EPA-GAN: Electric Power Anonymization via Generative Adversarial Network Model

Yixin Yang 1,2, Wen Shen 3, Qian Guo 3, QiuHong Shan 1,2, Yihan Cai 1,2 and Yubo Song 1,2,*

1 School of Cyber Science and Engineering, Southeast University, Nanjing 210003, China; 220224777@seu.edu.cn (Y.Y.); qiuishan@seu.edu.cn (Q.S.); 220224837@seu.edu.cn (Y.C.)
2 Purple Mountain Laboratories, Nanjing 211189, China
3 State Grid Smart Grid Research Institute Co., Ltd., Nanjing 210003, China; shenwen@geiri.sgcc.com.cn (W.S.); guoqian@geiri.sgcc.com.cn (Q.G.)
* Correspondence: songyubo@seu.edu.cn

Abstract: The contemporary landscape of electricity marketing data utilization is characterized by increased openness, heightened data circulation, and more intricate interaction contexts. Throughout the entire lifecycle of data, the persistent threat of leakage is ever-present. In this study, we introduce a novel electricity data anonymization model, termed EPA-GAN, which relies on table generation. In comparison to existing methodologies, our model extends the foundation of generative adversarial networks by incorporating feature encoders and feedback mechanisms. This adaptation enables the generation of anonymized data with heightened practicality and similarity to the original data, specifically tailored for mixed data types, thereby achieving a deliberate decoupling from the source data. Our proposed approach initiates by parsing the original JSON file, encoding it based on variable types and features using distinct feature encoders. Subsequently, a generative adversarial network, enhanced with information, downstream, generator losses, and the Was + GP modification, is employed to generate anonymized data. The introduction of random noise fortifies privacy protection during the data generation process. Experimental validation attests to a conspicuous reduction in both machine learning utility and statistical dissimilarity between the data synthesized by our proposed anonymization model and the original dataset. This substantiates the model’s efficacy in replacing the original data for mining analysis and data sharing, thereby effectively safeguarding the privacy of the source data.

Keywords: generative adversarial networks; data anonymization; privacy preservation; loss feedback; feature coding; power data protection

1. Introduction

Ensuring that the power system is secured and stable especially after it has been exposed to varieties of strains and different contingencies is a major challenge that energy stakeholders are facing in today’s world [1]. With the evolution of online collaboration channels in marketing and the innovative transformation of power company business models, the Power Marketing 2.0 business system aims to establish seamless data connectivity and business interaction across all operational channels to meet differentiated user needs on the basis of undertaking traditional customer service business [2]. Throughout this progression, the increasing prominence of data assets’ value is evident, accompanied by a substantial rise in the demand for data sharing. Concurrently, the risks associated with data security are becoming more pronounced.

To address potential privacy leakage issues stemming from the interaction of power data, scholars have proposed various solutions. Guan et al. [3] introduced an efficient communication scheme for safeguarding the privacy of meter data in smart grids. This scheme, requiring no trusted authority, ensures accurate power billing calculation by simply transmitting a set of data. Given its utilization of straightforward and computationally...
efficient operations, such as addition and hashing, it proves suitable for devices with limited computational resources. Jawureket et al. [4] presented a secure billing method employing homomorphic commitment technology. In this approach, measurement data are initially submitted and aggregated, with only the aggregated power data disclosed to the power company. Ultimately, the data’s accuracy is verified through zero-knowledge proofs. Kong et al. [5] proposed a group blind signature scheme to achieve conditional anonymity in smart grids, ensuring the integrity of power consumption data through homomorphic encryption. Li Yuyuan et al. [6] suggested a privacy protection request scheme for smart grids based on group blind signatures. In this scheme, smart meters can send power requests to substations, and the control center, unable to identify the real user identity through blind signatures, thereby ensures the privacy and security of power users. Nevertheless, these schemes fall short in effectively addressing the security concern of power data being reverse analyzed, leading to potential privacy leakage.

