Comparative assessment of generative models for transformer- and consumer-level load profiles generation

Xia, Weijie; Huang, Hanyue; Duque, Edgar Mauricio Salazar; Hou, Shengren; Palensky, Peter; Vergara, Pedro P.

DOI
10.1016/j.segan.2024.101338

Publication date
2024

Document Version
Final published version

Published in
Sustainable Energy, Grids and Networks

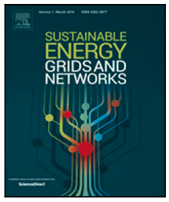
Citation (APA)
Xia, W., Huang, H., Duque, E. M. S., Hou, S., Palensky, P., & Vergara, P. P. (2024). Comparative assessment of generative models for transformer- and consumer-level load profiles generation. *Sustainable Energy, Grids and Networks*, *38*, Article 101338. https://doi.org/10.1016/j.segan.2024.101338

Sustainable Energy, Grids and Networks 38 (2024) 101338

Contents lists available at ScienceDirect

Sustainable Energy, Grids and Networks

journal homepage: www.elsevier.com/locate/segan



Comparative assessment of generative models for transformer- and consumer-level load profiles generation

Weijie Xia [a], Hanyue Huang [b], Edgar Mauricio Salazar Duque [c], Shengren Hou [a], Peter Palensky [a], Pedro P. Vergara [a,*]

[a] Department of Electrical Sustainable Energy, Delft University of Technology, Delft, 2628 CD, The Netherlands
[b] Faculty of Mathematics, Technical University of Munich, München, 80333, Germany
[c] Electrical Energy Systems Group, Eindhoven University of Technology, Eindhoven, 5612 AE, The Netherlands

ARTICLE INFO

Keywords:
Consumption profiles
Generative models
Distribution network
Generative adversarial networks

ABSTRACT

Residential load profiles (RLPs) play an increasingly important role in the optimal operation and planning of distribution systems, particularly with the rising integration of low-carbon energy resources such as PV systems, electric vehicles, small-scale batteries, etc. Despite the prevalence of various data-driven models for generating consumption profiles, there is a lack of clear conclusions about their relative strengths and weaknesses. This study undertakes a comprehensive comparison of frequently used data-driven models in recent research, including Generative Adversarial Networks (GANs), Variational Autoencoders (VAE), Wasserstein GANs (WGAN), WGANs with Gradient Penalty (WGANGP), Gaussian Mixture Models (GMMs), and Gaussian Mixture Copulas (GMC). The presented comparison explores the effectiveness of the above-mentioned models on transformer- and consumer-level consumption profiles, as well as for different time resolutions (15-min, 30-min, and 60-min). The objective of this research is to elucidate the respective advantages and drawbacks of these models, thereby providing valuable insights for subsequent research in this field.

1. Introduction

In 2019, the European Commission announced The European Green Deal, where achieving carbon neutrality by 2050 is one of the most critical goals [1]. The sustainability of cities plays a vital role in achieving carbon neutrality, which requires households to rapidly increase the adaptation rate of low-carbon energy resources [2]. However, these goals pose significant challenges to distribution system operators (DSOs), as the volatility of low-carbon energy resources such as photovoltaic (PV) and electric vehicles (EV) can create significant uncertainty in residential load profiles (RLPs) [3].

RLPs are the base for extensive applications such as demand response planning [4,5], integration of renewable energy sources [6] and EVs [7], system risk analysis [8,9] and distribution system planning [10], etc. However, despite their vast applications, the accessibility of consumer data in Europe is significantly hindered by strict privacy laws [11], with only a few open databases available [12–14]. Nevertheless, the effectiveness of these planning and operation approaches relies on the quality and representativeness of the used RLPs. They are crucial for DSOs to make well-informed planning and operation decisions [15,16]. Given the scarcity of data in most cases, coupled with the high demand for RLPs to support the higher-level tasks mentioned above, it is increasingly essential to develop models capable of generating RLPs with statistical characteristics similar to real-world data. The process of generating data with the same statistical features of a given data set is known in the machine learning field as *data augmentation* and has proven to improve the performance of such data-driven models [9,17]. In energy research, generated RLPs have been used in different applications, such as described in [18–20]. In these works, the generated profiles are used as augmented data for advanced load prediction model training. Similarly, in [21], generated profiles are used for non-intrusive load monitoring algorithms training. In [22,23], generated profiles can be used to understand consumption patterns and optimize the system planning.

In the existing literature, researchers employ three primary approaches to generate RLPs. The first approach exploits Markov Chain models [7,24,25]. These methods typically model RLPs indirectly, employing Markov Chains to simulate consumption behaviors first and then compute the corresponding RLPs. The focus of this paper is on directly modeling RLPs using data-driven models but not indirectly through modeling behaviors. Hence, Markov Chain models are not included in the presented model comparison. The second approach

* Corresponding author.
E-mail address: P.P.VergaraBarrios@tudelft.nl (P.P. Vergara).

https://doi.org/10.1016/j.segan.2024.101338
Received 9 November 2023; Received in revised form 16 January 2024; Accepted 4 March 2024
Available online 7 March 2024

employs classical statistical models such as Copulas, whereas the third employs deep learning methodologies.

Traditional models for RLPs often employ methods such as Gaussian Mixture Models (GMMs) and Copulas. In [26,27], GMMs are proposed to model the power distribution for each quarter of the day. In this model, the consumption at each time step is represented as a separate random variable. Although GMMs effectively reproduce the mean and variance of the original data, they may fail to model the correlation between different time steps, because GMMs model each time step independently. Building on [27], the work in [28] proposes a conditional multivariate elliptical Copulas model that can capture the correlation between time steps. Nevertheless, the model's performance decays as the resolution of the data increases. The work in [29] compares the performance of different multivariate Copulas functions in simulating EV charging consumption profiles. The student-t Copulas showed the best performance.

With advancements in artificial intelligence, machine learning offers a new approach to modeling RLPs, especially the ones based on deep learning models [30]. In [31], Generative Adversarial Networks (GANs) were employed to generate PV and wind generation profiles, successfully demonstrating GAN's capability to incorporate spatiotemporal correlations of renewable energy sources during profile generation. In [32], an Auxiliary Classifier GAN structure was proposed to generate RLPs associated with specific standard profiles obtained using k-means. Further, GANs have been used in [18–20] for RLPs generation to train other models. In [33], Conditional Wasserstein Generative Adversarial Network with Gradient Penalty (CWGAN-GP) is utilized for probabilistic load forecasting, where GANs are used to assist a forecaster by generating residual scenarios. These GAN-based models typically generate one profile at a time, which means they do not consider the correlation between profiles. To address this, the work in [34] proposed a MultiLoad-GAN that considers spatial–temporal correlations among a set of loads to generate many realistic synthetic consumption profiles. In a different approach, the work in [35] proposed a flow-based generative network to model RLPs, which outperformed GANs in aspects such as fitting peak and valley probability density functions. In [36], a transferable flow-based generation model was introduced for day-ahead consumption profile prediction, and pre-training was performed using transfer learning. Besides GANs and flow-based models, the Variational Auto-Encoder (VAE) is also a popular generative model. The work in [37] proposed VAE to generate EV charging profiles, and the results demonstrate that VAE can retain the temporal correlation and probability distribution of the original consumption profiles. The work in [38] proposed a multivariable load state generation model based on conditional VAE, which shows better generation quality than basic VAE. In contrast to conventional methods, machine-learning-based techniques typically do not require initial assumptions such as a specific distribution function (unlike models such as GMMs). This implies that machine-learning-based methods may be more effective in capturing latent correlations and generating more realistic RLPs. However, this flexibility can also make the models challenging to train and may lead to unstable results. For instance, GAN models are notoriously hard to train. Another potential issue specific to GAN-related models is *model collapse*, a scenario where the model starts to generate a limited diversity of outputs, consequently failing to reproduce the original distribution accurately.

Despite the broad range of models available in the literature for profile generation, a comprehensive comparison and a guide of model selection depending on the application between machine learning-based models, such as GANs, and traditional models, such as Copulas, is still missing. This study aims to address this gap by conducting a performance comparison of the most frequently employed data-driven models in recent research, namely GMMs [26,27], GANs [20,39], VAE [37,38], WGAN [18,31], WGAN-GP [33], and Gaussian Mixture Copula (GMC) [27,29]. The performance comparison presented in this paper explores the effectiveness of these models on both consumer- and transformer-level consumption profiles and in different time resolutions (15-min, 30-min, and 60-min). It is essential to highlight that conducting a wholly fair comparison is challenging, given the distinct mathematical underpinnings of each model. However, we are committed to providing a balanced evaluation. The contributions of this research are as follows.

- A comprehensive performance assessment of six state-of-the-art RLP modeling methods and a detailed analysis of their strengths and weaknesses are presented. This evaluation provides critical insights into the effectiveness of each method in various application scenarios.
- The identification and recommendation of the most appropriate RLP modeling approaches for different tasks, which are valuable for generating high-quality RLPs that are tailored for higher-level applications, such as demand response and system planning.

The Python code, datasets, and other related materials of this research are freely available in [40] and [41].

2. Modeling of residential load profiles

In the modeling of RLPs, a typical daily profile is segmented into T discrete time steps. For instance, RLPs with a resolution of 15 min are characterized by $T = 96$ time steps, while those with a 30-min resolution comprise $T = 48$ time steps. Each time step within these profiles corresponds to a specific value of active power consumption.

