Delft University of Technology

Generating multivariate load states using a conditional variational autoencoder

Wang, Chenguang; Sharifnia, Ensieh; Gao, Zhi ; Tindemans, Simon H.; Palensky, Peter

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Generating multivariate load states using a conditional variational autoencoder ☆

Chenguang Wang, Ensieh Sharifnia, Zhi Gao, Simon H. Tindemans *, Peter Palensky

Department of Electrical Sustainable Energy, Delft University of Technology, Delft, The Netherlands

ARTICLE INFO

ABSTRACT

For planning of power systems and for the calibration of operational tools, it is essential to analyse system performance in a large range of representative scenarios. When the available historical data is limited, generative models are a promising solution, but modelling high-dimensional dependencies is challenging. In this paper, a multivariate load state generating model on the basis of a conditional variational autoencoder (CVAE) neural network is proposed. Going beyond common CVAE implementations, the model includes stochastic variation of output samples under given latent vectors and co-optimizes the parameters for this output variability. It is shown that this improves statistical properties of the generated data. The quality of generated multivariate loads is evaluated using univariate and multivariate performance metrics. A generation adequacy case study on the European network is used to illustrate model's ability to generate realistic tail distributions. The experiments demonstrate that the proposed generator outperforms other data generating mechanisms.

1. Introduction

In order to plan power systems and calibrate operational tools, it is essential to analyse system performance through a large range of representative scenarios [1,2]. Historical data is a key source of such scenarios, but when the available data set is too small for the desired application or when it cannot be made available for privacy reasons, it becomes valuable to have a model that can generate relevant data in abundant quantities. The challenge is that generated scenarios should embody both univariate distributions and multivariate inter-dependencies of the historical data [3].

A common approach has been to fit parametric probabilistic models to historical scenarios, especially load states. In [4], *Gaussian mixture models* (GMM) have been proposed to augment load data in distribution networks. Another study has introduced hidden Markov models to generate house-hold electric loads [5]. More recently, a load generator has been designed using time-varying queuing models [6]. Due to the curse of dimensionality, it is especially challenging to use parametric methods for generation of high-dimensional states [7, chapter 3]. Copula-based models are one class of generative models that does scale to higher dimensions, either using the Gaussian copula, or by 'stacking' copulas in a vine structure, possibly in combination with dimension-reduction schemes [3].

As vine-based copula models are highly asymmetric and therefore prone to bias, it is appealing to investigate 'native' high-dimensional models, such as neural networks. The *variational autoencoder* (VAE) [8] is an unsupervised machine learning model based on a deep neural network architecture. It has been successfully used in generating electricity load series, such as theft detection [9] and electric vehicle load profiles [10]. However, the validation of generated data often remains limited to visual comparisons, which is not straightforward for snapshots of larger and more complex electricity systems.

Moreover, most VAE implementations do not make full use [11] of the flexibility permitted by the mathematical framework in [8]. Output noise tuning and training [12,13] has only recently been considered, with a focus on image and video data sets. In power system related data generation applications, the output noise parameter is usually treated as a hyperparameter (i.e. a preset value that controls the learning process) [14] and noise is not actually inserted into samples [9,10,15–17].

This paper bridges those identified gaps by investigating the impact of output noise and its parameterization, and by analysing generated data using performance metrics. This is done for the VAE and the *conditional* VAE (CVAE), in the context of large-scale spatial load patterns of European countries. This lays the basis for wider applications of this
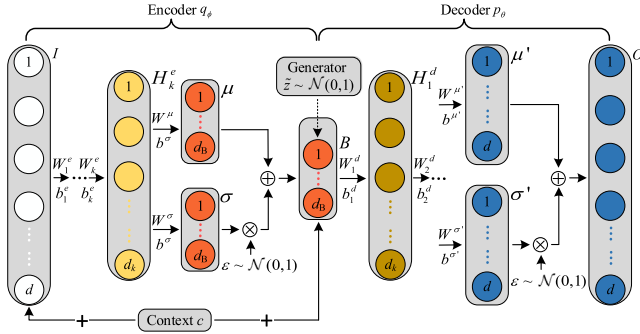

Fig. 1. Schematic of the CVAE.

method to synthetic load generation at lower aggregation levels, where consumption patterns are inherently more variable.

The main contributions of this paper are:

- We show how a sample-dependent output noise parameter can be co-optimized in the training process and how this noise is used in the generative process.
- We put forward a set of data quality metrics for generative models, consisting of three statistical tests for univariate distributions and multivariate dependencies.
- We introduce a simple multi-area adequacy assessment model that is used to test tail distributions.
- Through comprehensive experiments, we show the performance and practicality of VAE- and CVAE-based load generators in comparison with *Gaussian copula* and *conditional generative adversarial network* (cGAN) models.