Generative adversarial networks (GANs), recognized as an emerging unsupervised learning methodology, find extensive application in image generation and are progressively garnering attention and utility within the domain of data science. As the practical applications and theoretical advancements in the field of generative adversarial networks unfold, scholars are increasingly redirecting their focus toward exploring data science. However, the majority of research related to GANs is predominantly concentrated on continuous datasets. Challenges arise when dealing with discrete data, primarily due to GANs encountering difficulties in utilizing gradient backpropagation for model training, given the non-differentiability of sampling from discrete distribution layers. To address this, Jang et al. [7] and Kingma et al. [8] proposed the Gumbel-SoftMax and concrete distribution methods, respectively, to tackle the generation of discrete data within the framework of variational autoencoders (VAEs). Kusner et al. [9] extended these methodologies to GAN models, thereby generating discrete sequence data based on the introduced VAE approach. Yu et al. [10] put forth a random policy approach based on reinforcement learning to circumvent the backpropagation challenges associated with discrete sequences. Another approach, proposed by Zhao et al. [11], involves adversarially regularized autoencoders (AR-AE), which transforms discrete terms used for training into a continuous latent feature space. Subsequently, GAN is employed to generate the latent feature distribution. Edward Choi et al. [12] introduced a model, medical generative adversarial network (med-GAN), akin to ARAE, specifically designed for generating synthetic binary or numerical data. Vincent et al. [13] presented an encoder–decoder method wherein the encoder maps samples to a low-dimensional continuous space, and the decoder reverts to the original data space. Leveraging the advantages of GAN in generating continuous data, the model enhances the accuracy of synthetic data generation for label variables by decoding the low-dimensional continuous space corresponding to the high-dimensional continuous or discrete space. Mottini et al. [14] proposed a GAN-based data generation method aimed at generating personal name information composed of missing/NaN values in categorical and numerical features.

Existing GAN-based data generation models primarily focus on distinguishing between discrete and continuous data [15]. However, in the context of electricity marketing business systems, the prevalent format for electricity data is JSON, which encompasses nested, mixed-type, and missing values. Consequently, a need arises for a data generator proficient in generating privacy-secured JSON data for electricity data.

In summary, this paper addresses the potential privacy breaches associated with electricity data in exploration, analysis, and business interactions. Through an analysis of electricity data characteristics and types, we propose the electric power anonymization via generative adversarial network model (EPA-GAN). The key contributions are as follows:

1. To tackle the diverse characteristics of electricity data, a method employing different feature encoders for data preprocessing is introduced, enhancing EPA-GAN’s resilience to imbalanced discrete variables and skewed continuous variables.
2. By integrating conditional generative adversarial networks, we achieve data synthesis through a zero-sum game. We enhance GAN training stability and effectiveness
through the utilization of loss feedback. Additionally, Renyi differential privacy is employed to control privacy budgets by introducing random noise to the generator during the data generation process.

3. Experimental results validate the practicality and similarity of the data synthesized by EPA-GAN proposed in this paper. It proves to be a viable substitute for original data in exploration, analysis, and data sharing, ensuring the privacy protection of the original data.

The subsequent sections of this paper are organized as follows: Section 1 provides an introduction to the fundamental concepts of conditional generative adversarial networks; Section 2 establishes the generative adversarial network-based electricity data anonymization model, elaborating on electricity data preprocessing and the data synthesis method based on generative adversarial networks; Section 3 conducts experiments and presents comparative analyses of the proposed EPA-GAN; finally, Section 4 offers a succinct summary.

2. Materials and Methods

2.1. Conditional Generative Adversarial Network

Conditional generative adversarial network (CGAN) [16] represents an expanded iteration of the generative adversarial network, strategically incorporating additional conditional information during sample generation to augment the control prowess of the generator. In conventional generative adversarial networks, the generator’s input comprises random noise, endeavoring to derive realistic data samples from this stochastic input, while the discriminator undertakes the task of assessing distinctions between the generated samples and authentic data.

However, CGAN introduces an extra conditional vector, conventionally denoted as c, as one of the inputs to the generator. This conditional vector encompasses diverse forms of information, such as categorical labels, textual descriptions, or images. By collaboratively inputting the conditional vector and random noise into the generator, it enables the generation of samples imbued with specific attributes dictated by the provided conditions. This refinement imparts heightened controllability to the generation process, facilitating the production of samples tailored to meet predefined criteria.