In this paper, we denote each time step as a continuous random variable X_i, where $i = 1, 2, \dots, T$. The realized value of X_i is donated as x_i, then the probability density of x_i can be denoted as $p_i(x_i)$. The realized value of active power consumption at the ith time step in the jth RLP is represented by x_{ij}. Consequently, the set of all RLPs is expressed as $L = \{l_j\}_{j=1}^{N} = \{(x_{1j}, x_{2j}, \dots, x_{Tj})\}_{j=1}^{N}$, where each l_j represents an RLP in the dataset. Each l_j depicts a unique power consumption pattern across the T time steps, embodying the practical measurements or observations of a specific RLP.

3. Model introduction

Gaussian Mixture Models: A GMMs is a weighted sum of normal distributions, which is usually defined as [42]:

$$p(x) = \sum_{i=1}^{K} \omega_i \mathcal{N}(x|\mu_i, \sigma_i), \tag{1}$$

where K is the total number of components of GMMs, $\mathcal{N}$ is the normal distribution, μ_i, σ_i are the mean and variance of ith normal distribution, ω_i is the weight of ith model and $\sum_{i=1}^{K} \omega_i = 1$, x represents the data point or observation, and $p(x)$ is probability density of the data point x being generated by the GMMs. The weight and parameters of components of GMMs can be solved by the expectation maximization (EM) algorithm. In this paper, each $p_i(x_i), \forall i = 1, 2, \dots, T$ is modeled by GMMs. The details are described in Section 5.

Generative Adversarial Networks: A GANs comprises two components: a generator and a discriminator. The training process of a GAN is conceptualized as a min–max game. The objective of the generator is to maximize the losses of the discriminator by producing samples that the discriminator cannot distinguish from real ones, while the discriminator aims to maximize the losses of the generator by correctly identifying the synthetic samples [43]. Ultimately, the network is expected to achieve a Nash equilibrium, where neither the generator nor the discriminator can further improve by changing their strategies. The loss function of discriminator L_D is defined:

$$\begin{aligned} \min_D L_D = &-\frac{1}{2}\mathbb{E}_{l\sim p_{\text{data}}}\left[\log D(l)\right] \\ &-\frac{1}{2}\mathbb{E}_{z\sim p_z}\left[\log(1 - D(G(z)))\right], \end{aligned} \tag{2}$$

where p_{data} is the distribution of real data, p_z is the distribution of the noise, l and z represents the real sample (in this paper, l corresponds to RLP) and noise, $D(l)$ and $G(z)$ are the output of discriminator and generator respectively. The loss function of generator L_G is defined as:

$$\min_G L_G = -\mathbb{E}_{z\sim p_z}[\log D(G(z))]. \tag{3}$$

Thus, the total GAN loss is given by the sum of the generator and discriminator losses:

$$\min_G \max_D L_{GAN} = \min_D L_D + \min_G L_G, \tag{4}$$

where L_{GAN} is the total GAN loss.

Wasserstein Generative Adversarial Network: Similar to GAN, WGAN comprises a generator and a critic. WGAN modifies the GAN's loss function to use the Wasserstein distance, also known as the Earth-Moving (EM) distance [44]. The benefit of this approach is that it provides smoother gradients and avoids the problem of mode collapse often seen in standard GANs. The objective function of a WGAN can be written as follows:

$$\min_G \max_{D\in\tilde{D}} \mathbb{E}_{l\sim p_{\text{data}}}[D(l)] - \mathbb{E}_{z\sim p_z}[D(G(z))], \tag{5}$$

where $\tilde{D}$ is the set of 1-Lipschitz functions, $D(l)$ is the output of the critic for real data, and $D(G(z))$ is the output of the critic for generated data. The generator tries to minimize the objective in (3), while the critic tries to maximize it.

WGAN with Gradient Penalty: WGAN-GP is an extension of the original WGAN. It introduces a penalty term to the loss function to ensure that the critic of the network satisfies the Lipschitz constraint, which helps mitigate problems such as mode collapse and unstable training [45]. The objective function of a WGAN-GP is expressed as:

$$\begin{aligned}\min_G \max_D \mathbb{E}_{l\sim p_{\text{data}}}[D(l)] - \mathbb{E}_{z\sim p_z}[D(G(z))] \\ +\lambda\mathbb{E}_{\hat{l}\sim p_{\hat{l}}}[(\|\nabla_{\hat{l}} D(\hat{l})\|_2 - 1)^2],\end{aligned} \tag{6}$$

where $D(l)$ is the output of the critic for real data, and $D(G(z))$ is the output of the critic for generated data. The generator G attempts to minimize the objective in (4), while the critic D tries to maximize it. The third term, $\lambda\mathbb{E}_{\hat{l}\sim p_{\hat{l}}}[(\|\nabla_{\hat{l}} D(\hat{l})\|_2 - 1)^2]$, is the gradient penalty. Here, $\hat{l}$ is an interpolated sample created by blending a real data sample and a generated data sample. $\nabla_{\hat{l}} D(\hat{l})$ is the gradient of the critic's output with respect to this interpolated sample, and λ is a hyperparameter that controls the strength of the penalty. This gradient penalty term encourages the gradient norm of the critic's output to be close to 1 almost everywhere. This is done to enforce the Lipschitz constraint, leading to improved training stability.

Variational Autoencoder: VAE is a type of generative model that combines deep learning and variational inference techniques to model complex data distributions [46]. It is composed of two primary components: an encoder and a decoder. The encoder maps input data points l into a latent space z by modeling them as a distribution $Q(z|l)$. Meanwhile, the decoder maps these latent points back to the data space by modeling them as a distribution $P(l|z)$. The objective function of a VAE can be written as:

$$\begin{aligned}\mathcal{L}_{\text{VAE}}(\theta,\phi;l) = \mathbb{E}_{z\sim Q_\phi(z|l)}[\log P_\theta(l|z)] \\ -\text{KL}(Q_\phi(z|l) \parallel P(z)),\end{aligned} \tag{7}$$

where $P_\theta(l|z)$ represents the likelihood of data given the latent variable z, $Q_\phi(z|l)$ is the approximate posterior that encodes the data l into the latent space, $P(z)$ is the prior belief over the latent variable, often chosen to be a standard normal distribution for simplicity, KL stands for the Kullback–Leibler divergence. The first term $\mathbb{E}_{z\sim Q_\phi(z|l)}[\log P_\theta(l|z)]$ is the reconstruction loss, which ensures that the VAE can reconstruct the input data well after encoding it. The second term $\text{KL}(Q_\phi(z|l) \parallel P(z))$ is the regularization term, which measures the divergence between the approximate posterior $Q_\phi(z|l)$ and the prior $P(z)$.

Table 1
Five test cases including two transformer-level and three consumer-level dataset.

Test case	Year	Households	Resolution
Netherlands-traf [12]	2013	82	60 min
United Kingdom-traf [14]	2013	94	30 min
Netherlands-cons [12]	2013	1	60 min
United Kingdom-cons [14]	2013	1	30 min
Germany-cons [13]	2020	1	15 min

Gaussian Mixture Copulas: Derived from Sklar's theorem [47], Copulas are statistical constructs designed to separate the dependency structure of multivariate data from their marginal distributions. GMC model combines the strengths of both GMMs and Gaussian Copulas. In a GMC, the marginal distributions of the data are modeled using a GMMs, allowing the model to handle non-normal marginal distributions and mixtures of distributions. The dependence structure between the variables is then modeled using Gaussian Copulas. The first step in constructing a GMC is to fit a GMMs to each of the marginal distributions of the data:

$$p_i(x_i) = \sum_{k=1}^{K} \omega_{i,k}\mathcal{N}(x_i|\mu_{i,k},\sigma_{i,k}^2), \tag{8}$$

where $\mathcal{N}(x_i|\mu_{i,k},\sigma_{i,k}^2)$ denotes a normal distribution with mean $\mu_{i,k}$ and variance $\sigma_{i,k}^2$, $\omega_{i,k}$ are the mixture weights for the ith random variable x_i, and K is the number of components in the mixture. After fitting the GMMs, the cumulative distribution function (CDF) F_i of each marginal distribution can be estimated. The Gaussian Copulas is then applied to the resulting CDFs:

$$C_{\mathbf{R}}(l) = \phi_R\left(F_1^{-1}(x_1), F_2^{-1}(x_2), \ldots, F_T^{-1}(x_T)\,;\mathbf{R}\right), \tag{9}$$

where F_i^{-1} is the inverse CDF of the ith random variable (obtained from the GMMs), $\mathbf{R}$ is a correlation matrix, and ϕ_R is the joint CDF of a multivariate normal distribution with covariance matrix equal to $\mathbf{R}$.

4. Data description

To conduct an adequate evaluation of the models, five test cases with varying resolutions were designed using data from three different datasets. Table 1 outlines the details of the used test cases. The first two test cases represent transformer-level RLPs. The latter three cases focus on consumer-level RLPs from single households. These 5 test cases were built to comprehensively test all models' ability to model complex statistical behavior at high resolutions. Note that while the UK dataset (second test case in Table 1) includes data from more than 5,000 households, for a fair comparison, a subset of 94 households was randomly selected for this study.

