sustainability

Article

A Scenario Generation Method for Typical Operations of Power Systems with PV Integration Considering Weather Factors

Xinghua Wang, Xixian Liu *, Fucheng Zhong, Zilv Li, Kaiguo Xuan and Zhuoli Zhao

Department of Electrical Engineering, School of Automation, Guangdong University of Technology, Guangzhou 510006, China; xinghua.wang@gdut.edu.cn (X.W.); 2112104009@mail2.gdut.edu.cn (F.Z.); 2112204510@mail2.gdut.edu.cn (Z.L.); 2112204545@mail2.gdut.edu.cn (K.X.); zhuoli.zhao@gdut.edu.cn (Z.Z.)

* Correspondence: 2112204511@mail2.gdut.edu.cn

Abstract: Under the background of large-scale PV (photovoltaic) integration, generating typical operation scenarios of power systems is of great significance for studying system planning operation and electricity markets. Since the uncertainty of PV output and system load is driven by weather factors to some extent, using PV output, system load, and weather data can allow constructing scenarios more accurately. In this study, we used a TimeGAN (time-series generative adversarial network) based on LSTM (long short-term memory) to generate PV output, system load, and weather data. After classifying the generated data using the k-means algorithm, we associated PV output scenarios and load scenarios using the FP-growth algorithm (an association rule mining algorithm), which effectively generated typical scenarios with weather correlations. In this case study, it can be seen that TimeGAN, unlike other GANs, could capture the temporal features of time-series data and performed better than the other examined GANs. The finally generated typical scenario sets also showed interpretable weather correlations.

Keywords: deep learning; generative adversarial networks (GAN); time series; photovoltaic (PV); scenario generation; k-means; clustering; FP-growth; association rule

check for updates

Citation: Wang, X.; Liu, X.; Zhong, F.; Li, Z.; Xuan, K.; Zhao, Z. A Scenario Generation Method for Typical Operations of Power Systems with PV Integration Considering Weather Factors. *Sustainability* **2023**, *15*, 15007. https://doi.org/10.3390/su152015007

Academic Editors: Liang-Ruei Chen, Nien-Che Yang and Wei-Tzer Huang

Received: 9 September 2023
Revised: 11 October 2023
Accepted: 13 October 2023
Published: 18 October 2023

1. Introduction

With the intensification of energy and environmental issues, the human society has begun to transform the energy structure towards sustainability, which has led to the integration of a large amount of renewable energy and new types of loads into distribution networks [1]. These changes pose new challenges to the operation and dispatch of power systems. The current related research mainly focuses on aspects like robust optimization [2], stochastic optimization [3], and distributionally robust optimization [4]. Stochastic optimization relies heavily on a fixed probability distribution. In the solution process, it often requires the use of finite discrete scenarios to approximate the probability model. In other words, stochastic optimization formulates strategies based on a set of typical operating scenarios; so, the generation of typical operating scenarios is a critical issue for stochastic optimization [5]. To scientifically plan the operation of power systems including those for new energy generation like photovoltaics, an accurate scenario analysis is a prerequisite. Since photovoltaic power generation is highly influenced by environmental factors, its stochastic and fluctuating characteristics introduce high uncertainty into the system [6]. The load scenarios are also diverse [6], not only closely related to people's production and life cycles, but also constrained by factors such as temperature, humidity, precipitation, and holidays. These factors lead to diverse PV and load scenarios with a large amount of conflicts and overlaps, making it critical to accurately construct typical operating scenarios of a power system, which plays an important role in the planning, operation, and economics of high photovoltaic penetration distribution networks. However, there has been limited research in the field of scenario generation on utilizing weather factors to correlate PV output and load, despite the real-world impact of weather on PV and load [7]. Therefore, it is imperative to conduct studies in this area.

Currently, the common scenario generation methods mainly include statistical methods and deep learning methods. The statistical methods consist of probabilistic modeling methods like Markov chains [8], Gaussian processes [9], autoregressive moving average models [10], and copula functions [11], as well as scenario clustering methods [12]. In the above methods, probabilistic modeling methods regard scenario data as random processes, essentially fitting the joint probability density distribution of the random process through probability models but rely heavily on prior knowledge and reveal limitations in the face of increasingly high-dimensional data [13]. The scenario clustering methods only cluster PV and load data and are unable to accurately describe the characteristics of PV output and load data, which makes it difficult to ensure the accuracy and diversity of the generated operational scenarios. The deep learning methods include variational autoencoders [14], generative adversarial networks (GANs) [15], and deep belief networks [16]. In theory, the deep learning methods can approximate arbitrary functions and effectively fit high-dimensional functional relationships in data. Scenario generation relies critically on fitting the training data; so, deep learning techniques have been widely applied and studied. In recent years, GANs have been extensively researched by scholars. In related research, convolutional neural networks (CNNs) were utilized in the generator and discriminator of GANs in [17,18]. One of these two studies constructed a conditional generative adversarial network (CGAN) with load factors as the condition, but only included a single load type. The other optimized the network structure for day-ahead scenario generation by using ReLU activation functions in the output layers of the generator and discriminator and removing the normalization layers and achieved good performance. References [19,20] introduced gradient penalties and the Wasserstein distance, effectively improving model generalization, slow convergence, and difficulty in convergence, but convergence issues may still exist under certain specific inputs. Reference [21] used a CGAN based on deep convolution (DCCGAN) to learn data from existing renewable energy power stations near a new plant, generating better scenario data for the new plant compared to CGAN, but the deep convolutional neural network structure required parameter initialization tuning based on the dataset size.

With increasing integration scales of PV power, the distribution networks face operational scenarios involving both PV and load, with coupling relationships among PV output, meteorological data, and load and mutual influence between data, large data volumes, and high dimensions, imposing high requirements on the generation model. This requires a GAN model that can generate data highly consistent with the original time-series characteristics and solve the current difficulties in GAN training and sensitivity to parameters. The aforementioned methods have their own limitations. Introducing Wasserstein distance or using a CNN-based CGAN in GANs may allow the generated data to capture the overall feature distribution of the original data. However, this remain insufficient to grasp the intricate internal details within time-series data. Therefore, it is imperative to employ data generation algorithms that can comprehensively reflect both the overall statistics and the internal structures of the data.

A good time series generation model should generate sequences that conform to the original relationships between variables. In other words, for a time series $X_{1:t} = (X_1, \ldots, X_t)$ (T is the length of the nth time series), besides capturing the overall feature distribution, the model should also accurately capture the complex latent relationships $p\ (X_t|X_{1:t-1})$ between time steps like autoregressive models [22]. The classical GANs consist of a generator and a discriminator, each with a loss function, the generator aiming to minimize the loss, and the discriminator to maximize it. The model essentially leverages the confrontation between generator and discriminator to achieve convergence after training, generating new data conforming to the original data distribution. A study [22] pointed out that time-series data should contain static features and temporal features, while classical GANs focus on describing the overall probability distribution of time series without learning the conditional probability distribution at each time step. TimeGAN combines autoregressive models and GANs [22]. In addition to the unsupervised confrontation loss between real and generated sequences of the traditional GANs, it also introduces supervised loss using the original data; training both losses enables the generated sequences to fully reconstruct the static and temporal features of time-series data.

This paper proposes a method to generate typical operation scenarios of power systems with photovoltaic integration based on weather factors. The novelty of this work lies in utilizing TimeGAN to capture temporal features of time-series data and incorporating weather factors to establish associations between PV, load, and weather scenarios. The objectives were to (1). accurately generate PV, load, and weather data scenarios reflecting both static and temporal characteristics via TimeGAN; (2). discover association rules between PV, load, and weather factors through data mining; (3). match PV and load scenarios based on weather conditions to construct typical operating scenarios. The specific process of the method is introduced here. Firstly, the TimeGAN model was trained with historical PV output data and load data containing weather information to learn the distributions of the original sequences and generate sufficient time-series data. Then, the k-means algorithm was used to cluster the generated data and obtain classifications of PV and load scenarios with weather information separately. Furthermore, the FP-growth algorithm was utilized to mine frequent items from the PV and weather data to obtain association rules for PV scenarios under different confidence levels. Based on the weather factors in these association rules, the load scenarios were matched with the PV scenarios to generate typical PV–load operational scenarios for different confidence levels. Finally, the accuracy of the generated data was evaluated by comparing with the original data and the data generated by TimeGAN. Case results demonstrated that the TimeGAN model used in this paper can precisely and effectively generate time-series data compared to the Wasserstein generative adversarial network with gradient penalty (WGAN-GP), WGAN, DCCGAN, and GAN. And the proposed typical scenario generation method can successfully generate typical scenarios and discover association rules between PV output scenarios and load scenarios.