Through the incorporation of conditional information, CGAN bestows upon the generated samples distinctive attributes, fostering increased controllability and personalization in the resultant outcomes. This methodological approach presents a formidable instrument applicable across various tasks, empowering users to finesse outputs with nuanced precision during the generation process.

2.2. Differential Privacy

Differential privacy (DP) is a widely applied concept in the fields of data analysis and privacy protection. Its core objective is to minimize, to the greatest extent possible, the potential risk of individual privacy leakage while participating in data analysis. DP achieves this by introducing moderate noise or randomness in a mathematically rigorous and controllable manner, ensuring that the analysis process does not reveal sensitive information about individual entities.

The fundamental idea of DP lies in the addition of noise during data analysis or queries to obfuscate the true characteristics of the data, thereby preventing precise inferences about specific individuals. The careful design of this noise is crucial to balance the trade-off between protecting privacy and maintaining the accuracy of data analysis results.

Key steps in introducing differential privacy in GAN include modifying the training of the generator, parameter clipping to limit the model’s overfitting to individual data, injecting noise to reduce the model’s dependency on individual data, and setting a privacy budget to balance privacy and practicality. Such introductions help ensure that the generated data maintain a level of sensitivity to individual privacy, thereby enhancing the level of privacy protection.
2.3. Wasserstein Distance with Gradient Penalty

Wasserstein distance is a metric used to quantify the difference between two probability distributions. In the context of GAN, Wasserstein distance is widely employed to assess the distance between the data distribution generated by the generator and the real data distribution, typically serving as a measure of the quality of generated data.

Gradient penalty (GP) is a technique used to enhance the stability of GAN training. It ensures that the discriminator satisfies Lipschitz continuity conditions, meaning the gradient of the discriminator is bounded across the entire input space. By introducing a gradient penalty term, the discriminator’s gradient is constrained within a reasonable range, preventing pattern collapse and training instability issues.

Wasserstein distance with gradient penalty (Was + GP) [17] refers to the simultaneous use of Wasserstein distance as a loss function and gradient penalty in GAN training for both the generator and discriminator.

Was + GP combines the advantages of Wasserstein distance and the stability offered by gradient penalty, contributing to improved GAN training effectiveness and the quality of generated results. And this further stabilizes the training of the network and requires less hyper-parameter tuning.

3. Model Design

The configuration of EPA-GAN is illustrated in Figure 1. It consists of two main modules: a data preprocessing module and a data generation module. During the process of data anonymization, the original JSON file undergoes parsing using the JSON data parser within the data preprocessing module. Following this, encoding is carried out based on variable types and features, employing distinct feature encoders. Subsequently, the data generation module, comprising the generator G, discriminator D, and auxiliary components (classifier or regressor) C, is employed to generate anonymized data. Notably, given the reliance on conditional generative adversarial networks in the model algorithm, a noise vector and a conditional vector are introduced into the generator’s input for enhanced generative capabilities.

![Figure 1](image_url)

**Figure 1.** The general structure of EPA-GAN.

3.1. Data Preprocessing Module

Before transmitting the data to D and C, it is imperative to parse the data within the original JSON file and encode the variables using distinct feature encoders based on their types and characteristics.
3.1.1. JSON Data Parser

JSON (JavaScript Object Notation) format, as a lightweight data interchange format, has limitations such as supporting only a limited set of data types, lacking built-in metadata support, lack of standardized representations for dates and binary data, increased overhead for handling complex structures, and inability to handle circular references. However, despite these limitations, JSON is widely used in scenarios like power systems due to its simplicity, ease of use, and cross-platform compatibility.

The elements within the JSON file manifest in three forms: JSON objects, which hierarchically store data; JSON arrays, which concurrently store elements; and JSON primitives, directly encapsulating data.

This study initiates by parsing the original power data JSON file, denoted as J, into key-value pairs (denoted as P). Initializing P as an empty set, a tree structure T is then established for J, elucidating hierarchical relationships. Commencing from T, a comprehensive traversal of all nodes is undertaken. Upon encountering a JSON object during traversal, its key is appended to the existing prefix, initialized as an empty string. Subsequently, an underscore “_” is affixed to denote the hierarchical relationship, followed by continued exploration of the child nodes. In instances where a JSON array is encountered during parsing, parallel access to these elements suffices. Alternatively, when confronted with a JSON primitive, its key is appended to the prefix, facilitating backtracking and continued traversal until all nodes are visited. Figure 2 illustrates an exemplary scenario of JSON file parsing.