5. Modeling approach

This section introduces the modeling approach of the six models described in Section 2. For each test case in Table 1, the performance of these six selected models was compared. Fig. 1 summarizes the structure and modeling approach of each of these models, highlighting the input (or original) data and the generated data, as well as each model's component, including the loss function and the latent variables (or space) if the model considers one. A summary of the models' setup is described next.

GMMs: As we mentioned in Section 2, each time step can be considered as a random variable $X_i, i = 1, 2, \ldots, n$, following the distribution $p_{real}(x_i)$. We then use GMMs to model the distribution of each X_i to obtain the modeled distribution $p_{gmm}(x_i)$. Therefore, n distinct GMMs are constructed during the development phase. During the data generation phase, sample once from all GMMs associated with each time step to get $\{\hat{x_1}, \hat{x_2}, \ldots, \hat{x_n}\}$ and then combined to obtain one RLP. The optimal number of components is determined by the Bayesian

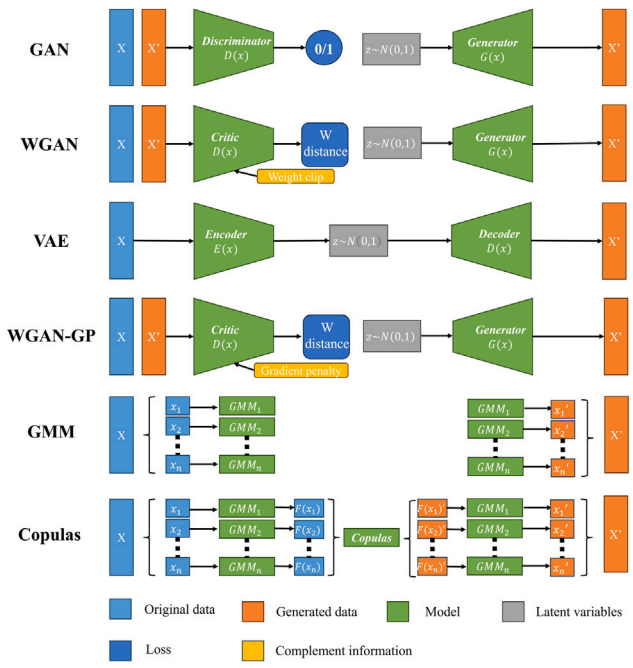


Fig. 1. Illustration of the construction of six models and modeling approach highlighting the input (or original) data and the generated data, as well as each model's component, including the loss function and the latent variables (or space) if the model considers one.

Information Criterion (BIC). Further details regarding the selection of component numbers are provided in Appendix A.1.

GMC: GMC builds upon the foundation of GMMs. In the modeling phase, each GMMs associated with a single time step is treated as a separate marginal distribution, resulting in a total of n marginal distributions. The RLPs data is then transformed into cumulative probabilities (bounded between $[0, 1]$) by applying the CDF of the GMMs. Subsequently, a multivariate Gaussian distribution is utilized to fit the transformed data. In the generating phase, cumulative probabilities are sampled from multivariate Gaussian distribution, and then the value of each time step is generated by applying inverse CDF on sampled cumulative probabilities.

GAN & WGAN & WGANGP: In the training phase, RLPs together with generated RLPs (generated by the generator) are fed into the discriminator (critic), as Fig. 1 shows. Then, the discriminator (critic) is trained to minimize the classification loss, as defined in (2), while the generator is trained to maximize the loss of the discriminator (critic), as defined in (3). After the model is trained, random noise is fed into the generator to generate RLPs.

VAE: For VAE, a similar sampling process as done with the GAN model is performed. Here, each RLP is considered as one sample and is fed to the model. The training is conducted to minimize the reconstruction loss and KL divergence loss, as defined in (7). After the model is trained, random noise is fed into the encoder to generate RLPs.

For the sake of fairness in comparison, the architecture of the deep generative models, including the VAE, whose decoder is designed to have the same structure as the GAN's generator, is kept consistent in depth and width. Further details on the configuration and parameter settings of these deep generative models are comprehensively documented in Appendix A.2.

6. Comparison metrics

Considering the metrics used in [27,28,32,48], the assessment of the generated RLPs is conducted using a variety of statistical measures to ensure a comprehensive and robust evaluation. These metrics are presented next:

Wasserstein distance: Also known as the Earth Moving Distance, this metric is sensitive to the geometry of the distribution [49] and measures the minimum cost of transporting mass to transform one distribution into the other. The Wasserstein distance $W(\mu, \nu)$ between two probability measures μ and ν is defined as:

$$W(\mu,\nu) = \inf_{\pi\in\Pi(\mu,\nu)} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|l_u - l_v\|\, d\pi(l_u, l_v), \tag{10}$$

where $W(u, v)$ is the Wasserstein distance between μ and ν, l_u and l_v are the sample in μ and ν. A smaller value of $W(\mu, \nu)$ indicates greater similarity between u and v. Specifically, $W(\mu, \nu) = 0$ means u and v are identical.

Jensen–Shannon (JS) divergence: It is a symmetric measure of the similarity between two probability distributions. Ranges from 0 (when the distributions are the same) to $\log 2$ (when the distributions are disjoint). The JS divergence $D_{JS}(P \parallel Q)$ between two probability distributions P and Q is given by:

$$D_{JS}(P \parallel Q) = \frac{1}{2} D_{KL}(P \parallel M) + \frac{1}{2} D_{KL}(Q \parallel M), \tag{11}$$

where $M = \frac{1}{2}(P + Q)$ and D_{KL} is the Kullback–Leibler divergence.

Multi-Maximum Mean Discrepancy (MK-MMD): This metric quantifies the difference between distributions using their mean embeddings within a reproducing kernel Hilbert space (RKHS) [50]. Typically employing Gaussian kernels, MK-MMD can detect diverse distributional differences in a high-dimensional space. MK-MMD is defined as:

$$\begin{aligned} MK-MMD^2(P,Q) = &\frac{1}{n(n-1)} \sum_{i\neq j}^{n} \sum_{p=1}^{P} \alpha_p k_p(l_i^p, l_j^p) \\ &- \frac{2}{nm} \sum_{i,j}^{n,m} \sum_{p=1}^{P} \alpha_p k_p(l_i^p, l_j^q) \\ &+ \frac{1}{m(m-1)} \sum_{i\neq j}^{m} \sum_{p=1}^{P} \alpha_p k_p(l_i^q, l_j^q), \end{aligned} \tag{12}$$

where α_p are coefficients of the linear combination, P is the total number of kernels, k_p is the pth kernel function, n and m are the number of samples from distribution P and Q, l^p and l^q are the sample in distribution P and Q respectively. A smaller value of MK-MMD indicates greater similarity between two distributions, and a 0 value of MK-MMD indicates two distributions are identical.

Kolmogorov–Smirnov (KS) Distance: It is a measure of the distance between the cumulative distribution functions of two distributions. The KS distance $D_{KS}(F, G)$ between two CDFs F and G is defined as:

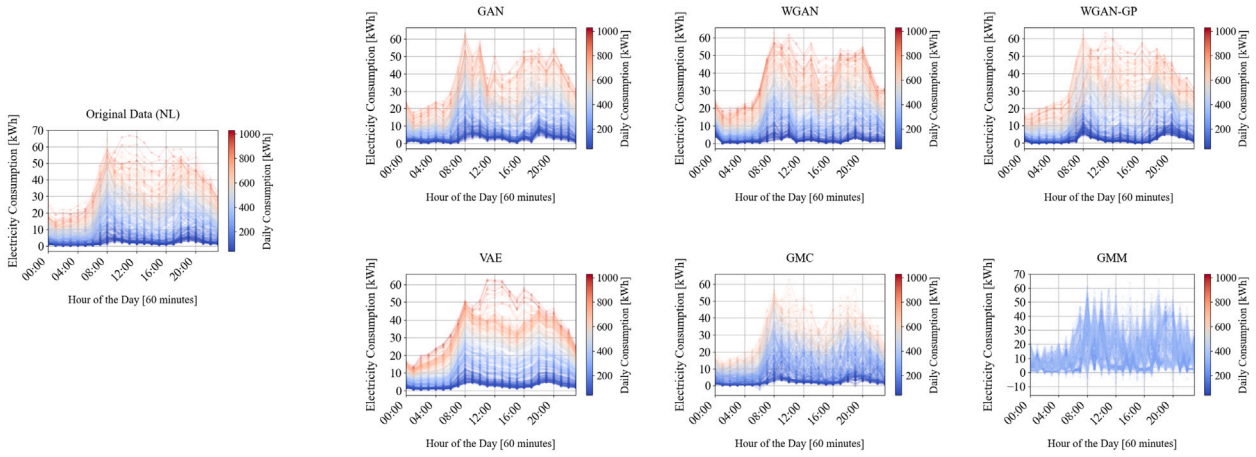
$$D_{KS}(F,G) = \sup_{l} |F(l) - G(l)|. \tag{13}$$

KS distance can handle any distribution and is sensitive to differences in both the location and shape of the empirical CDFs. Similar to MK-MMD, a smaller value of KS distance indicates greater similarity between two distributions, and a 0 value indicates two CDFs are identical.