Here, we define the data collected from 0 to 23 h in a day as one scenario. From the collection of multi-source data to the generation of typical scenarios with weather correlations, the whole system consisted of five modules: (1) generation of PV dataset and load dataset with weather data; (2) time-series data augmentation; (3) scenario classification; (4) weather association rules mining for PV scenarios; (5) correlation of PV and load scenarios based on weather factors. The system framework is formally presented in Figure 1.

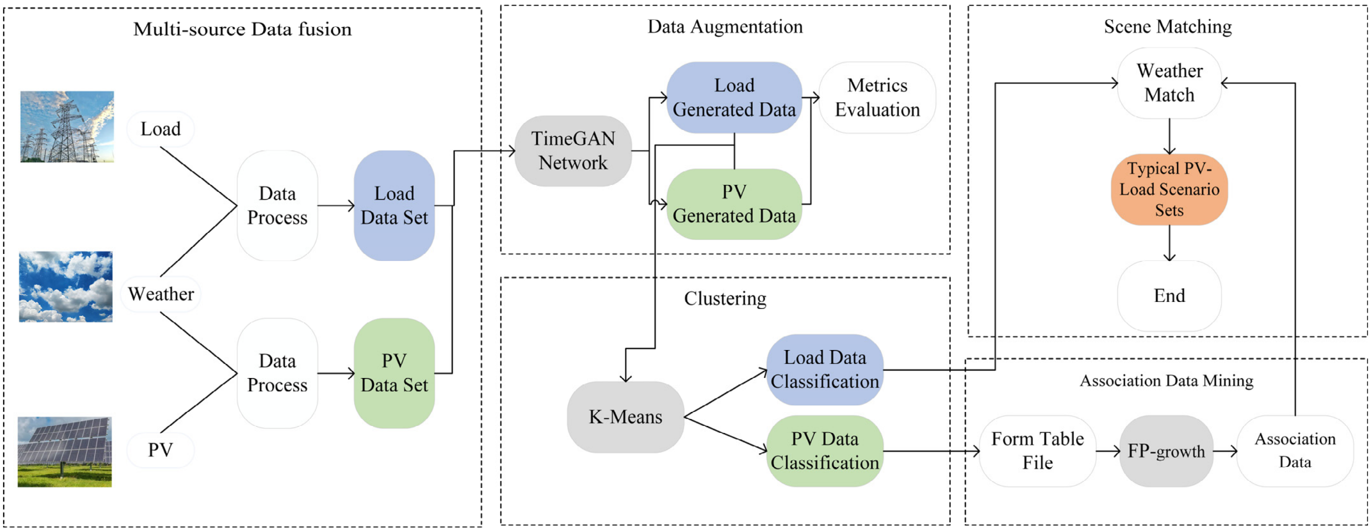


Figure 1. System framework.

The structure of this article is organized as follows:

Section 1 first introduces the background of renewable energy integration and the resulting challenges in power system operation and planning. It then analyzes the significance of generating typical operating scenarios for optimization and economics studies; the current research methods and their limitations are reviewed. The proposed method and innovations are outlined at the end.

Section 2 elaborates the detailed process and algorithms utilized in this work. It starts with the TimeGAN model, explains its architecture, objectives, and training process. Then k-means clustering and the FP-growth association rule mining algorithms are presented; evaluation metrics consisting of Wasserstein distance, MMD, ACF, and PACF are also introduced.

Section 3 presents the experimental case study results. Data visualization using t-SNE verifies the similarity between generated and original data distributions. TimeGAN is evaluated and compared with other GAN models. Typical scenario generation results are provided, with weather correlations interpreted. The generated scenarios are verified against historical data.

Finally, Section 4 concludes this work, summarizes the main contributions, and discusses aspects that can be improved in future research.

2. Materials and Methods

Here, we present the proposed method. All the algorithms utilized, starting from data input, are introduced first. Then, the evaluation metrics corresponding to the characteristics of the algorithms are described.

2.1. TimeGAN Network

In PV-integrated power systems, the time-series data directly constituting operational scenarios can be divided into two types, i.e., PV output and load, each containing different static and temporal features. For PV, the output sizes in different regions and environments are distinct static features, while seasonal, weather, and time factors constitute influencing temporal features. The load is closely related to geographical location and people's living habits and economic levels, which are static features, and varies with time, workdays/holidays, and weather changes, which are temporal influencing factors.

To fully capture the static and temporal features of time-series data, TimeGAN learns the conditional probability distribution at each time step in addition to the overall probability distribution captured by the classical GAN contest structure. Compared to merely differentiating between real and generated data, the original input data contain more exploitable information. Therefore, TimeGAN incorporates the original data as supervision to train on supervised loss and learn the pointwise conditional probability distribution. Meanwhile, deep LSTM networks are utilized to construct the entire TimeGAN model.

The data used to train TimeGAN are time-series data. Let there be N time series in total. To represent the static and temporal features of the original data, M denotes the set of all vectors in the temporal feature space, and S denotes the set of all vectors in the static feature space. To indicate the pointwise relationships between time steps in the temporal features, $M_{1:W_n}$ is used (W is the length of the nth time series), and the joint distribution of instances in M and S is denoted as $p \rightarrow \left(S, M_{1:W_n}\right)$, which leads to the training set $D = \sum_{n=1}^{N} \left(S_n, M_{n,1:W_n}\right)$. The goal of TimeGAN is to learn a distribution $\overline{p}$ through the training set D that best approximates the true joint distribution p.

2.1.1. Objectives

As described above, TimeGAN aims to best approximate the distribution p through two objectives. The first one is global, corresponding to the static features of the data:

$$\binom{min}{\overline{p}} D\left(p\left(S, M_{1:W_n}\right) \| \overline{p}\left(S, M_{1:W_n}\right)\right) \tag{1}$$

where D denotes the distance between the two distributions. The second one is local, corresponding to the step-wise temporal features:

$$\binom{min}{\overline{p}} D(p(M_t|S, M_{1:W_n-1})\|\overline{p}(M_t|S, M_{1:W_n-1})) \tag{2}$$

2.1.2. TimeGAN Architecture

TimeGAN consists of four components: a generator, a discriminator, an embedding function, and a recovery function; its architecture is shown in Figure 2.

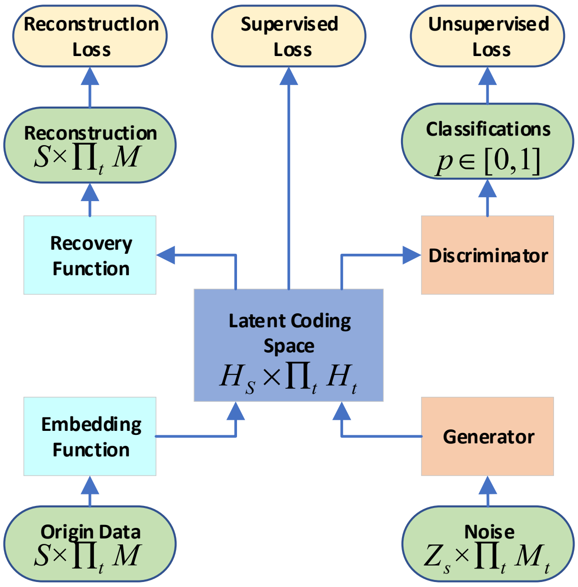

Reconstruction Loss
Supervised Loss
Unsupervised Loss
Reconstruction $S \times \prod_t M$
Classifications $p \in [0,1]$
Recovery Function
Discriminator
Latent Coding Space $H_S \times \prod_t H_t$
Embedding Function
Generator
Origin Data $S \times \prod_t M$
Noise $Z_s \times \prod_t M_t$

Figure 2. TimeGAN architecture.

The introduction of embedding and recovery functions is based on the fact that high-dimensional, complex time dynamics are often driven by lower-dimensional, simpler key influencing factors. Therefore, the embedding and recovery functions provide a low-dimensional latent space for the network to learn critical influencing factors.

The objective of the embedding function is to reduce the dimensionality of the original time series to improve the learning efficiency of the model. The embedding function implements the processing of static and temporal features recursively. For the static features, it projects them into a low-dimensional space. For the temporal features, it mines the relationships between time steps and projects into a low-dimensional space [22]:

$$h_S = e_S\ (s) \tag{3}$$

$$h_t = e_m\ (h_S, h_{t-1}, M_t) \tag{4}$$

where e_S and e_m represent the processing of static and temporal features by the embedding function, respectively, h_S denotes the low-dimensional static feature after dimensionality reduction by the embedding function, h_t denotes the low-dimensional temporal feature at time t after reduction, M_t represents the high-dimensional temporal feature at time t.