![JSON file parsing example.](image)

3.1.2. Feature Encoder

The feature encoder comprises three components: a hybrid encoder, a MinMax normalization encoder, and a logarithmic transformation encoder. During data preprocessing, the hybrid encoder is initially employed to individually encode variables. Subsequently, the MinMax normalization encoder is applied to discrete variables to address the issue of expanding dimensions due to a large number of discrete variable categories. The logarithmic transformation encoder is used to process continuous values, handling multi-modal data distribution and resolving the long-tail distribution—specifically, the challenge of rare points being distant from a significant amount of data.

- Hybrid Encoder

After parsing, data are encoded variable by variable. This paper categorizes variables into three types: discrete variables, continuous variables, and hybrid variables. A hybrid variable is defined if it contains both discrete and continuous values or if there are missing values within continuous values. To address such variables, this paper proposes a hybrid encoder. In this encoder, the values of hybrid variables are represented as sequences of value-pattern pairs. Here, “value” signifies specific values, and “pattern” denotes the nature of the variable, providing a more flexible representation of hybrid variables.

This manuscript elucidates the encoding methodology through the example of the mixed-variable distribution illustrated in Figure 3a in red. Notably, variable values can precisely be $\mu_0$ or $\mu_3$, representing discrete variables, or exhibit a distribution around the peaks of $\mu_1$ and $\mu_2$, indicative of continuous variables. When employing the variational
Gaussian mixture model (VGM) [18] to estimate the number of modes, denoted as k (e.g., in this instance, k = 2), and fitting the Gaussian mixture model, the study incorporates the modal-specific normalization (MSN) concept from [19] to address the continuous component. The Gaussian mixture acquired consists of \( P = \sum_{k=1}^{2} \omega_k N(\mu_k, \sigma_k) \) components, with \( N \) representing the normal distribution, and \( \omega_k, \mu_k, \) and \( \sigma_k \) corresponding to the weights, means, and standard deviations of each mode.

\[
\begin{align*}
&\text{(a) Mixed variable distribution} \\
&\text{(b) Mode selection for discontinuous variables}
\end{align*}
\]

**Figure 3.** Encoding of mixed data type variables.

To encode values of continuously distributed variables, each value is associated with the mode exhibiting the highest probability and subsequently normalized (refer to Figure 3b). For a given variable value \( \tau \), the probability densities of the two modes, labeled \( \rho_1 \) and \( \rho_2 \), are considered, and the mode with the highest probability is selected. In this specific example, given the higher probability of \( \rho_1 \), Mode 1 is utilized to normalize \( \tau \), resulting in the normalized value: \( \alpha = \frac{\tau - \mu_1}{\tau} \). Furthermore, one-hot encoding is employed for encoding the mode \( \beta \) associated with \( \tau \). The final encoding is a union of \( \alpha \) and \( \beta: \alpha \oplus \beta \), where \( \oplus \) denotes the vector union operator. The methodology for handling discrete components (e.g., \( \mu_0 \) or \( \mu_3 \) in Figure 3a) parallels that of the continuous range of mixed variables, with \( \alpha \) set directly to 0. For instance, for a value \( \beta = [0, 0, 0, 1] \) in \( \mu_3 \), the ultimate encoding is \( 0 \oplus [0, 0, 0, 1] \). Missing values are treated as a distinct and unique category. A row encompassing \([1, \ldots, N]\) variables is constituted by concatenating the encodings of all variable values, where the encoding of continuous variables and mixed variables is denoted as \( (\alpha \oplus \beta) \), and the encoding of discrete variables is \( \gamma \). In cases where there are \( n \) continuous/mixed variables and \( m \) discrete variables \( (n + m = N) \), the conclusive encoding is represented as:

\[
\oplus_{i=1}^{n} \alpha_i \oplus_{i=1}^{N} \beta_i \oplus_{i=n+1}^{N} \gamma_i
\]

**MinMax Normalization Encoder**

To address the challenges associated with handling discrete variables with a high number of categories, especially in the context of simple distributions like single Gaussian, we introduce MinMax normalization (MMN) in this paper. MMN efficiently mitigates algorithmic complexity by mapping variable ranges to \((-1, 1)\) through a combination of shifting and scaling.