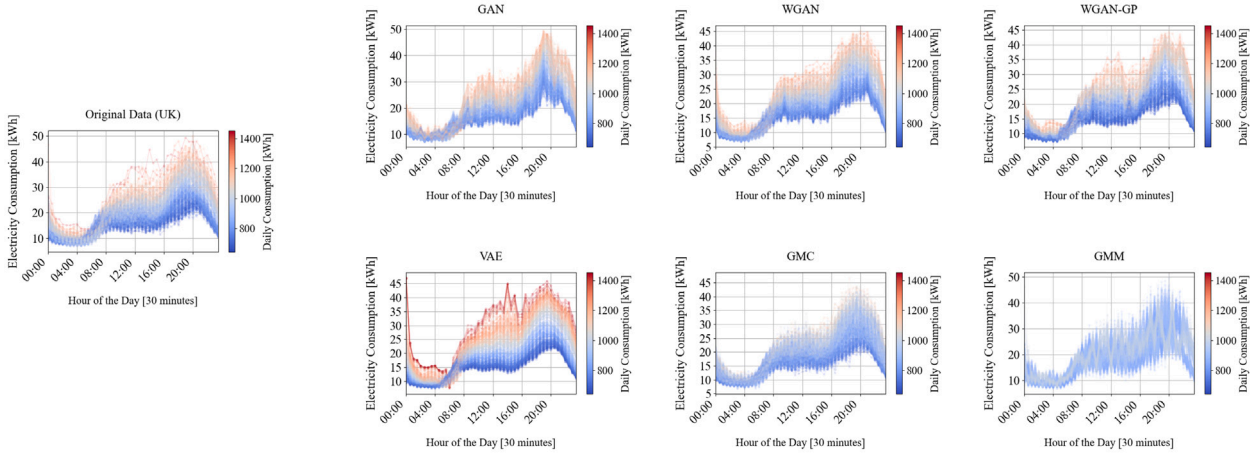
Autocorrelation: It is also an important factor in assessing the quality of generated time series data. The autocorrelation $R(k)$ of a time series at lag k is given by:

$$R(k) = \frac{\sum_{t=1}^{N-k}(X_t - \bar{X})(X_{t+k} - \bar{X})}{\sum_{t=1}^{N}(X_t - \bar{X})^2}, \tag{14}$$

where $R(k)$ is the autocorrelation at lag k, N represents the total number of data points in the time series, X_t represents the value of the time series at the time t, $\bar{X}$ is the sample mean of the time series. In this research, we use the Mean Square Error (MSE) of Autocorrelation of generated RLPs and original RLPs to evaluate the quality of modeling time correlation.



(a) Generated results and original data of NL-traf test case



(b) Generated results and original data of UK-traf test case

Fig. 2. Right side: Results generated by six models for two transformer-level test cases. Left side: Original data. The color of the RLPs corresponds to the sum of daily consumption. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

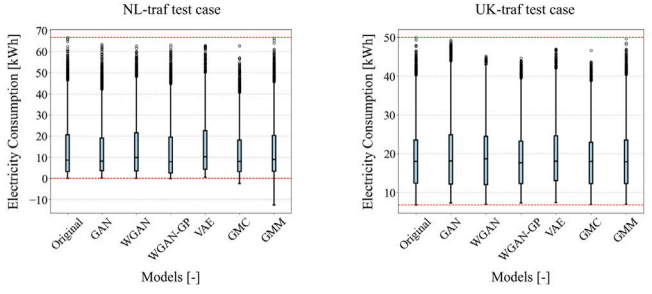


Fig. 3. Box plots of the results generated by different models and the original data of NL-traf and UK-traf test cases, where the red dashed line represents the highest and lowest consumption values in the original data.

7. Generated results

7.1. Transformer-level data results

The original data and results generated by the six models for the NL-traf and UK-traf test cases are shown in Fig. 2. One of the uses of RLPs is supporting the planning and operation of distribution systems, peak power consumption and its occurrence time should be correctly estimated to improve the reliability of the distribution network (e.g., by considering the worst-case consumption scenario). After comparing the generated results with the original data in Fig. 2, it is clear that all models successfully replicated the original pattern. For example, in the NL-traf test case, the peaks usually occur around 9 AM and 8 PM, while in the UK-traf test case, the peaks usually occur around 8 PM. All models effectively capture these fluctuations.

Another observation is that the GMMs only generate RLPs of an average energy consumption value (which can be observed in Fig. 2, that RLPs generated by GMMs have the basically same color). This is due to the GMMs modeling each time step independently. As a result, GMMs typically produce RLPs with average daily consumption when generating the RLPs. Besides that, GMMs fall short in modeling the time correlation between different time steps. Table 2 demonstrates that GMMs have the highest MSE of the correlation matrix. Additionally, in Fig. 3, the box plot reveals that GMMs is the only model that successfully captures the highest peak value in both the UK-traf and NL-traf test cases, where the red dashed line represents the extreme values of the original data.

In Fig. 4, it is observed that the generated RLPs of GMM and GMC have negative values, an anomaly that would not occur in real-world data. This phenomenon is evident in the box plot in Fig. 3. In the NL-traf test case, the lowest consumption is 0 kWh. However, GMMs generate non-positive values. This phenomenon arises from the Gaussian's long tails, and it is absent in deep generative models because the output of their generator (or the decoder for VAE) is bounded by the activation function (e.g., Tanh [−1,1]).

Table 2
Evaluation of different models using comparison metrics for NL/UK-traf test case.

Models	Wasserstein distance	JS divergence	KS distance	MK-MMD	Correlation MSE
NL test case (60 min)					
GAN	0.8583	**0.0676**	0.0300	0.0253	0.0045
WGAN	0.8169	0.0743	0.0386	0.0218	**0.0043**
WGANGP	0.7543	0.0931	0.0743	0.0247	0.0060
VAE	1.4678	0.0903	0.0728	0.0283	0.0234
GMC	1.4816	0.2890	0.0483	**0.0111**	0.0341
GMMs	**0.2516**	0.4877	**0.0243**	0.0741	0.5991
UK test case (30 min)					
GAN	0.6929	0.0954	0.0465	0.0330	0.0170
WGAN	0.6000	0.1318	0.0439	0.0178	**0.0023**
WGANGP	0.2824	0.1080	0.0249	0.0132	0.0107
VAE	0.6684	0.1296	0.0481	0.0255	0.0804
GMC	0.4860	0.0748	0.0289	**0.0107**	0.0247
GMMs	**0.0481**	**0.0460**	**0.0058**	0.0482	0.2323

Table 3
Evaluation of different models using comparison metrics for NL/UK/GE-cons test case.

Models	Wasserstein distance	JS divergence	KS distance	MK-MMD	Correlation MSE
NL test case (60 min)					
GAN	0.0192	**0.1587**	0.3723	0.0144	0.0528
WGAN	0.0291	0.1766	0.2708	0.0133	0.0154
WGANGP	0.0211	0.1890	0.3214	0.0084	0.0434
VAE	0.0112	0.3062	0.2752	**0.0031**	0.0273
GMC	0.0331	0.6113	**0.2142**	0.0107	**0.0134**
GMMs	**0.0061**	0.6510	0.2437	0.0265	0.0844
UK test case (30 min)					
GAN	0.0145	0.1147	0.0704	0.0063	0.0441
WGAN	0.0091	0.1446	0.0535	0.0060	0.0059
WGANGP	0.0096	**0.0998**	0.0060	0.0126	0.0142
VAE	0.0174	0.2104	0.0485	0.0201	0.0039
GMC	0.0043	0.4574	**0.0153**	**0.0055**	**0.0034**
GMMs	**0.0029**	0.7503	0.0232	0.0056	0.0734
GE test case (15 min)					
GAN	0.0306	**0.1820**	0.2799	0.0222	0.0373
WGAN	0.0159	0.1993	0.1515	0.0144	0.0176
WGANGP	0.0178	0.1836	0.2361	0.0273	0.0541
VAE	0.0745	0.2060	0.1604	0.0218	**0.0091**
GMC	0.0391	0.6058	**0.1375**	**0.0143**	0.0748
GMMs	**0.0032**	0.6554	0.1553	0.0257	0.1313

Comparing the generation results of GMC with other models, it can be noticed that GMC falls short in generating RLPs with high daily consumption in both NL-traf and UK-traf test cases. As can be seen from Fig. 4, the RLPs generated by GMC have a relatively light color that represents low daily consumption. This phenomenon is also observed in consumer-level test cases, evident in the box plot in Fig. 3. A more detailed explanation of this phenomenon is provided in Section 8. On the other hand, looking closely at the generated RLPs of VAE for both NL-traf and UK-traf, it can be noticed that there is a tendency for the model to produce RLPs of higher total daily consumption. This phenomenon is more clear in consumer-level test cases, as shown in Section 7.2.

Table 2 presents the results using the evaluation metrics of the six models for the NL-traf and UK-traf test cases. The findings suggest that (1) GMMs, while struggling with modeling time step correlations, excel in modeling the distribution of individual time steps, consistently performing the best in terms of the Wasserstein distance. (2) GAN-related models (GAN, WGAN, and WGAN-GP) generally perform well in modeling time correlations, with GAN exhibiting a particular strength in minimizing JS divergence. However, GAN-related models fail to reproduce the mean and variances. (3) GMC often emerges as the top performer in MK-MMD, a metric that compares means in high-dimensional space. This might suggest that GMC could be the most balanced approach.