The recovery function aims to reconstruct the original high-dimensional vectors from the low-dimensional ones after dimensionality reduction, with the process [22]:

$$\tilde{S} = r_S\ (h_S) \tag{5}$$

$$\tilde{M}_t = r_M\ (h_t) \tag{6}$$

where r_S and r_M represent the processing of low-dimensional static and temporal features by the recovery function, respectively. $\tilde{S}$ denotes the reconstructed static feature, $\tilde{M}_t$ denotes the reconstructed temporal feature at time t.

The generator first randomly samples vectors from the known distribution of static and temporal feature vector spaces as input into the low-dimensional latent space, with the process [22]:

$$\overset{\frown}{h}_S = g_S\ (z_S) \tag{7}$$

$$\overset{\frown}{h}_t = g_M\ (\overset{\frown}{h}_S, \overset{\frown}{h}_{t-1}, z_t) \tag{8}$$

where g_S and g_M are the generation networks for static and temporal features, z_S and z_t are the sampling from static and temporal feature vector spaces, $\overset{\frown}{h}_S$ and $\overset{\frown}{h}_t$ are the generated static and temporal feature vector collections.

The outputs of the generator and embedding function after joint encoding are input into the discriminator, which will judge real or fake data. The process is [22]:

$$\tilde{y}_S = d_S\ (\tilde{h}_S) \tag{9}$$

$$\tilde{y}_t = d_M\ (\overleftarrow{u}_t, \overrightarrow{u}_t) \tag{10}$$

where $\tilde{y}_S$ and $\tilde{y}_t$ are the discrimination results for the static and temporal features of the input data, d_S and d_M are the discrimination networks for static and temporal features, using bidirectional recurrent networks with feedforward output layers, $\overleftarrow{u}_t$, $\overrightarrow{u}_t$ are the forward and backward hidden state sequences.

Training Losses

The goal of the embedding and recovery functions is to generate the low-dimensional latent space and reconstruct the original high-dimensional feature space as precisely as possible. Therefore, the first loss is introduced [22]:

$$L_{e-r} = E_{S,M_{1:T\sim p}}\left[\left\|S - \tilde{S}\right\|_2 + \sum_t \left\|M_t - \tilde{M}_t\right\|_2\right] \tag{11}$$

where S and M_t denote the static and temporal feature spaces, $\tilde{S}$ and $\tilde{M}_t$ are defined as shown in Equations (5) and (6). For the generator and discriminator of the model, there is a classical confrontation, with the second loss [22]:

$$L_{g-d} = E_{S,M_{1:T\sim p}}\left[log y_S + \sum_t log y_t\right] + E_{S,M_{1;T\sim \overline{p}}}\left[log\ (1 - y_S) + \sum_t\ (1 - log y_t)\right] \tag{12}$$

where y_S and y_t are the discrimination results of the original data for static and temporal features, y_S and y_t are the discrimination results for the generated data.

The introduction of loss L_{g-d} enables the model to focus on describing the overall probability distribution of time series without learning the conditional probability distribution at each time step. Therefore, the third loss L_S is introduced to achieve this [22]:

$$L_S = E_{S,M_{1:T\sim p}}\left[\sum_t \|h_t - g_M\ (h_S, h_{t-1}, z_t)\|_2\right] \tag{13}$$

2.1.3. TimeGAN Training

The training process of TimeGAN is illustrated using the PV dataset with weather information.

To begin with, Min–Max scaling was employed for data preprocessing to rescale the raw training data into the range [0, 1]. The normalized dataset was thus obtained.

In the model training phase, the objectives of the embedding and recovery networks are to provide a reversible feature-latent space mapping. LSTM networks are implemented in the generator, discriminator, embedding network, and recovery network. The generative network consists of num_layers of unidirectional LSTM layers, each LSTM layer having a hidden dimension (hidden_dim), where hidden_dim denotes the size of the hidden layer in the LSTM. The discriminator is composed of one bidirectional LSTM layer and one fully connected layer. The embedding and recovery networks share an identical architecture with the generator. The training iteration involves:

(1) Separately training the embedding and reconstruction networks, extracting batch_size groups of (max_seq_len,4) data from the raw data for training at each iteration.

(2) Training the generator. At each iteration, batch_size groups of (max_seq_len,4) data from both raw data and random noise are extracted for supervised training.

(3) Joint training, training the generator, discriminator, and embedding–recovery networks alternately. Batch_size groups of (max_seq_len,4) data from the raw data and random noise are extracted at each iteration.

The above iterations persist until the predefined number of iterations is fulfilled.

In the model generation stage, Gaussian random noise is fed into the generative network to produce generated samples of size (8760,4).

2.2. K-Means Clustering

The purpose of using clustering algorithms in this paper was to resolve complex relationships between multi-variable objects. By clustering the data after dimensionality reduction, all factors in the PV–weather–load trio could be categorized to facilitate association rule mining using the FP-growth algorithm.

It should be pointed out here that k-means was used for simplicity; if other clustering algorithms with better performance were utilized, the final association rules and generated typical scenario sets would be even better.

K-Means Algorithm

The k-means algorithm is an unsupervised clustering algorithm that can divide data into a finite number of categories. For a dataset $X = \{x_1, x_2, \ldots, x_n\}$ (x_i is a j-dimensional vector, $x_i = \{x_{i1}, x_{i2}, \ldots, x_{ij}\}$), k-means divides X into k classes, with each data point in a class nearest to the cluster center of that class. In this paper, the k-means++ algorithm initialization was adopted instead of k-means, which selected points with larger mutual distances as initial cluster centers with higher probability compared to k-means. This modification could effectively avoid the slow convergence issue of k-means.

2.3. PV–Load Association Rule Mining Based on FP-Growth

The FP-growth algorithm, as described in [23–25], is an association rule mining algorithm that stores the data in a frequent pattern tree (FP-tree) structure composed of itemsets. FP stands for frequent pattern, which refers to the frequent patterns stored in the tree as paths. Let $S = \{s_1, s_2, \ldots s_n\}$ represent the set of all distinct items in the dataset D. For any transaction T, we have $T \in S$. The support count of an itemset X $(X \subseteq S)$ is defined as the total number of transactions N.

Assume an association rule $X \to Y$ $(X \subseteq S, Y \subseteq S)$ is formed. The parameters of the FP-growth algorithm are defined as:

(1) Support, for an itemset X, refers to the probability of X occurring in the total transactions, as shown in Equation (14):

$$support(X) = P(X) \tag{14}$$

For a rule $X \rightarrow Y$, it refers to the probability of $X \cup Y$ occurring in the total transactions, as shown in Equation (15):

$$support(X \rightarrow Y) = P(X \cup Y) \tag{15}$$

In FP-growth, a minimum support is set to filter out infrequent itemsets. For example, if the minimum support is 0.05, only itemsets appearing in at least 5% of the total transactions are retained as frequent itemsets.

(2) Confidence refers to the probability of itemset Y occurring given that itemset X has occurred in the total transactions, as shown in Equation (16):

$$confidence(X \rightarrow Y) = P(X|Y) \tag{16}$$

In FP-growth, a minimum confidence is set so that rules with confidence no less than the minimum will be retained. For example, if the minimum confidence is 0.7, rules with confidence ≥ 0.7 will be kept.

In summary, association rules satisfying both minimum confidence and minimum support are called strong association rules.

In this study, the FP-growth algorithm was utilized to mine frequent itemsets of PV scenarios and discover association rules between weather scenarios and PV scenarios. To construct typical operating scenarios of PV-integrated power systems with strong descriptiveness and representativeness, typical load scenarios needed to be associated with PV scenarios based on the clustering results. After mining frequent itemsets of PV scenarios using FP-growth and deriving association rules, matching load scenarios with PV scenarios finally yielded the typical operational scenarios. The algorithm flowchart is illustrated in Figure 3.