The fundamental principle behind MMN is to ensure compatibility between the encoding and the output range of a generator utilizing the tanh activation function. Mathematically, this normalization operation is expressed as \( x_i' = 2 \cdot \frac{x_i - \text{min}(x)}{\text{max}(x) - \text{min}(x)} - 1 \), \( x_i \) represents the original variable’s value, \( x_i' \) denotes the normalized value, and \( \text{min}(x) \) and \( \text{max}(x) \) denote the minimum and maximum values of continuous variables.

For the reverse transformation of the normalized value \( x_i' \), it can be mapped back to the original range using the following formula: \( X_i = (\text{max}(x) - \text{min}(x)) \cdot \frac{x_i' + 1}{2} + \text{min}(x) \). Continuous variables can undergo direct normalization and reverse-normalization using this formula, while discrete variables, before normalization, are initially encoded as integers and rounded to integers after reverse-normalization.

Despite its utility, MMN comes with limitations. Specifically, its application leads to the loss of pattern indicators in the conditional vector, hindering the enhancement of
correlation between specific category variables. As a result, MMN is best suited for simple distributions, such as single-mode Gaussian distributions, within continuous columns, and is less effective in handling more complex distributions.

By default, EPA-GAN prioritizes the use of MSN for handling variables, reserving the application of MMN exclusively for instances where the number of categories in discrete variables overwhelms existing models’ capacity to train on encoded data.

- Logarithmic Transformation Encoder

In this study, we employ variational Gaussian mixture (VGM) for encoding continuous values, specifically designed for handling multimodal data distributions. It is noteworthy, however, that VGM encounters limitations in addressing long-tail distributions. To surmount this challenge, we integrate a preprocessing step involving a logarithmic transformation for variables exhibiting long-tail distributions. The logarithmic transformation serves the purpose of elongating the tail portion of the data, aligning it more closely with a normal distribution.

For variables constrained by lower bounds, we introduce a compressive alternative. The original value $\tau$ is replaced by the compressed logarithmic value $\tau^c$:

$$
\tau^c = \begin{cases} 
\log(\tau) & \text{if } l > 0 \\
\log(\tau - l + \epsilon) & \text{if } l \leq 0, \text{where } \epsilon > 0
\end{cases}
$$

This compression strategy facilitates the uniformity of the entire data range, enhancing the capability of VGM to capture the intricate relationship between tail values and the overall data distribution. It is imperative to note that the term “VGM” refers to the variational Gaussian mixture model, a crucial component in our encoding methodology.

3.2. Data Generation Module

The data generation module employs a generative adversarial network (GAN) trained through a zero-sum game, where $D$ (discriminator) aims to maximize a specified criterion, and $G$ (generator) endeavors to minimize the same criterion. Concurrently, $G$ is provided with additional feedback to enhance the quality and applicability of the generated data.

$G$ and $D$ utilize a CNN structure identical to that proposed by Park et al. in TableGAN [20]. To adapt the CNN for handling row-record data stored in vector form, this paper transforms the row data into the nearest square matrix of dimension $d \times d$, where $d$ is the square root of the dimensionality of the row data, with missing values filled with zeros. $C$ (classifier) employs a multilayer perceptron (MLP) with four hidden layers, each comprising 256 neurons. $C$ is trained on the original data to capture semantic integrity effectively.

During the data generation process, a random noise vector combines with a conditional vector input to $G$, producing synthetic data. Synthetic and real data are encoded into matrix form and compared by $D$. The encoded data are then reverse transformed into vectors, serving as input for $C$ to generate corresponding category label predictions.

3.2.1. Loss Feedback

This paper introduces additional feedback mechanisms for $G$ by incorporating information loss, downstream loss, and generator loss. Information loss aims to enhance the quality of synthetic data by making it closer to the original data, emphasizing both semantic and statistical consistency. Downstream loss is introduced to align the generator with the requirements of downstream tasks, improving the practical applicability of generated data. Generator loss, implemented through adversarial training, encourages the generation of more realistic data, thereby enhancing the overall quality and robustness of the generative model, making it challenging to distinguish synthetic data from real data.