2. Data generation mechanism

In this section, a representative multivariate load state generation mechanism is proposed, based on the *conditional variational autoencoder* (CVAE).

2.1. CVAE-based generative model

The CVAE is a neural network architecture that is trained to learn the salient features of historical data by mapping (*encoding*) historical system states onto a lower-dimensional latent space where the latent distribution is approximately normal, and transforming latent vectors back (*decoding*) into a high-dimensional state space [18]. The decoder is used in conjunction with contextual information c to generate representative states (which can be omitted to obtain a regular VAE model). Consequently, the model is able to generate samples with a similar distribution to the historical data, by transforming normally distributed samples in the latent space back to the data space. We note that the latent (i.e. hidden) representation of a data point is used solely to facilitate reconstruction and synthesis. It does not need to be imbued with a particular meaning.

The structure of the CVAE algorithm is depicted in Fig. 1. The *encoder* maps the d-dimensional input data x to the code z in the lower-dimensional latent space through k hidden layers H_l^e, $l = 1, \ldots, k$. Weight matrices W_l^e, bias vectors b_l^e and the context c are utilized in the encoding process as

$$\begin{pmatrix} \mu \\ \sigma \end{pmatrix} = \begin{pmatrix} W^\mu \\ W^\sigma \end{pmatrix} (a(W_k^e(\ldots a(W_1^e(x,c) + b_1^e) \ldots) + b_k^e))$$
$$+ \begin{pmatrix} b^\mu \\ b^\sigma \end{pmatrix}, \tag{1a}$$

$$z = \mu + \epsilon \odot \sigma, \tag{1b}$$

where a represents an element-wise nonlinear activation function. Vectors μ and σ parameterize an input-dependent normal distribution in the latent space. The output z is sampled accordingly, using ϵ, a vector that is sampled from a standard normal distribution, and the Hadamard product $\odot$.

Mirroring the encoder network, the decoder maps the sampled latent space code z to the d-dimensional output data $\hat{x}$ using

$$\begin{pmatrix} \mu' \\ \sigma' \end{pmatrix} = \begin{pmatrix} W^{\mu'} \\ W^{\sigma'} \end{pmatrix} (\ldots a(W_1^d(z,c) + b_1^d) \ldots) + \begin{pmatrix} b^{\mu'} \\ b^{\sigma'} \end{pmatrix}, \tag{2a}$$

$$\hat{x} = \mu' + \epsilon \odot \sigma', \tag{2b}$$

where W_l^d and b_l^d denote weight matrices and bias vectors for decoding, respectively. μ' and σ' parameterize a z-dependent normal distribution in the x space.

2.2. Training and generation process

In the training stage, the whole structure of the CVAE model is utilized as Fig. 1. Weight matrices W and bias vectors b are updated in an iterative way with the goal of minimizing the loss function [18]

$$\mathcal{L} = \mathcal{L}_{D_{KL}} + \mathcal{L}_{Re}. \tag{3}$$

The *Kullback–Leibler loss* $\mathcal{L}_{D_{KL}} = \sum_i D_{KL}(q_\phi(z|x_i) \parallel p(z))$ is the sum over all training data points x_i (assumed i.i.d.) of the Kullback–Leibler divergence between that point's posterior distribution $q_\phi(z|x_i)$ and the prior distribution $p(z)$ (chosen as the standard normal distribution). The posterior distribution $q_\phi(z|x_i)$ is determined by the parameters ϕ of the encoder network and represents the mapping of the point x_i into a normal distribution in the latent space using (1a) and (1b). As the Kullback–Leibler divergence between two normal distributions can be evaluated directly [14], the Kullback–Leibler loss is computed as

$$\mathcal{L}_{D_{KL}} = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{d} (-1 + \sigma_{i,j}^2 + \mu_{i,j}^2 - \log \sigma_{i,j}^2), \tag{4}$$

where n denotes total number of observations used for training and (μ_i, σ_i) are evaluated for data point x_i and condition c using (1a).

The *reconstruction loss* $\mathcal{L}_{Re}$ stands for the negative log-likelihood of reconstructing the inputs x_i via their latent space codes and the decoder that is parameterized by θ. The reconstruction loss is thus computed as

$$\mathcal{L}_{Re} = - \sum_{i=1}^{n} \mathbb{E}_{Z \sim q_\phi(z|x_i)}[\log_{P_\theta}(x_i|Z)] \tag{5}$$

$$\approx \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{d} ((x_{i,j} - \mu'_{i,j})^2 / \sigma'^2_{i,j} + \log \sigma'^2_{i,j}) + \frac{nd}{2} \log 2\pi,$$

where the final step involves a single-point approximation of the expectation and (μ'_i, σ'_i) are obtained from the randomly generated latent code $z(x_i)$ and the condition c using (2a). During training, the full-sample sum in loss functions (4) and (5) are replaced by batch-sample averages. The constant $\frac{nd}{2} \log 2\pi$ of $\mathcal{L}_{Re}$ is omitted.