7.2. Consumer-level data results

Fig. 4 illustrates the original data and the results generated by six models for UK-cons, NL-cons, and GE-cons test cases. In alignment with previous observations, all models proficiently reproduce the patterns of peaks and valleys. Fig. 4 also shows that the results generated by the GAN model begin to exhibit reduced diversity, or in other words, the GAN model starts to generate several similar RLPs. This phenomenon is known as *model collapse* and has been described in [51]. Low RLPs generation diversity is apparent in Fig. 4(b), where the generated results are noticeably less diverse than the original data. Compared to GAN, other deep generative models such as WGAN, WGAN-GP, and VAE appear more adept at generating diverse results. As previously discussed, VAE tends to generate RLPs with higher total daily consumption. This trend is again observable in Fig. 4(b), and it is particularly noticeable in Fig. 4(c).

The generated RLPs in consumer-level test cases by GMC and GMMs, compared to transformer-level cases, shows clearly that these models can produce results with negative values. This observation is more evident in Fig. 5. Aligned with the findings in Section 7.1, GMC struggles to generate RLPs with high peak values. However, it is noted that GMC exhibits exceptional performance in the UK-cons test case, where it ranks first in KS Distance, MK-MMD, Correlation MSE, and second in Wasserstein Distance (see Table 3). Further exploration of this occurrence is presented in Section 8. Finally, as a general result, it is observed that GAN-related models continue to demonstrate good performance in modeling time correlation. Moreover, it was found that WGAN had a very stable performance in modeling time correlation no matter if consumer- or transformer-level test cases.

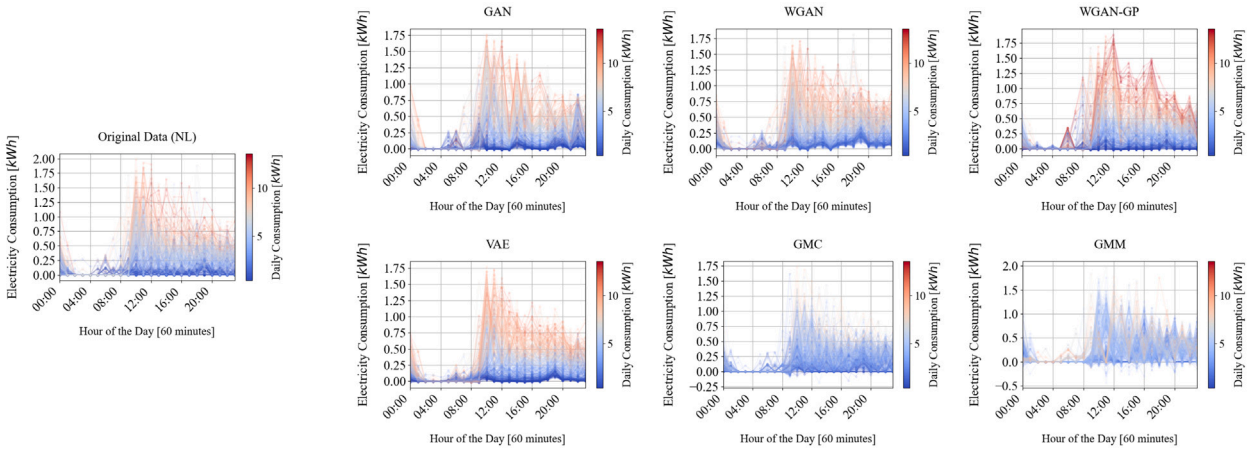
7.3. Peak generation analysis

Peak consumption is a critical factor in modeling RLPs due to its significance in system planning and operation. This section offers a detailed quantitative evaluation of how accurately models replicate these peak values, expanding upon the preliminary discussion in Section 7. To evaluate the model's performance in this dimension, a dedicated *time-peak* plot is used as observed in Fig. 6. This *time-peak* plot displays the peak consumption of each RLP and its corresponding time of occurrence, represented as a point, denoted as *(time, peak)*. Then, the average (or average *time-peak*) center is derived from the set of *(time, peak)* points generated by the model. Subsequently, the proximity of these centers are calculated using the Euclidean distance metric. The following comparison provides an informative perspective on the models' ability to generate peak consumption at the correct time step.

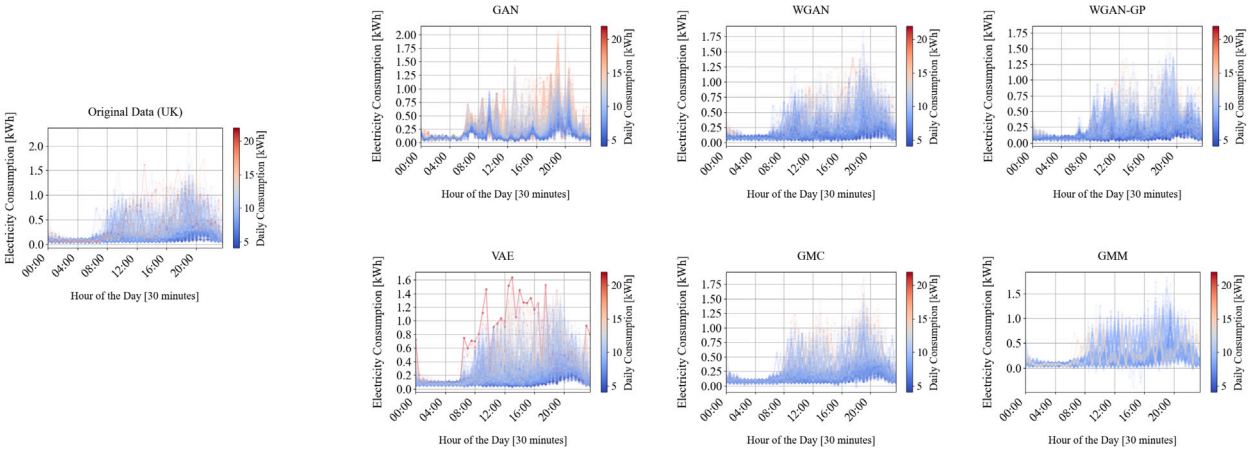
Fig. 6 shows the *time-peak* plots for five different test cases. As shown in Fig. 6, the GAN and WGAN-GP models are more closely aligned to the center of the original data when compared with the other models. This finding is further confirmed when Table 4 is examined, which shows that the GAN correlation model performs well in modeling both temporal and peak correlations. Similarly, the GMC model also shows commendable performance in transformer-level test cases. In contrast, it is evident from Fig. 6 and Table 4 that GMMs consistently underperform. This finding echoes the previous conclusions that GMMs fail to account for correlations between time steps, whereas the GAN models show superior performance in this regard.

8. Discussion

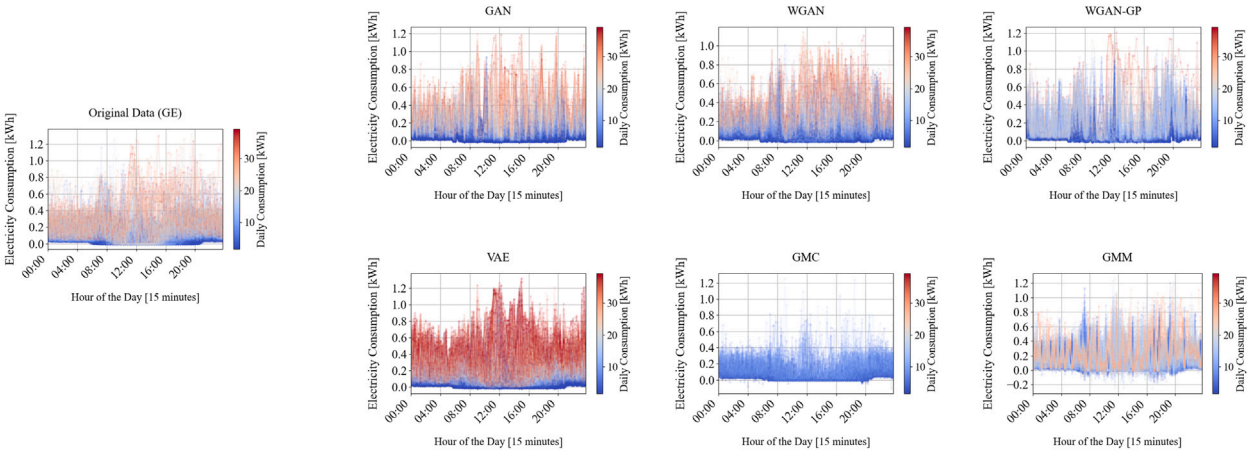
It is worth noting that the GMC shows solid performance on the UK-cons test case, ranking highly on almost all metrics. This raises the question of why GMC performs so well in this case and why



(a) Generated results and original data of NL-cons test case



(b) Generated results and original data of UK-cons test case



(c) Generated results and original data of GE-cons test case

Fig. 4. Right side: Results generated by six models for two transformer-level test cases. Left side: Original data. The color of the RLPs corresponds to the sum of daily consumption. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 4
The Euclidean distance between the centers of generated data and the center of the original data in 'Time-Peak' graph.