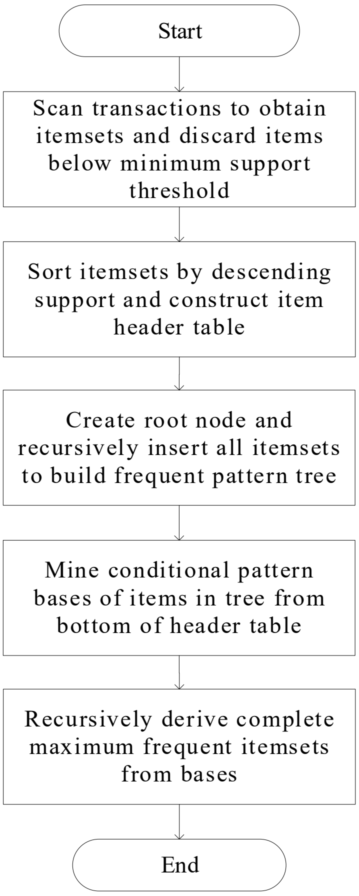

Start
Scan transactions to obtain itemsets and discard items below minimum support threshold
Sort itemsets by descending support and construct item header table
Create root node and recursively insert all itemsets to build frequent pattern tree
Mine conditional pattern bases of items in tree from bottom of header table
Recursively derive complete maximum frequent itemsets from bases
End

Figure 3. Flow chart of the FP-growth algorithm.

In this study, the generated PV output and weather data were first classified and formatted into tables suitable as input for the FP-growth algorithm. Then, frequent itemset mining was performed. With the obtained frequent itemsets, the load scenarios were matched with the PV scenarios to acquire typical PV–load scenario sets under different confidence levels.

3. Experimental Results and Related Discussion

3.1. Dataset

A case study was performed using full-year data from 2022, with an hourly collection frequency and 8760 data groups. Each data group contained PV output, weather, and load data values. The weather data included temperature, humidity, and precipitation data.

3.2. Data Preprocessing

After data cleaning, as neural network training requires data normalization, min–max normalization was applied to scale all data between 0 and 1 using Equation (17):

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{17}$$

where x' is the normalized data of the original data x, x_{min} and x_{max} are the minimum and maximum values in the original data.

3.3. Parameter Settings

Here, we provide the parameter settings for the algorithms used in this study.

3.3.1. TimeGAN

The relevant parameters of TimeGAN are presented in Table 1.

Table 1. TimeGAN parameter setting.

Parameter Meaning	Parameter	Value
Number of Layers	num_layer	3
Number of Hidden Units per Layer	hidden_dim	24
Max Sequence Length of Data	max_seq_len	24
Iterations	iterations	25,000
Batch Size	batch_size	128
Learning Rate	learing_rate	0.001
1st Moment Decay Rate of Adam Optimizer	β_1	0.9
2nd Moment Decay Rate of Adam Optimizer	β_2	0.999

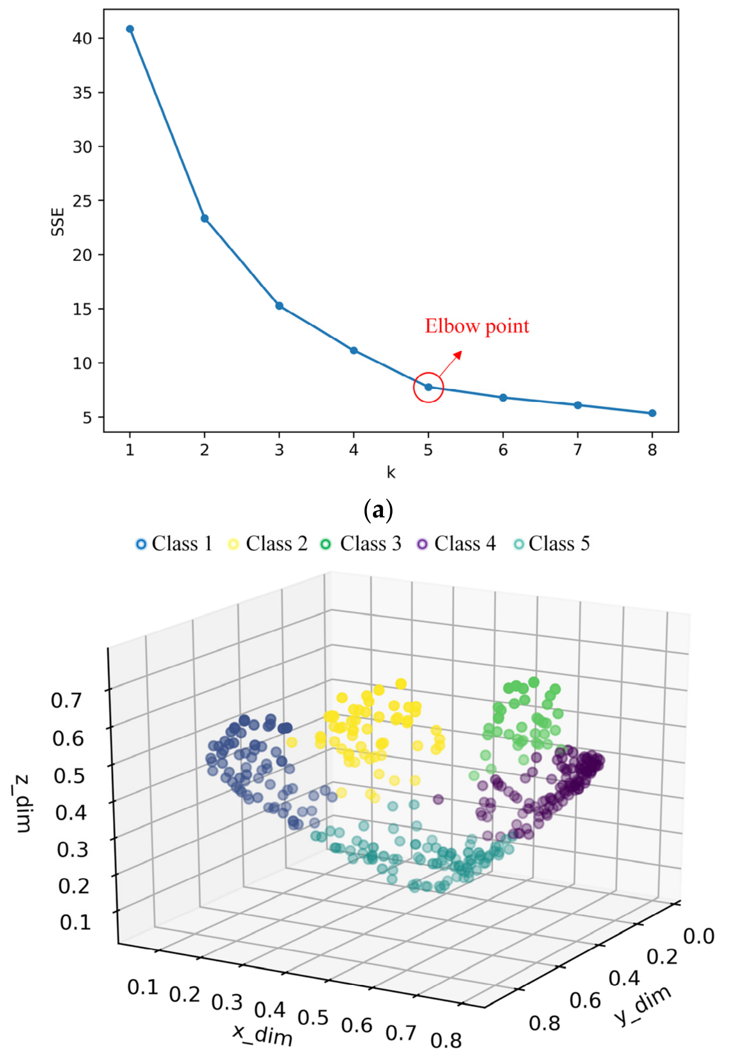
3.3.2. K-Means

The number of clusters k needed to be manually determined in k-means. The elbow method [26] can be utilized for the selection, which is based on the sum of squared errors (SSE), defined as:

$$SSE = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \tag{18}$$

where k is the number of clusters, C_i is the ith cluster, p is a sample point in C_i, m_i is the cluster center of C_i.

As k increases from 1, each cluster in the space will be divided more precisely, and SSE will decrease rapidly. When k exceeds the true number of clusters, the increasing k will no longer significantly decrease SSE, and the SSE–k curve will become flat, resembling an elbow. The turning point of the curve indicates the true number of clusters k. The result is shown in Figure 4a, and the 3D graph of the relevant clustering results is shown in Figure 4b.



(**a**)

(**b**)

Figure 4. (**a**) Selection of the k value; (**b**) k-means clustering results.

According to Figure 4, the k-means clustering parameter k was set as 5.

3.3.3. FP-Growth

The FP-growth algorithm was utilized to mine the association rules from the classified PV data with weather. The results served as the association data. FP-growth requires setting a minimum confidence and support thresholds, which impact the mining results. The parameter settings are shown in Table 2.

Table 2. FP-growth parameter setting.

Parameter Meaning	Parameter	Value
Minimum Confidence	min_con	0.75
Minimum Support	min_sup	0.06

3.3.4. Summary of the Global System Parameters

For the readability of the article, we provide the summary of the global system parameters as shown in Table 3.

Table 3. Global system parameters.

Algorithm	Parameter Meaning	Parameter	Value
TimeGAN	Number of Layers	num_layer	3
	Number of Hidden Units per Layer	hidden_dim	24
	Max Sequence Length of Data	max_seq_len	24
	Iterations	iterations	25,000
	Batch Size	batch_size	128
	Learning Rate	learing_rate	0.001
	1st Moment Decay Rate of Adam Optimizer	β_1	0.9
	2nd Moment Decay Rate of Adam Optimizer	β_2	0.999
K-Means	Optimal Number of Clusters	k	5
FP-growth	Minimum Confidence	min_con	0.75
	Minimum Support	min_sup	0.06

3.4. Discussion

Here, we proposed a method to generate typical operating scenarios for PV-integrated power systems considering weather factors. The TimeGAN model was utilized in the data generation stage to accurately capture static and temporal features of time series, learning the original data distribution space and generate a large number of scenarios, as demonstrated in Section 3.6 In addition, Section 3.7 shows the comparison of autocorrelation and partial autocorrelation coefficients between the generated and the original data, proving that the generated data learned the characteristics of the original data distribution both globally and locally.

To highlight the advantages of the algorithm, the study compared TimeGAN with four types of GANs. The comparison in Section 3.7 showed that whether evaluating each part of the generated data independently using Wasserstein distance or assessing the overall data using MMD, TimeGAN outperformed the other GANs. The excellent performance of TimeGAN is attributable to:

(1) Explicitly encouraging the model to learn the conditional probability distribution at each point of the time-series data, improving the probability distribution fitting capability of scenario generation algorithms, enabling the generated data to fully express the temporal characteristics of the original data.

(2) Adopting an LSTM network architecture, which effectively resolved gradient issues during training and fully mined temporal information over longer time periods.

For generating typical scenarios, this paper first obtained scenario classifications through clustering, then mined the association rules using the FP-growth algorithm as described in Section 2.3, and finally correlated PV and load scenarios based on the objective weather conditions in the association rules, effectively establishing relationships for the PV–weather–load trio.

The results demonstrated that the proposed method could effectively generate typical scenarios for PV-integrated power systems with interpretable weather correlations. The main contributions and significance of this work are as follows:

(1) A deep learning model TimeGAN was leveraged to generate time-series data capturing both static and temporal features of PV output, load, and weather data. This solved the problem of classical GAN models, which are unable to learn temporal relationships within time series.