During training, these three loss terms are added to $G$’s default loss terms ($W_{as} + GP$). The calculation involves the use of $f_x$ and $f_{G(x)}$ representing features of real and generated samples passed through the softmax layer of $D$. 

$$
\begin{align*}
&f_x & & f_{G(x)} \\
&\text{where } f_x & & f_{G(x)} \\
&\text{represent features of real and generated samples passed through the softmax layer of } D.
\end{align*}
$$
• Information Loss

Information loss $L_{\text{info}}^G$ is computed based on first-order (mean) and second-order (standard deviation) statistical data of synthetic and original data. The objective is to optimize synthetic data to align statistically with the original data.

$$L_{\text{info}}^G = ||E[f_x|x \sim p_{\text{data}}(x)] - E[f_{G(z)}|z \sim p(z)]||_2 + ||SD[f_x|x \sim p_{\text{data}}(x)] - SD[f_{G(z)}|z \sim p(z)]||_2$$

• Downstream Loss

Downstream loss $L_{\text{dstream}}^G$ reflects the correlation between the target variable and other variable values. It checks the semantic integrity of synthetic data and penalizes semantically incorrect combinations of values in synthetic records.

$$L_{\text{dstream}}^G = E[|l(G(z)) - C(fe(G(z)))|]_{z \sim p(z)}$$

• Generator Loss

Generator loss $L_{\text{generator}}^G$ is the cross-entropy between the given condition vector and the generated output class. It ensures the generator produces output classes consistent with the given condition vector.

$$L_{\text{generator}}^G = H(m_i, \hat{m}_i)$$

• Was + GP

Let $L_{\text{default}}^D$ and $L_{\text{default}}^G$ represent the discriminator and generator GAN losses for Was + GP. The unique objective function of D aims to maximize discriminator output for real data and minimize it for generated data, encouraging the generator to produce more realistic data. The gradient penalty term stabilizes the training process.

$$L = E_{x \sim P_x}[D(x)] - E_{x \sim P_x}[D(x)] + \lambda E_{\epsilon \sim P_\epsilon}[(\| \nabla_x D(\epsilon) \|_2 - 1)^2]$$

For $\lambda$, the complete training objective is to minimize $L^G = L_{\text{default}}^G + L_{\text{info}}^G + L_{\text{dstream}}^G + L_{\text{generator}}^G$. The training loss for C is similar to the downstream loss for the generator, i.e., minimizing $L_{\text{dstream}}^C = E[|l(x) - C(fe(x))|]_{x \sim p_{\text{data}}(x)}$.

3.2.2. Conditional Vectors

In the context of EPA-GAN, the deployment of CGAN proves instrumental in managing imbalances within training datasets. Furthermore, a method of sampled training is employed to expand the dataset, incorporating patterns from both continuous and mixed columns. During the sampling of real data, the introduction of conditional vectors becomes pivotal for data filtering and the reestablishment of balance in the training dataset.

The conditional vector, denoted as $V$, takes the form of a binary vector. It is meticulously crafted by amalgamating the one-hot encoding $\beta$ for all patterns across variables (pertaining to continuous and mixed variables) and the one-hot encoding $\gamma$ for all categories (pertaining to discrete variables) as outlined in Formula (1). Each unique conditional vector corresponds to a specific pattern or category. To illustrate, consider Figure 4, which presents an example featuring three variables—a continuous variable ($C_1$), a mixed variable ($C_2$), and a discrete variable ($C_3$) belonging to category 2.

<table>
<thead>
<tr>
<th>$C_1$</th>
<th>$C_2$</th>
<th>$C_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$\beta_1$</td>
<td>$\beta_2$</td>
<td>$\gamma_3$</td>
</tr>
</tbody>
</table>

Figure 4. Example of Conditional Vector.
In instances where a conditional vector is requisite during the training phase, a variable is initially chosen at random with uniform probability. Subsequently, the probability distribution of each pattern (or category, in the case of discrete variables) within that chosen variable is computed based on its frequency. The sampling of patterns ensues, guided by the logarithm of their respective probabilities. The adoption of logarithmic probabilities, as opposed to raw frequencies, is deliberate, fostering a more equitable representation of rare patterns or categories and mitigating the challenges associated with their scarcity. The extension of conditional vectors to encompass both continuous and mixed variables serves to rectify the issue of imbalanced pattern frequencies inherent in the representation of these variables. Notably, the generator’s imposition of conditions across all data types during the training process significantly enhances the inter-variable learning correlations.

