S. Surani, D. O'Grady, T. Kozlowski, Shapley effect application for variance-based sensitivity analysis of the few-group cross-sections, Annals of Nuclear Energy 129 (2019) 264–279.

[55] D. Efimov, H. Sulieman, Sobol sensitivity: a strategy for feature selection, in: International Conference on Mathematics and Statistics, Springer, 2015, pp. 57–75.

[56] M. I. Radaideh, T. Kozlowski, Analyzing nuclear reactor simulation data and uncertainty with the group method of data handling, Nuclear Engineering and Technology 52 (2) (2020) 287–295.