(2) Weather factors were explicitly incorporated to establish associations between PV scenarios, load scenarios, and weather scenarios. This enabled interpreting the underlying meteorological conditions behind the generated typical scenarios.

(3) The proposed method reduced reliance on subjective prior knowledge during typical scenario generation by mining objective association rules between PV, load., and weather factors. This enhanced the diversity and representativeness of the generated scenarios.

(4) The generated typical scenarios can better support the optimization and planning of PV-integrated power systems by providing more accurate approximates of real-world operating conditions.

3.5. Metrics

In light of the characteristics of TimeGAN and time-series data, we incrementally validated the effectiveness of the generated data using four metrics. For static features, Wasserstein distance and MMD distance (maximum mean discrepancy) were utilized. For temporal features, ACF (autocorrelation coefficient) and PACF (partial autocorrelation coefficient) were employed.

3.5.1. Wasserstein Distance

The Wasserstein distance, also known as the Earth mover's distance (EMD), signifies the minimum cost of transporting one distribution into another. As it can effectively measure the distance between two distributions even without overlap, it has been widely applied [27]. For two different distributions P1 and P2, the Wasserstein distance is defined as

$$W\ (\mu,v)=\left(\underset{\pi\in\Pi\ (\mu,v)}{inf}\int_{\mathbb{R}\times\mathbb{R}}\|x-y\|^{p}d\pi\ (x,y)\right)^{\frac{1}{p}} \tag{19}$$

where $\Pi\ (\mu,v)$ is the set of joint distributions π on $\mathbb{R}\times\mathbb{R}$ with marginal distributions μ and v on $\Pi\ (\mu,v)$, and $\|\mathrm{x}-\mathrm{y}\|$ is the distance between the elements x and y in π. The smaller the Wasserstein distance, the closer the two distributions.

3.5.2. MMD Distance

MMD is used to measure differences between two distributions. For example, given $(x_1,x_2,\ldots,x_n)\ \sim P\ (x)$ and $(y_1,y_2,\ldots,y_n)\ \sim Q\ (x)$, MMD is defined as

$$MMD(P,Q)=\underset{\|f\|H\leq 1}{sup}\ E_P[f\ (x)]-E_q[f\ (y)] \tag{20}$$

where $E_P[f\ (x)]$ and $E_q[f\ (y)]$ represent the expectation of the set of functions $f\ (x)$ and $f\ (y)$ that maps x, y to higher order, and $\|f\|H\leq 1$ constrains the norm of the function f in the reproducing Hilbert space to be less than or equal to 1.

3.5.3. ACF and PACF

ACF $R\ (k)$ analyzes correlations between two segments of a time series lagged by k and can evaluate whether the model captures the autocorrelations within the original time-series observations. PACF $\rho\ (k)$ examines correlations between two points lagged by k and can determine if the model captures the independence between observations in the original time series. They are defined as

$$R\ (k)=\frac{E[(X_i-\mu)\ (X_{i+k}-\mu)]}{E(X_i-\mu)^2} \tag{21}$$

$$\rho\ (k)=\frac{E\left[(X_i-\hat{E}X_i)\ (X_{i-k}-\hat{E}X_{i-k})\right]}{\sqrt{E(X_i-\hat{E}X_i)^2}\sqrt{E(X_{i-k}-\hat{E}X_{i-k})^2}} \tag{22}$$

where X_i is the value at time i in the time series, k is the lag time, μ is the mean of the time series, and $\hat{E}X_i=E[X_i|X_{i-1},\ldots,X_{i-k+1}]$.

3.6. Data Visualization

Since linear dimensionality reduction struggled to extract the nonlinear intrinsic structures and patterns within the nonlinear data of weather, PV output, and power load, nonlinear dimensionality reduction was utilized here for data visualization. Figure 5 shows the comparison between the original data and the generated data embedded into a 2D plane using the t-distributed stochastic neighbor embedding (t-SNE) algorithm.

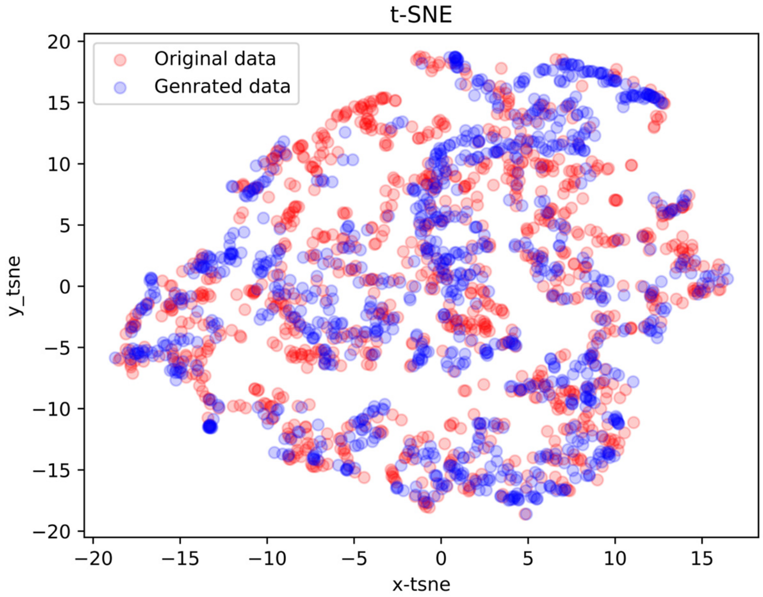


Figure 5. Visualization of t-SNE for generating data (iterations = 25,000).

From the comparison, it can be observed that the generated data remained similar to the original data in both global structure and local features, with their 2D plane distributions nearly overlapping.

3.7. Evaluation of TimeGAN-Generated Data

The Wasserstein distance was used to evaluate each part of the data generated by the TimeGAN, WGAN-GP, WGAN, DCCGAN, and GAN models, with results shown in Table 4.

Table 4. Quality evaluation of the Wasserstein distance.

Model	Temperature (°C)	Humidity (%)	Rainfall (mm/h)	PV (MW)	Load (MW)
TimeGAN	2.70	13.6	0.079	0.008	0.017
WGAN-GP	3.86	14.7	0.048	0.014	0.026
WGAN	3.43	15.1	0.060	0.018	0.019
DCCGAN	4.02	16.4	0.091	0.016	0.037
GAN	4.86	18.0	0.112	0.019	0.036

The overall data were evaluated using the MMD distance, with results reported in Table 5.

Using the PV data generated by the TimeGAN model as the evaluation object, the original data were used for comparison; the ACF and PACF in lags from 0 to 10 h are shown in Figure 6.

Table 5. Quality evaluation of the MMD distance.