Test case	GAN	WGAN	WGAN-GP	VAE	GMC	GMMs
Netherlands-traf	**1.0477**	2.3562	1.5200	3.0638	1.2061	17.1290
United Kingdom-traf	5.5904	1.1452	0.6670	1.7507	**0.5954**	6.6306
Netherlands-cons	**0.1012**	0.3438	0.1989	1.2130	0.8944	0.8941
United Kingdom-cons	**0.2632**	0.6793	0.3087	0.9894	0.4694	1.0246
Germany-cons	4.4367	4.6575	**2.7051**	2.4672	5.9377	5.2930

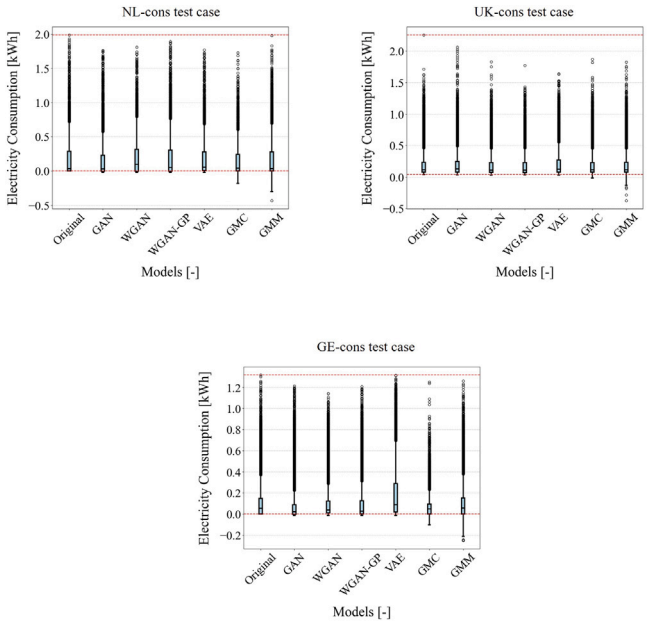


Fig. 5. Box plots of the results generated by different models and the original data of three consumer-level test cases, where the red dashed line represents the highest and lowest consumption values in the original data.

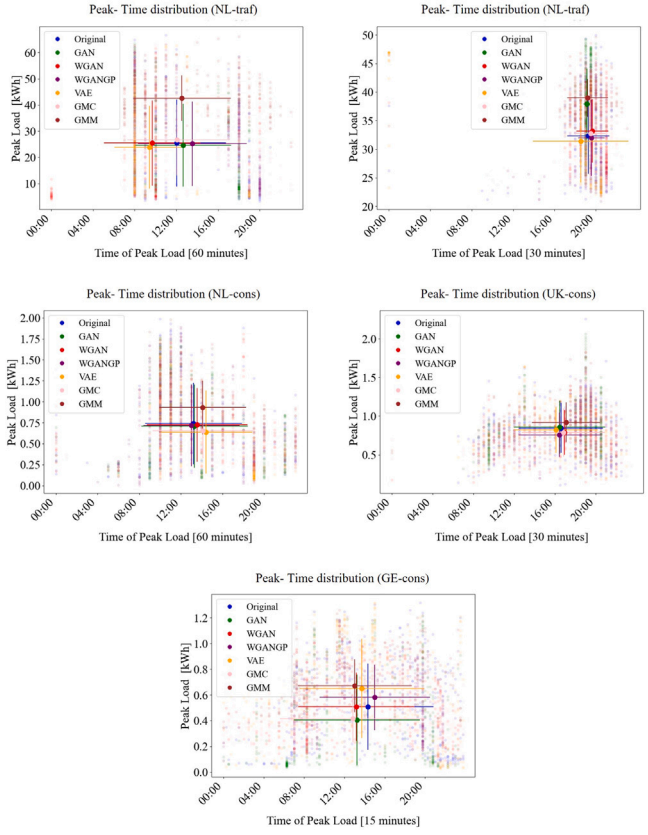


Fig. 6. Time-peak graph for five test cases. The length of the line perpendicular to and parallel to the coordinate axis through the center point represents the variance on this axis.

its performance decays in other test cases. Test case results have shown that GMMs, as the basis of GMCs, perform well in modeling

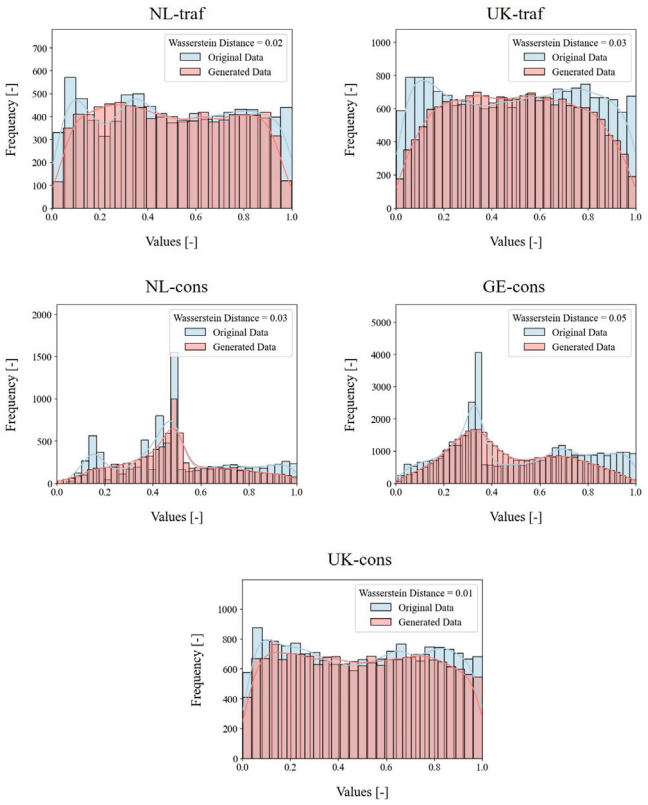


Fig. 7. The distribution of original and generated cumulative probabilities of 5 test cases by GMC. The Wasserstein distance of generated and original cumulative probabilities is computed. The figures show that the performance of the GMC is influenced by the distribution shape of the cumulative probabilities.

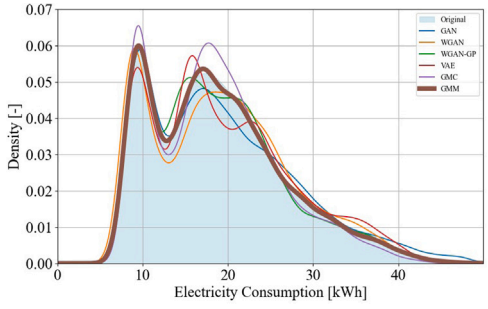


Fig. 8. The distribution of electricity consumption of original and generated data (UK-traf 30 min). The x-axis represents the active power consumption of a time step and the y-axis represents the frequency.

marginal distributions. Therefore, the errors in the GMC model are likely caused by inaccuracies in constructing the correlations of the marginal distributions. Fig. 7 shows the distribution of the original and generated cumulative probabilities, also known as uniform distribution margins. Cumulative probabilities are obtained using the CDF to transform the original data into the range $[0, 1]$ as mentioned in Section 5. Results show that the Gaussian Copulas more accurately align with the cumulative probabilities of the UK-cons test case than otherwise, especially in the tails, which closely match the distribution of the original data. The Wasserstein distance further validates this observation, showing significantly lower values for GMC (0.1) than the other distances (greater than 0.2). Conversely, GMC also does not

Table 5
Summary of main findings.

Q1: Which model should be selected to model residential load profiles? Considering the trade-off between performance and complexity, WGAN is often preferred, consistently providing satisfactory results in various scenarios. However, choosing the appropriate model depends on the specific application requirements. As detailed in Section 8, standard GANs are advantageous for capturing precise peak timings and temporal patterns, but they suffer from model collapse. On the other hand, GMM offers simple implementation and fast training (within a few minutes for test cases in this paper). Although GMMs effectively capture general statistical consumption patterns, they do not adequately reproduce time correlations.
Q2: How long does it take to train the compared models (model training complexity)? Training duration escalates with higher data resolution. GMMs train rapidly, within minutes, across all test cases. GMC training spans under a minute to around twenty minutes for 15-minute resolution RLPs. Utilizing dual GPU parallelism, deep generative models require several minutes to twenty to thirty minutes for 15-min resolution RLPs (based on the structure of the models and data in this paper).
Q3: Which model should be selected in case of insufficient amounts of data? Deep generative models often require large amounts of data to produce reliable and diverse results. Insufficient amounts of data increase the likelihood that these models will overfit (models memorize data instead of learning to generate new data). In this data-limited scenario, it is recommended to choose GMMs and GMC (or other Copulas models), which are more robust to limited data sets and, therefore, more likely to provide stable performance.
Q4: How to design a customized deep generative model for data with different time resolutions? The deep generative model framework proposed in this study provides a reference for customized model development. Appendix A.2 details the adaptive architecture of the network, where scalability is addressed through the parameter s. Adjusting s upward accommodates higher data resolution, thereby tailoring the model to specific data granularity requirements (for super high resolution, a redesign might be necessary).

match the tails of the other test cases, which explains its inability to generate high peak RLPs in these instances. From the above, it can be concluded that the performance of GMC might depend heavily on the distribution shape of the cumulative probabilities in practical applications. Notably, the GMC exhibits enhanced congruence with distributions that are closer to a bell shape, as exemplified by the UK-consumer dataset. The reason behind that is GMC uses a multivariant Gaussian to model the correlation between time steps which might not align with the real data correlations. In contrast, deep generative models exhibit greater adaptability in modeling temporal correlations. This flexibility stems from their inherent design, which is not constrained by the assumption of a Gaussian distribution, thereby enabling more consistent performance across varying data characteristics.