3.2.3. Noise Vectors

In EPA-GAN, the utilization of Renyi differential privacy (RDP) imposes a more stringent control over privacy budgets, ensuring privacy protection during the data generation process.

RDP, an extension of differential privacy (DP), focuses on delivering heightened privacy protection in scenarios involving multiple queries or mechanisms. It introduces the concept of a budget, representing the cumulative upper limit of privacy leakage permitted across a series of queries or mechanisms. At the core of the RDP framework lies the composition theorem, allowing the synthesis of privacy losses from multiple queries into a comprehensive privacy constraint. This framework enhances the precision and robustness of privacy protection in the context of multiple queries or mechanisms, simplifying the management and control of privacy leakage in practical applications [21].

In this study, $M$ corresponds to a GAN model equipped with a privacy budget $(\lambda, \varepsilon)$. The $(\lambda, \varepsilon)$-RDP mechanism can be expressed as

$$\left(\frac{\log \frac{1}{\delta}}{\varepsilon + \frac{1}{\lambda - 1}}, \delta\right) = DP$$

(3)

if, for any adjacent datasets $S$ and $S'$, the Renyi divergence $D_{\lambda}(M(S) \parallel M(S')) = \frac{1}{\lambda - 1} \log E_{X \sim M(S)}[\frac{P[M(S)=x]}{P[M(S')=x]}]^{\lambda - 1} \leq \varepsilon$ holds, where $D_{\lambda}(P \parallel Q) = \frac{1}{\lambda - 1} \log E_{X \sim P}[\frac{(P(x)}{Q(x)}]^\lambda$ signifies the Renyi divergence.

Additionally, the Gaussian mechanism with parameter $M_G$ [7] is represented as $M_G(x) = f(x) + N(0, \sigma^2 I)$, where $f$ is an arbitrary function, and its sensitivity $\Delta_f = \max_{S \in S} \| f(S) - f(S') \|_2$ encompasses all adjacent datasets $S$ and $S'$, adhering to a Gaussian distribution with mean zero and covariance matrix $\sigma^2 I$ (where $I$ is the identity matrix) [21].

Furthermore, if $M$ satisfies $(\varepsilon, \delta) - DP$, then $F \circ M$ holds, where $F$ can be any arbitrary random function, $\circ$ denotes the composite operator. Consequently, in EPA-GAN, training only the generator in the generative adversarial network architecture ensures the entire GAN adheres to differential privacy, effectively curtailing privacy budgets.

4. Experimental Evaluation and Analysis

4.1. Parameterization and Evaluation Methods

This article’s anonymized experimental dataset is derived from the actual measurement data of Irish smart meters released by ISSDA. The experiments were conducted in the Ubuntu 20.04 environment on a personal laptop with a 3.20 GHz CPU and 16.00 GB of memory, utilizing standard tools and software.

To substantiate the efficacy of the proposed EPA-GAN in this paper, a comparative analysis is performed against data generation models based on CWGAN [22], CT-GAN [19] and TableGAN [20]. The evaluation of EPA-GAN encompasses two key aspects: the machine learning performance on anonymized data and the assessment of statistical similarity. This scrutiny aims to determine the viability of EPA-GAN for safeguarding privacy in
power data. Notably, decision trees and random forests have a maximum depth of 28, and MLP employs 128 neurons, with all algorithms implemented using scikit-learn 1.3.0 and default parameters.

For assessing the machine learning performance in classification tasks, the study employs five widely used algorithms (decision tree classifier, linear support vector machine, random forest classifier, multinomial logistic regression, and MLP) to measure performance metrics (accuracy, F1 score, and AUC) on both real and synthetic data.