Model	MMD
TimeGAN	0.5346
WGAN-GP	0.6616
WGAN	0.6934
DCCGAN	0.7364
GAN	0.8089

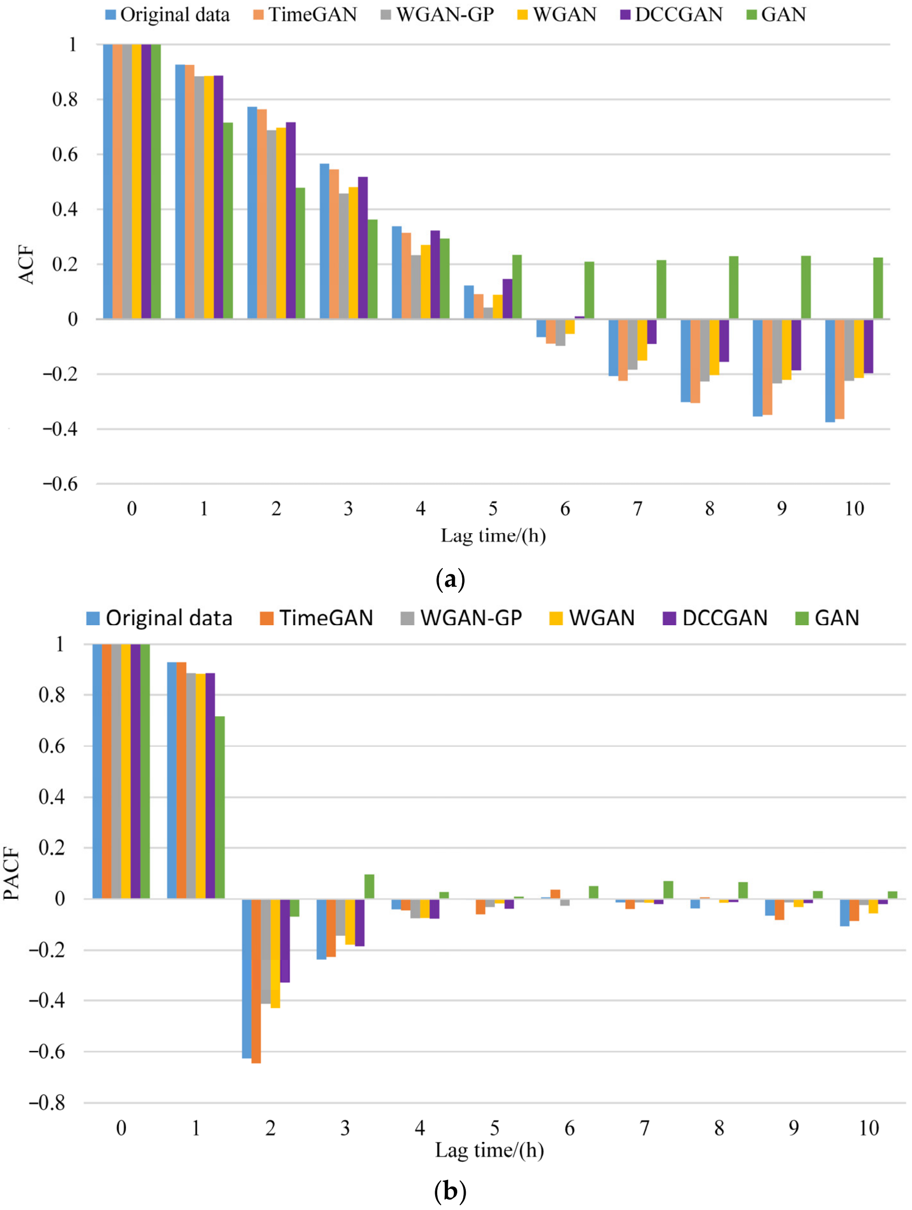


Figure 6. ACF and PACF of the PV dataset. (**a**) ACF for original and generated data; (**b**) PACF for original and generated data.

Figure 6a shows that the ACF of the PV scenarios generated by TimeGAN and the original PV scenarios basically overlapped, exhibiting positive autocorrelations within 0–5 h, which diminished as the lag increased, especially in the 1–2 h range. In contrast, the ACF of the PV scenarios generated by other GANs showed significant differences compared to the original PV scenarios. Figure 6b shows that the PACF of the PV scenarios generated by TimeGAN matched those of the original PV scenarios, with extremely strong partial autocorrelations within 1 h lags, rapidly weakening beyond 1 h intervals. In comparison, the PACF of the PV scenarios generated by the other examined GANS showed noticeable differences compared to the original PV scenarios. The above analysis indicated that the PV scenarios generated by the TimeGAN model highly conformed to the fluctuation characteristics of the original PV data, fully capturing the temporal features of the original time series, in contrast to the PV scenarios generated by the other examined GAN models.

3.8. Typical Scenario Generation

The k-means clustering parameter k was set as 5, and the generated PV data with weather and load data with weather were clustered separately.

3.8.1. Rule Repository Generation

The FP-growth algorithm was utilized to mine the association rules from the classified PV data with weather. The results served as the rule repository. FP-growth requires setting a minimum confidence and support thresholds, which impact the mining results, as shown in Table 6.

Table 6. Number of association rules under different parameters.

Minimum Confidence	Minimum Support	Number of Rules
0.75	0.06	30
0.7	0.05	35
0.8	0.07	26
0.75	0.04	38
0.85	0.06	21

In this case study, the minimum confidence was set as 0.75, and the minimum support as 0.06, giving 30 rules. Partial results are shown in Table 7.

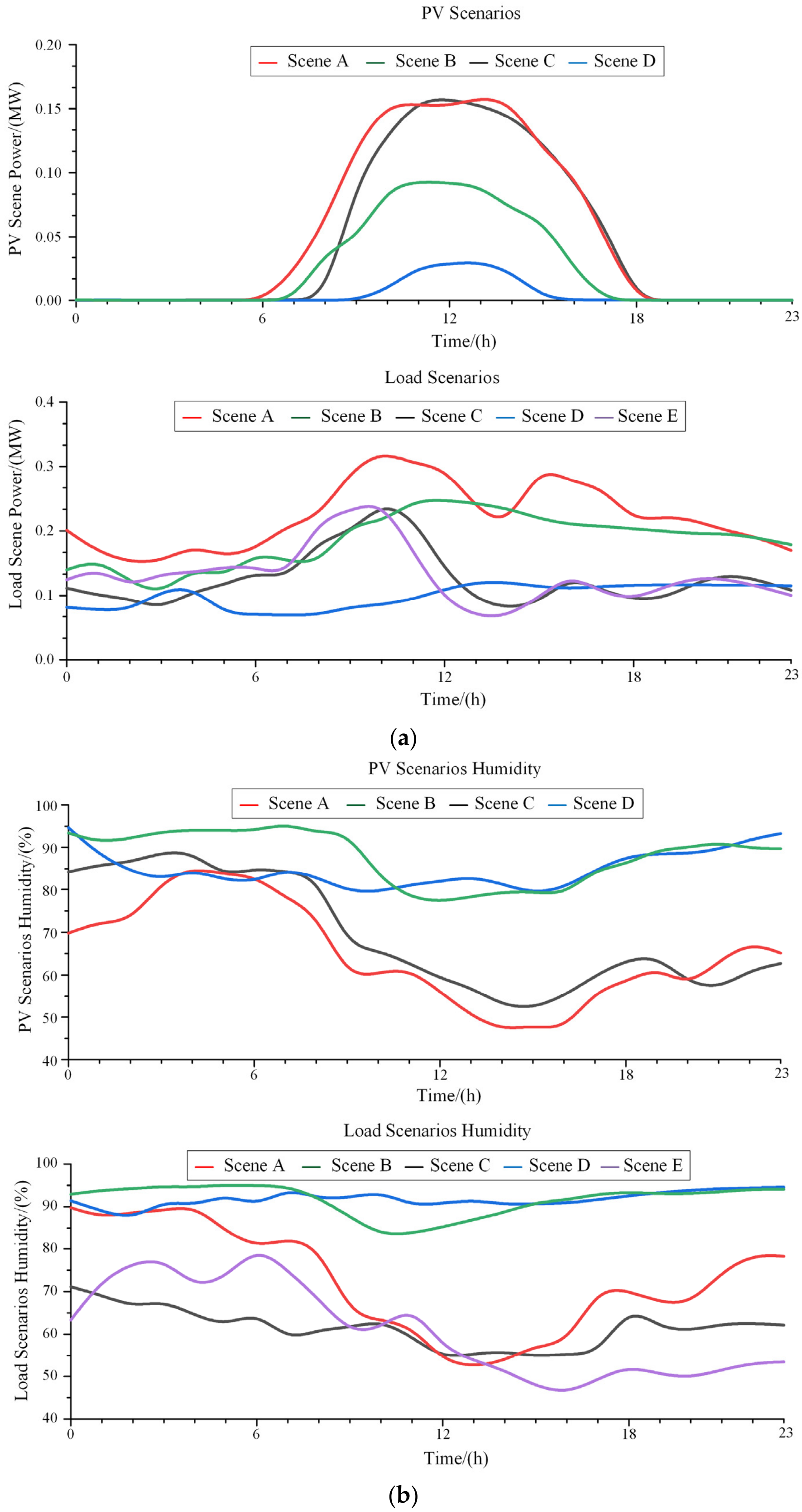
Table 7. Partial association rules.

Confidence	Weather Feature Type	PV Scenarios Type
1	Temperature: 4, Rainfall: 4, Humidity: 2	PV: 4
0.95	Humidity: 3, Rainfall: 0, Temperature: 0	PV: 1
0.946	Humidity: 3, Temperature: 4	PV: 2
0.898	Humidity: 2, Rainfall: 0, Temperature: 2	PV: 0
0.829	Rainfall: 1, Humidity: 1	PV: 3

3.8.2. Typical Scenario Set Generation

Based on the weather factors in the rules from the previous section, the corresponding PV scenarios were associated with the load scenarios to obtain typical PV–load scenarios matched according to the association rules. An example matching the first rule in Table 7 is shown in Figure 7, with same-color curves indicating scenarios with a similar trend.