In our observations, GMMs consistently recorded the lowest Wasserstein distance. As mentioned earlier in Section 6, the Wasserstein distance is particularly sensitive to the shape of the distribution. Fig. 8 shows the distribution of the original and generated data. It is worth noting that while the data generated by other models often deviates slightly from the shape of the original distribution, the distribution generated by the GMMs closely matches the shape of the original data. The same phenomenon can be observed in other test cases, which explains why GMMs always show the lowest Wasserstein distance.

After evaluating the performance of six generative models, it was observed that GMMs are good at modeling the overall distribution but poor at capturing temporal dependencies. The GMC model builds on the GMMs and performs better in modeling time-step dependencies. However, as discussed previously, GMC often has difficulty generating RLPs with peaks, and its ability to fit various distributions may not be as good as deep generative models. Moreover, GMC tends to perform well on MK-MMD, but determining the reason for this is challenging, given the complexity of the MK-MMD metric. Nonetheless, it is recognized that, unlike other models, GMC does not exhibit significant weaknesses on all evaluation metrics, suggesting that it may be a model with the best balance of features. Concerning the deep generative models, they have shown excellent performance in modeling temporal dependencies, especially the WGAN model. Furthermore, GAN-related models show superiority in modeling correct $(time, peak)$ pairs compared to other models. GANs also achieve good results on JS divergence, probably because GANs are trained fundamentally to minimize JS divergence. However, deep generative models fail to reproduce the mean and variance accurately of original data. To further assist in modeling RLPs, Table 5 summarizes some practical questions and answers based on the conclusions of this paper.

9. Conclusion

In this paper, the performance of six models was evaluated, simulating five different test cases, including two sets of transformer-level data and three consumer-level data. Six evaluation metrics for this analysis were used. The results showed that no single model significantly outperforms the others on all metrics, but each model exhibits unique strengths and weaknesses. Deep generative models such as GAN, WGAN, GAN-GP, and VAE have shown superiority in modeling time-step correlations, while WGANs excel at modeling overall time step dependencies, GANs excel at accurately predicting peak times. Both GMC and GMMs, especially GMMs, are good at replicating overall statistical patterns, although GMMs are not good at temporal correlation. Additionally, a practical guide is provided in Table 5. As far as future research on RLPs generation is concerned, we believe it would be beneficial to develop a model that accurately represents temporal correlations and overall statistical patterns. Furthermore, to facilitate the optimal planning of future power distribution systems with highly integrated PV, EV, etc., accurate future RLPs generation can be crucial.

CRediT authorship contribution statement

Weijie Xia: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Resources, Software, Validation, Writing – original draft, Writing – review & editing. **Hanyue Huang:** Formal analysis, Methodology. **Edgar Mauricio Salazar Duque:** Resources, Visualization, Writing – review & editing. **Shengren Hou:** Data curation, Resources. **Peter Palensky:** Project administration, Resources, Supervision. **Pedro P. Vergara:** Methodology, Project administration, Resources, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset is cited in the paper, and the repository of the code is also provided in the paper.

Acknowledgments

This publication is part of the project ALIGN4Energy (with project number NWA.1389.20.251) of the research programme NWA ORC 2020 which is (partly) financed by the Dutch Research Council (NWO), The Netherlands. This research utilized the Dutch National e-Infrastructure with support from the SURF Cooperative, The Netherlands (grant number: EINF-5398).

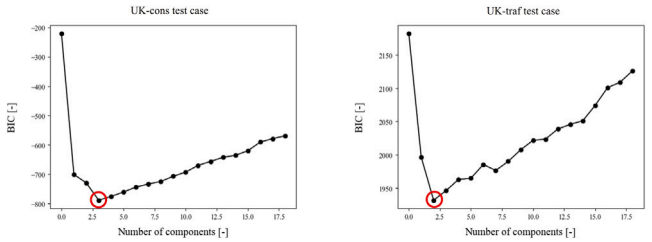


Fig. 9. Best number of components for timestep from 00:00 to 00:30 for UK-traf and UK-cons test cases.

Table 6
Structure of the generator of GAN-related models. The 's' is 1, 2, 3 for 60 min, 30 min, and 15 min resolution test cases, respectively.

Layer	Shape
Input Noise	$5 \times s$
First layer: Fully Connected + BatchNorm + LeakyReLU	$(5 \times s) \times (30 \times s)$
Second layer: Fully Connected + BatchNorm + LeakyReLU	$(30 \times s) \times (60 \times s)$
Last layer: Fully Connected + BatchNorm + Tanh	$(60 \times s) \times (24 \times s)$

Table 7
Structure of the discriminator (critic) of GAN-related models. The 's' is 1, 2, 3 for 60 min, 30 min, and 15 min resolution test cases, respectively.

Layer	Shape
First layer: Fully Connected + BatchNorm (LayerNorm) + LeakyReLU	$(24 \times s) \times (240 \times s)$
Second layer: Fully Connected + BatchNorm (LayerNorm) + LeakyReLU	$(240 \times s) \times (120 \times s)$
Last layer: Fully Connected + BatchNorm (LayerNorm) + Sigmod (No Simoid)	$(120 \times s) \times 1$

Appendix

A.1. Number of components of GMMs

The optimal number of components is determined by the BIC, which is defined as:

$$\mathrm{BIC} = \ln(N)k - 2\ln(\hat{L}), \tag{15}$$

where $\ln(N)$ is the natural logarithm of the number of observations, k is the number of parameters estimated by the model, $\hat{L}$ is the maximized value of the likelihood function for the model.

For each test case, the optimal number of components is chosen corresponding to the minimum BIC value for each individual timestep. Fig. 9 provides an illustration of this process, demonstrating how the optimal number of component for timesteps between 00:00 and 00:30 are determined for the 'UK-traf' and 'Uk-cons' test cases.

A.2. Deep generative model building and training

The architectural design of the generators for GAN, WGAN, and WGAN-GP models is shown in Table 6. In order to accommodate higher-resolution data, we expand the network's width accordingly. The term 's' is utilized as a scaling factor. For instance, the scaling factor for WGAN-GP is set to 3, consequently augmenting the first layer's dimensions to 15×90. The structure of the discriminator (also referred to as the 'critic' for WGAN and WGAN-GP models) is shown in Table 7. Analogous to the generators, 's' operates as a scaling parameter to adjust the network's width in line with the resolution. In the discriminator design, GAN employs BatchNorm and a Sigmoid activation function. However, in the WGAN and WGAN-GP models, BatchNorm is substituted with LayerNorm, and the final layer has no sigmoid function. The structure of VAE is shown in Table 8.

Table 8
Structure of VAE. The 's' is 1, 2, 3 for 60 min, 30 min, and 15 min resolution test cases, respectively.

Layer	Shape
Encoder	
First layer: Fully Connected + BatchNorm + LeakyReLU	$(24 \times s) \times (240 \times s)$
Second layer: Fully Connected + BatchNorm + LeakyReLU	$(240 \times s) \times (120 \times s)$
Mean layer: Fully Connected	$(120 \times s) \times (5 \times s)$
Log variance: Fully Connected	$(120 \times s) \times (5 \times s)$
Decoder	
First layer: Fully Connected + BatchNorm + LeakyReLU	$(5 \times s) \times (30 \times s)$
Second layer: Fully Connected + BatchNorm + LeakyReLU	$(30 \times s) \times (60 \times s)$
Last layer: Fully Connected + BatchNorm + Tanh	$(60 \times s) \times (24 \times s)$

In the training process of deep generative models, the parameters are set differently for transformer and consumer-level data. For transformer-level data, a learning rate of 0.002 and a batch size of 64 are employed. Additionally, we find there is no obvious difference using a larger learning rate for transformer-level data. Besides these, the weight clip value for WGAN is set at 0.05, while the penalty coefficient λ for WGAN-GP is set at 10. For consumer-level data, a larger learning rate is observed to yield better results. Consequently, a learning rate of 0.01 and a larger batch size of 180 are used in this case. For the training of VAE, we find the classical structure works well for transformer-level RLPs but fails to generate good results for consumer-level RLPs. To solve this problem, we adjust the original loss function by adding weight $\beta = 10$ to construction loss. The new loss function becomes: $\mathcal{L}_{\mathrm{VAE}}(\theta, \phi; x) = \beta \mathbb{E}_{z \sim Q_\phi(z|x)}[\log P_\theta(x|z)] - \mathrm{KL}(Q_\phi(z|x) \parallel P(z))$.