Statistical similarity is evaluated using three metrics: Jensen–Shannon divergence (JSD), Wasserstein distance (WD), and correlation (corr). JSD is used to quantify the difference between the probability mass distribution of individual categorical variables for real and synthetic data sets. However, the JSD measure is numerically unstable for evaluating the quality of continuous variables, so WD is used to quantify the correspondence of the distribution of variables between the synthesized data set and the original data set. And coor. is used to calculate the feature correlation between the original data and the synthesized data.

4.2. Experimental Results and Analysis

Table 1 and Figure 5 present quantitative data on the differences in machine learning performance during classification tasks on real and synthetically generated data using three models. The table reveals that the proposed EPA-GAN significantly reduces differences across all metrics. This suggests that the featurized encoding and loss feedback in EPA-GAN contribute to enhancing feature representation and GAN training.

Table 1. Differences of ML utility between raw and synthetic data.

<table>
<thead>
<tr>
<th>Method</th>
<th>Differences of ML Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
</tr>
<tr>
<td>EPA-GAN</td>
<td>5.19%</td>
</tr>
<tr>
<td>CT-GAN</td>
<td>21.53%</td>
</tr>
<tr>
<td>TableGAN</td>
<td>11.44%</td>
</tr>
<tr>
<td>CWGAN</td>
<td>19.94%</td>
</tr>
</tbody>
</table>

Figure 5. Differences of ML utility between raw and synthetic data.

Table 2 illustrates the contrast in statistical similarity between synthetic and real data. Notably, in JSD, the performance of EPD-GAN is 47.9% higher than CT-GAN, 53.2% higher than TableGAN, and 72.6% higher than CWGAN. Superiority is also evident in terms of WD and data correlation. This highlights EPA-GAN’s effective preprocessing of imbalanced distribution data.

...
Table 2. Differences of statistical similarity.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>EPA-GAN</td>
<td>0.037</td>
<td>0.486</td>
<td>2.00</td>
</tr>
<tr>
<td>CT-GAN</td>
<td>0.071</td>
<td>1.771</td>
<td>2.68</td>
</tr>
<tr>
<td>TableGAN</td>
<td>0.079</td>
<td>2.122</td>
<td>2.29</td>
</tr>
<tr>
<td>CWGAN</td>
<td>0.135</td>
<td>237.089</td>
<td>5.72</td>
</tr>
</tbody>
</table>

The experimental results indicate high similarity in machine learning performance and statistical resemblance between the anonymized data synthesized by EPA-GAN and the original data. Therefore, these anonymized data can substitute for the original for mining analysis and interactive sharing, contributing to privacy protection in power system business data.

5. Conclusions

In response to potential privacy leaks during the interaction and mining analysis of power data in the marketing environment, this study proposes a generative adversarial network for anonymizing power data. This method synthesizes anonymized data with high practicality and similarity to the original, enabling its use in mining analysis and data sharing, effectively ensuring the privacy of original data. Initially, variables are encoded based on type and feature using different feature encoders to handle mixed categorical and continuous variables effectively. Subsequently, a generative adversarial network, improved through information, downstream, generator loss, and Was + GP, synthesizes anonymized data, incorporating random noise for privacy protection during data generation. Experimental verification demonstrates a significant reduction in machine learning performance and statistical similarity differences between the anonymized data synthesized by this method and the original data, compared to the current state-of-the-art models. The outcomes of this research provide an innovative privacy protection solution for the power industry, with broad application prospects. Therefore, in the future, we will research EPA-GAN to address scalability issues in handling large datasets, enhance its robustness against various types of attacks, and mitigate potential biases introduced during the anonymization process. Additionally, we will validate its effectiveness across different datasets and explore extending this method to other industries or domains, paving the way for its practical integration into real-world applications.

Author Contributions: Conceptualization, Y.Y.; methodology, W.S.; software, Q.G.; writing—review and editing, Q.S.; visualization, Y.C.; project administration, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Project of State Grid Anhui Electric Power Co., Ltd. [5400-202318221A-1-1-ZN].

Data Availability Statement: We confirm that the data supporting the findings of this study are available within the article. Additional data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: We would like to express our gratitude to the Science and Technology Project of State Grid Anhui Electric Power Co., Ltd. [5400-202318221A-1-1-ZN] for their support. We would also like to acknowledge the collective efforts of all the authors in this study.

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.
References


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.