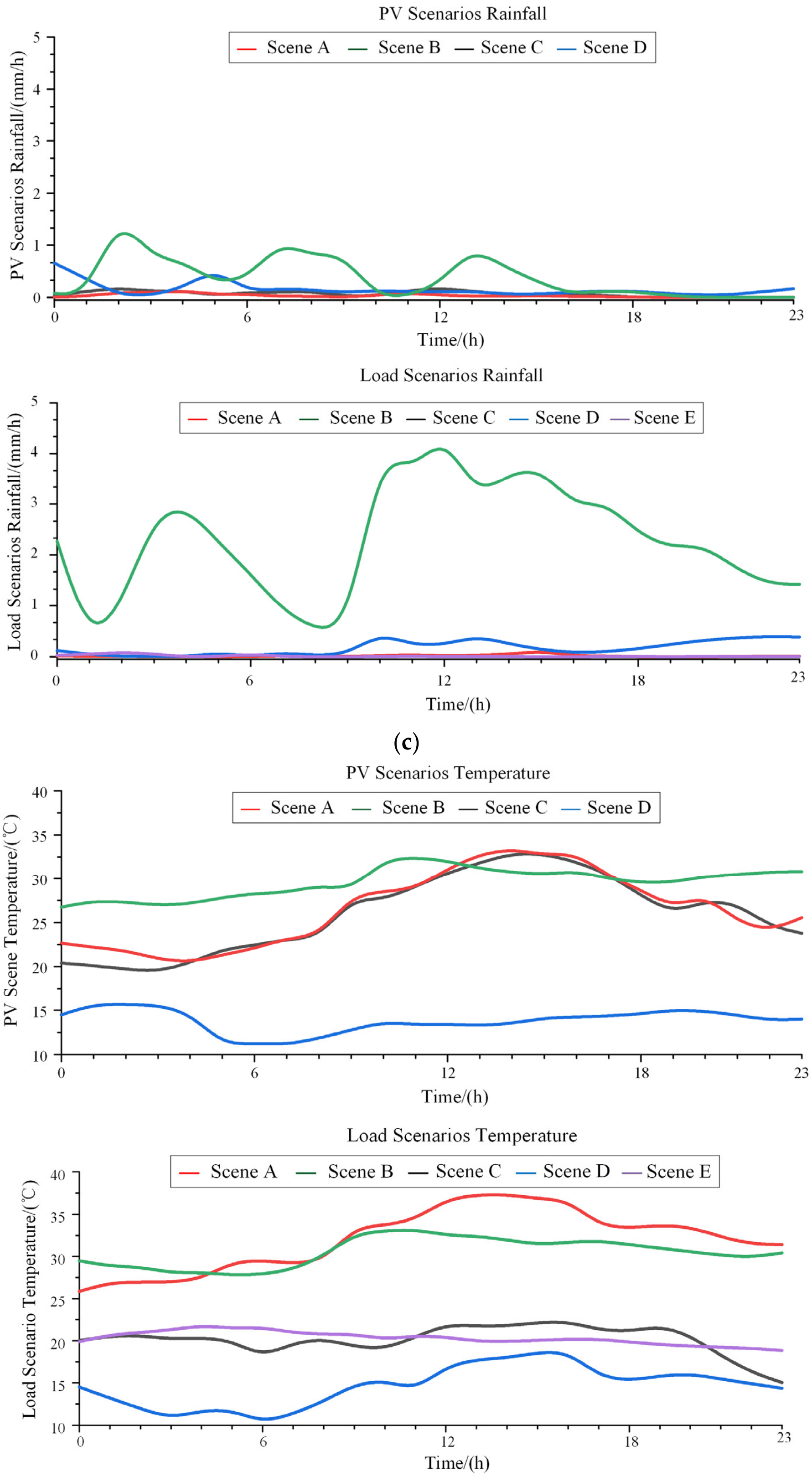
Analyzing the three weather scenarios corresponding to PV and load in Figure 7, the generated typical scenarios exhibited certain weather correlations. For example, the green curves correspond to PV and load scenarios with similar weather variation trends, representing high temperature and rainy weather. The red PV and load scenario weather changes also exhibited similar trends, with high temperature transforming into rain. The stacked typical load curves and PV curves of typical operational scenarios are shown in Figure 8 with the original scenarios.



(**a**)

(**b**)

Figure 7. *Cont*.



(**c**)

(**d**)

Figure 7. Partial PV–load scenario matching results. (**a**) PV–load typical scenarios; (**b**) humidity curves corresponding to PV and load typical scenarios; (**c**) precipitation curves corresponding to PV and load typical scenarios; (**d**) temperature curves corresponding to PV and load typical scenarios.

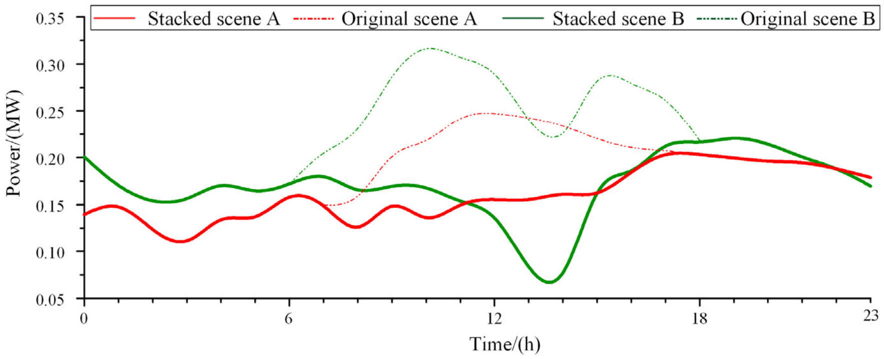
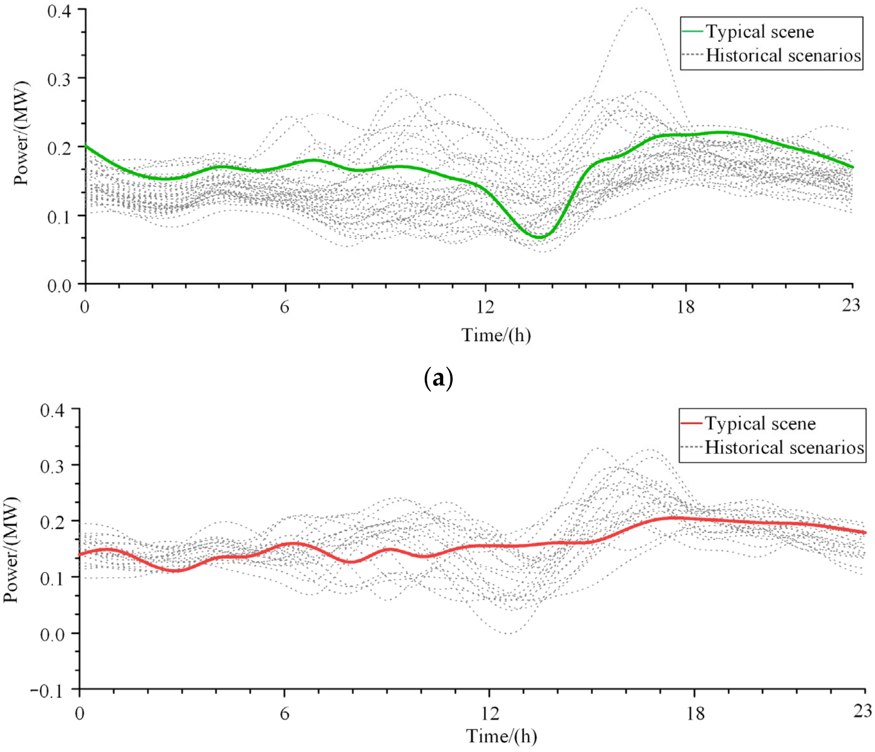


Figure 8. Stacked scene and original scenarios.

To verify the validity of the generated scenarios, historical data were screened under the weather conditions of the typical scenarios in Figure 8 to obtain corresponding historical scenarios. The relationships between the typical scenarios and the historical scenarios in Figure 8 are shown in Figure 9a,b.



(**b**)

Figure 9. Superimposed typical scenarios and historical scenarios. (**a**) The overlaid scenario of the green curve and the corresponding historical scenarios; (**b**) the overlaid scenario of the red curve and the corresponding historical scenarios.

In Figure 9, it can be seen that the typical scenarios generated by the proposed method were contained within the historical scenarios.

In the association rules selected in this section, the PV scenarios belonged to the fifth class, and the associated load scenarios belonged to the second and fifth classes. This showed that, compared to the traditional clustering for typical scenarios, the addition of weather association rules enabled the proposed method to extract typical scenarios across different classes, effectively ensuring the diversity of the generated scenarios. Moreover, the weather association rules allowed the generation of typical scenario sets under the meteorological conditions of different seasons or months based on statistical meteorological patterns, reducing the reliance on subjective prior knowledge for typical scenario generation to some extent.