References

[1] Alicja Sikora, European Green Deal–legal and financial challenges of the climate change, in: Era Forum, Vol. 21, (4) Springer, 2021, pp. 681–697.
[2] Matthias Braubach, Arnaud Ferrand, Energy efficiency, housing, equity and health, Int. J. Public Health 58 (3) (2013) 331–332.
[3] Else Veldman, Power play: impacts of flexibility in future residential electricity demand on distribution network utilisation, 2013.
[4] Shengnan Shao, Manisa Pipattanasomporn, Saifur Rahman, Development of physical-based demand response-enabled residential load models, IEEE Trans. Power Syst. 28 (2) (2012) 607–614.
[5] Gianfranco Chicco, Roberto Napoli, Federico Piglione, Comparisons among clustering techniques for electricity customer classification, IEEE Trans. Power Syst. 21 (2) (2006) 933–940.
[6] Stefan Krauter, Simple and effective methods to match photovoltaic power generation to the grid load profile for a PV based energy system, Sol. Energy 159 (2018) 768–776.
[7] Wouter Labeeuw, Geert Deconinck, Residential electrical load model based on mixture model clustering and Markov models, IEEE Trans. Ind. Inform. 9 (3) (2013) 1561–1569.
[8] Edgar Mauricio Salazar Duque, Juan S Giraldo, Pedro P Vergara, Phuong H Nguyen, Anne van der Molen, JG Slootweg, Risk-Aware Operating Regions for PV-rich distribution networks considering irradiance variability, IEEE Trans. Sustain. Energy (2023).
[9] Ciaran Gilbert, Jethro Browell, Bruce Stephen, Probabilistic load forecasting for the low voltage network: forecast fusion and daily peaks, Sustain, Energy, Grids Netw. 34 (2023) 100998.
[10] B. Asare-Bediako, W.L. Kling, P.F. Ribeiro, Future residential load profiles: Scenario-based analysis of high penetration of heavy loads and distributed generation, Energy Build. 75 (2014) 228–238.
[11] Patrick McDaniel, Stephen McLaughlin, Security and privacy challenges in the smart grid, IEEE Secur. Priv. 7 (3) (2009) 75–77.
[12] Liander Open Data. https://www.liander.nl/partners/datadiensten/open-data/data.
[13] Household Data - Open Power System Data. https://data.open-power-system-data.org/household_data/2020-04-15.
[14] Smart Meter Data - London Households. https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households.
[15] Gianfranco Chicco, Overview and performance assessment of the clustering methods for electrical load pattern grouping, Energy 42 (1) (2012) 68–80.
[16] Jovana Forcan, Miodrag Forcan, Optimal placement of remote-controlled switches in distribution networks considering load forecasting, Sustain, Energy, Grids Netw. 30 (2022) 100600.

[17] Connor Shorten, Taghi M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (1) (2019) 1–48.
[18] Mohammad Navid Fekri, Ananda Mohon Ghosh, Katarina Grolinger, Generating energy data for machine learning with recurrent generative adversarial networks, Energies 13 (1) (2019) 130.
[19] Dengji Zhou, Shixi Ma, Jiarui Hao, Dong Han, Dawen Huang, Siyun Yan, Taotao Li, An electricity load forecasting model for Integrated Energy System based on BiGAN and transfer learning, Energy Rep. 6 (2020) 3446–3461.
[20] Chenlu Tian, Chengdong Li, Guiqing Zhang, Yisheng Lv, Data driven parallel prediction of building energy consumption using generative adversarial nets, Energy Build. 186 (2019) 230–243.
[21] J. Francou, D. Calogine, O. Chau, M. David, P. Lauret, Expanding variety of non-intrusive load monitoring training data: Introducing and benchmarking a novel data augmentation technique, Sustain, Energy, Grids Netw. 35 (2023) 101142.
[22] Christian Wagner, Christian Waniek, Ulf Häger, Modeling of household electricity load profiles for distribution grid planning and operation, in: 2016 IEEE International Conference on Power System Technology, POWERCON, IEEE, 2016, pp. 1–6.
[23] Francesco Lombardi, Sergio Balderrama, Sylvain Quoilin, Emanuela Colombo, Generating high-resolution multi-energy load profiles for remote areas with an open-source stochastic model, Energy 177 (2019) 433–444.
[24] Xiufeng Liu, Yanyan Yang, Rongling Li, Per Sieverts Nielsen, A stochastic model for residential user activity simulation, Energies 12 (17) (2019) 3326.
[25] Matteo Muratori, Matthew C Roberts, Ramteen Sioshansi, Vincenzo Marano, Giorgio Rizzoni, A highly resolved modeling technique to simulate residential power demand, Appl. Energy 107 (2013) 465–473.
[26] Ravindra Singh, Bikash C. Pal, Rabih A. Jabr, Statistical representation of distribution system loads using Gaussian mixture model, IEEE Trans. Power Syst. 25 (1) (2009) 29–37.
[27] Raoul Bernards, Johan Morren, Han Slootweg, Statistical modelling of load profiles incorporating correlations using copula, in: 2017 IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT-Europe, IEEE, 2017, pp. 1–6.
[28] Edgar Mauricio Salazar Duque, Pedro P Vergara, Phuong H Nguyen, Anne van der Molen, Johannes G Slootweg, Conditional multivariate elliptical copulas to model residential load profiles from smart meter data, IEEE Trans. Smart Grid 12 (5) (2021) 4280–4294.
[29] Johannes Einolander, Risto Lahdelma, Multivariate copula procedure for electric vehicle charging event simulation, Energy 238 (2022) 121718.
[30] Antoine Langevin, Mohamed Cheriet, Ghyslain Gagnon, Efficient deep generative model for short-term household load forecasting using non-intrusive load monitoring, Sustain, Energy, Grids Netw. 34 (2023) 101006.
[31] Yize Chen, Yishen Wang, Daniel Kirschen, Baosen Zhang, Model-free renewable scenario generation using generative adversarial networks, IEEE Trans. Power Syst. 33 (3) (2018) 3265–3275.
[32] Yuxuan Gu, Qixin Chen, Kai Liu, Le Xie, Chongqing Kang, GAN-based model for residential load generation considering typical consumption patterns, in: 2019 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference, ISGT, IEEE, 2019, pp. 1–5.
[33] Yi Wang, Gabriela Hug, Zijie Liu, Ning Zhang, Modeling load forecast uncertainty using generative adversarial networks, Electr. Power Syst. Res. 189 (2020) 106732.
[34] Yi Hu, Yiyan Li, Lidong Song, Han Pyo Lee, PJ Rehm, Matthew Makdad, Edmond Miller, Ning Lu, MultiLoad-GAN: A GAN-based synthetic load group generation method considering spatial-temporal correlations, 2022, arXiv preprint arXiv:2210.01167.
[35] Leijiao Ge, Wenlong Liao, Shouxiang Wang, Birgitte Bak-Jensen, Jayakrishnan Radhakrishna Pillai, Modeling daily load profiles of distribution network for scenario generation using flow-based generative network, IEEE Access 8 (2020) 77587–77597.
[36] Lin Lin, Cheng Chen, Boxu Wei, Hao Li, Jiancheng Shi, Jie Zhang, Nantian Huang, Residential electricity load scenario prediction based on transferable flow generation model, J. Electr. Eng. Technol. 18 (1) (2023) 99–109.
[37] Zhixin Pan, Jianming Wang, Wenlong Liao, Haiwen Chen, Dong Yuan, Weiping Zhu, Xin Fang, Zhen Zhu, Data-driven EV load profiles generation using a variational auto-encoder, Energies 12 (5) (2019) 849.
[38] Chenguang Wang, Ensieh Sharifnia, Zhi Gao, Simon H Tindemans, Peter Palensky, Generating multivariate load states using a conditional variational autoencoder, Electr. Power Syst. Res. 213 (2022) 108603.
[39] Zhe Wang, Tianzhen Hong, Generating realistic building electrical load profiles through the Generative Adversarial Network (GAN), Energy Build. 224 (2020) 110299.
[40] Weijie xia, Generative Models for Aggregated and Individual Residential Load Profile Generation. Online avaliable. https://github.com/xiaweijie1996/Generative-Models-for-Customer-Profile-Generation.
[41] Pedro P. Vergara, Generative Models for Aggregated and Individual Residential Load Profile Generation. Online avaliable. https://github.com/distributionnetworksTUDelft/Generative-Models-for-Customer-Profile-Generation.
[42] G. Valverde, A.T. Saric, V. Terzija, Probabilistic load flow with non-Gaussian correlated random variables using Gaussian mixture models, IET Gener., Transm. Distrib. 6 (7) (2012) 701–709.
[43] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[44] Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 214–223.
[45] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron C Courville, Improved training of wasserstein gans, in: Advances in Neural Information Processing Systems, vol. 30, 2017.
[46] Diederik P. Kingma, Max Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.
[47] Ludger Rüschendorf, On the distributional transform, Sklar's theorem, and the empirical copula process, J. Stat. Plan. Inference 139 (11) (2009) 3921–3927.
[48] Zhenyi Wang, Hongcai Zhang, Customized load profiles synthesis for electricity customers based on conditional diffusion models, 2023, arXiv preprint arXiv:2304.12076.
[49] Ryo Okano, Masaaki Imaizumi, Inference for projection-based wasserstein distances on finite spaces, 2022, arXiv preprint arXiv:2202.05495.
[50] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, Alexander Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (1) (2012) 723–773.
[51] Hoang Thanh-Tung, Truyen Tran, Catastrophic forgetting and mode collapse in GANs, in: 2020 International Joint Conference on Neural Networks, IJCNN, IEEE, 2020, pp. 1–10.