4. Conclusions and Prospects

In sustainability research promoting power energy transformation, typical scenarios of power systems hold great significance, as they directly influence relevant decisions of SO or DRO. GANs have been extensively applied in domains like transfer learning and data augmentation in the power field, and numerous improved GAN algorithms have been proposed to heighten data authenticity [17–21]. However, power system scenarios are commonly time-series scenarios. Therefore, we propose adopting TimeGAN for data augmentation, since the TimeGAN's capability to excavate inherent temporal features of time-series data makes it highly suitable for applications in sustainable power systems. For example, in this study, TimeGAN enabled the generated scenarios to better resemble real-world situations. Clustering categorized scenarios under identical features into the same class, rendering the application of the FP-growth algorithm for association rule mining viable. Ultimately, we obtained typical scenarios with interpretable weather traits that better fit real-world temporal characteristics. Hence, we believe this holds substantial significance for the sustainable development of PV-integrated energy systems.

We investigated scenario generation incorporating weather factors in power systems utilizing the proposed framework consisting of TimeGAN, k-means algorithm, and FP-growth algorithm. Experiments also demonstrated the interpretability of weather in the generated typical scenarios. For future work, we have the following suggestions:

(1) Related research on Wasserstein distance could be introduced into TimeGAN.
(2) Building upon the weather factors, more impact factors such as holidays and electricity prices could be integrated for scenario generation to further enhance the method's practical applicability.
(3) The FP-growth algorithm and its generated results are relatively abstract. Clearer rule interpretations need to be further provided. Highly interpretable algorithms like classification based on associations (CBA) could be utilized for generating explanatory association rules.

Author Contributions: Conceptualization, X.W. and X.L.; methodology, X.W. and F.Z.; software, X.W. and Z.L.; validation, X.W. and X.L.; investigation, X.W. and K.X.; data curation, X.L. and F.Z.; writing—original draft preparation, X.W.; writing—review and editing, X.W. and X.L.; visualization, X.W. and K.X.; supervision, X.W.; project administration, Z.Z.; funding acquisition, X.W. and Z.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 62273104, the Science and Technology Program of Guangdong Power Grid Power Grid Co., Ltd., grant number 030800KK52220016.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data supporting this study's findings are available from the corresponding author upon reasonable request.

Acknowledgments: This work was supported by the National Natural Science Foundation of China, grant number 62273104, the Science and Technology Program of Guangdong Power Grid Power Grid Co., Ltd., grant number 030800KK52220016.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Uddin, M.; Mo, H.; Dong, D.; Elsawah, S.; Zhu, J.; Guerrero, J.M. Microgrids: A review, outstanding issues and future trends. *Energy Strategy Rev.* **2023**, *49*, 101127. [CrossRef]
2. Kong, X.; Xiao, J.; Liu, D.; Wu, J.; Wang, C.; Shen, Y. Robust stochastic optimal dispatching method of multi-energy virtual power plant considering multiple uncertainties. *Appl. Energy* **2020**, *279*, 115707. [CrossRef]

3. Hu, J.; Li, H. A transfer learning-based scenario generation method for stochastic optimal scheduling of microgrid with newly-built wind farm. *Renew. Energy* **2022**, *185*, 1139–1151. [CrossRef]
4. Yang, H.; Liang, R.; Yuan, Y.; Chen, B.; Xiang, S.; Liu, J.; Zhao, H.; Ackom, E. Distributionally robust optimal dispatch in the power system with high penetration of wind power based on net load fluctuation data. *Appl. Energy* **2022**, *313*, 118813. [CrossRef]
5. Zeng, Y.; Li, C.; Wang, H. Scenario-set-based economic dispatch of power system with wind power and energy storage system. *IEEE Access* **2020**, *8*, 109105–109119. [CrossRef]
6. Guo, M.; Wang, W.; Chen, R.; Li, Y. Research on bi-level model power dispatch considering the uncertainty of source and load. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102689. [CrossRef]
7. Huang, S.; Lu, H.; Chen, M.; Zhao, W. Integrated energy system scheduling considering the correlation of uncertainties. *Energy* **2023**, *283*, 129011. [CrossRef]
8. Dong, L.; Meng, T.; Chen, N.; Li, Y.; Pu, T. Optimized scheduling of AC/DC hybrid active distribution network using markov chains and multiple scenarios technique. *Autom. Electr. Power Syst.* **2018**, *42*, 147–153.
9. Zhang, C.; Wei, H.; Zhao, X.; Liu, T.; Zhang, K. A Gaussian process regression based hybrid approach for short-term wind speed prediction. *Energy Convers. Manag.* **2016**, *126*, 1084–1092. [CrossRef]
10. Huang, S.J.; Shih, K.R. Short-term load forecasting via ARMA model identification including non-Gaussian process considerations. *IEEE Trans. Power Syst.* **2003**, *18*, 673–679. [CrossRef]
11. Krishna, A.B.; Abhyankar, A.R. Time-coupled day-ahead wind power scenario generation: A combined regular vine copula and variance reduction method. *Energy* **2023**, *265*, 126173. [CrossRef]
12. Yao, G.; Wu, Y.; Huang, X.; Ma, Q.; Du, J. Clustering of typical wind power scenarios based on K-means clustering algorithm and improved artificial bee colony algorithm. *IEEE Access* **2022**, *10*, 98752–98760. [CrossRef]
13. Li, H.; Ren, Z.; Fan, M.; Li, W.; Xu, Y.; Jiang, Y.; Xia, W. A review of scenario analysis methods in planning and operation of modern power systems: Methodologies, applications, and challenges. *Electr. Power Syst. Res.* **2022**, *205*, 107722.
14. Pan, Z.; Wang, J.; Liao, W.; Chen, H.; Yuan, D.; Zhu, W.; Fang, X.; Zhu, Z. Data-driven EV load profiles generation using a variational auto-encoder. *Energies* **2019**, *12*, 849. [CrossRef]
15. Zhang, Y.; Ai, Q.; Xiao, F.; Hao, R.; Lu, T. Typical wind power scenario generation for multiple wind farms using conditional improved Wasserstein generative adversarial network. *Int. J. Electr. Power Energy Syst.* **2020**, *114*, 105388. [CrossRef]
16. He, Y.; Deng, J.; Li, H. Short-term power load forecasting with deep belief network and copula models. In Proceedings of the 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 26–27 August 2017; IEEE: Piscataway, NJ, USA, 2017; Volume 1.
17. Lin, S.; Wang, H.; Qi, L.; Feng, H.; Su, Y. Short-term load forecasting based on conditional generative adversarial network. *Autom. Electr. Power Syst.* **2021**, *45*, 52–60.
18. Dong, X.; Sun, Y.; Pu, T. Day-ahead scenario generation of renewable energy based on conditional GAN. In Proceedings of the CSEE, Munich, Germany, 9–12 November 2020; Volume 40.
19. Wang, C.; Tang, L.; Pu, Y.; Geng, Y. Scene Generation Method of Wind-solar Joint Output Based on Generative Adversarial Network. In Proceedings of the 2023 8th Asia Conference on Power and Electrical Engineering (ACPEE), Tianjin, China, 14–16 April 2023; pp. 104–109. [CrossRef]
20. Peng, B.; Sun, Z.; Liu, M. Medium and Long Term Scenario Generation Method Based on Autoencoder and Generation Adversarial Network. In Proceedings of the 2023 3rd International Conference on Neural Networks, Information and Communication Engineering (NNICE), Guangzhou, China, 24–26 February 2023; pp. 639–645. [CrossRef]
21. Zhang, C.; Shao, Z. Renewable power generation data transferring based on conditional deep convolutions generative adversarial network. *Power Syst. Technol* **2022**, *46*, 2182–2190.
22. Yoon, J.; Jarrett, D.; Van der Schaar, M. Time-series generative adversarial networks. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
23. Zeng, Y.; Yin, S.; Liu, J.; Zhang, M. Research of improved FP-growth algorithm in association rules mining. *Sci. Program.* **2015**, *2015*, 6. [CrossRef]
24. Rácz, B. nonordfp: An FP-Growth Variation without Rebuilding the FP-Tree. In *IEEE Icdm Workshop on Fimi*; IEEE: Piscataway, NJ, USA, 2004.
25. Han, J.; Pei, J.; Yin, Y.; Mao, R. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data Min. Knowl. Discov.* **2004**, *8*, 53–87. [CrossRef]
26. Yuan, C.; Yang, H. Research on K-value selection method of K-means clustering algorithm. *J* **2019**, *2*, 226–235. [CrossRef]
27. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein GAN. *arXiv* **2017**, arXiv:1701.07875.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.