After the training process, only the decoder part of the trained CVAE network is utilized to generate data. Latent space codes $\tilde{z}$ are sampled from the standard normal distribution $\mathcal{N}(0, I)$ (see Fig. 1). Then, data space samples $\tilde{x}$ are sampled from distribution $\mathcal{N}(\mu'(\tilde{z}, c), \sigma'(\tilde{z}, c))$, whose parameters are determined by $\tilde{z}$ and c using (2a). We note that although the amount of available training data determines the information contained within the model, there is no limit to the amount of data that can be generated.

In this way, a complex data distribution in the x space is constructed as a continuous superposition of normal distributions that are parameterized by the normally distributed coordinate z. Using the procedure above, the encoder and decoder networks are trained to adapt any distribution to this normally distributed latent space. We note that other distributions besides the normal distribution can be used as the prior for the latent space coordinate z [19,20] – selecting the best latent space representation for a particular class of problems remains an open research problem.

Table 1
Overview of model permutations used in experiments.

Strategy (with β)	Objective function	Generation
Auto σ', Noisy	$\beta\mathcal{L}_{D_{KL}} + \mathcal{L}_{Re}$	$\mathcal{N}(\mu'(\tilde{z},c),\sigma'(\tilde{z},c))$
Auto σ', Noise free	$\beta\mathcal{L}_{D_{KL}} + \mathcal{L}_{Re}$	$\mu'(\tilde{z},c)$
Fixed σ', Noisy	$\beta\mathcal{L}_{D_{KL}} + \tilde{\mathcal{L}}_{Re}$	$\mathcal{N}(\mu'(\tilde{z},c),sI)$
Fixed σ', Noise free	$\beta\mathcal{L}_{D_{KL}} + \tilde{\mathcal{L}}_{Re}$	$\mu'(\tilde{z},c)$

2.3. Network and output noise co-optimization strategy

It is common for CVAE implementations to generate data $\tilde{x}$ not by sampling from $\mathcal{N}(\mu'(\tilde{z},c),\sigma'(\tilde{z},c))$ via (2b), but by directly using the mean value $\mu'(\tilde{z},c)$ (the maximum likelihood sample). Moreover, the standard deviation σ' is not co-optimized in the training process of (3), but considered a hyperparameter that fixes $\sigma'_{i,j} = s$ identically in all dimensions, so that (5) can be replaced by

$$\tilde{\mathcal{L}}_{Re} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{d}\frac{(x_{i,j}-\mu'_{i,j})^2}{s^2}. \qquad (6)$$

In contrast, we investigate the model in which the parameters σ' of the output noise distribution are co-optimized as a function of z during training, as was recently also (independently) proposed in [13]. In addition, we explicitly add output noise $\epsilon \odot \sigma'(\tilde{z},c)$ to the generated data. To compare the different approaches, the quality of the generated data is evaluated under all four combinations (Table 1): whether σ' is co-optimized in the training stage (Auto σ') or set to a fixed value (Fixed σ'); whether the noise $\epsilon \odot \sigma'(\tilde{z},c)$ is added to the outputs (Noisy) or not (Noise free).

2.4. Loss function weight tuning strategy

The two loss terms have opposing effects. The Kullback–Leibler loss $\mathcal{L}_{D_{KL}}$ ensures a good fit with the prior distribution that samples are generated from, thus suppressing spurious generated points at the expense of 'smoothing' the output. The reconstruction loss $\mathcal{L}_{Re}$, on the other hand, promotes exact reconstruction of the training data. In this paper, in addition to the output noise, we also study the effect of a heuristic weighting factor β [21] for the Kullback–Leibler loss term $\mathcal{L}_{D_{KL}}$ on statistical properties of the generated data.

All aforementioned combined strategies are explicated in Table 1. Their impacts to the quality of generations will be investigated in the following sections. Particularly, the settings of standard deviation σ' and weight β influence the objective function in the training process and will ultimately affect the generated data. On the other hand, the use of output noise, $\epsilon \odot \sigma'(\tilde{z},c)$, will directly impact the data generation stage.

3. Case study: European load data

In this section, the performance of our proposed CVAE-based generative model is analysed using a European load data set. This is done with three data quality metrics that measure univariate distributions and multivariate dependencies. Impacts to the quality of generated data are investigated under the experimental settings in Table 1, using both conditional and regular VAEs.

3.1. Data source and generation

Historical hourly load data for 32 European countries between 2013 and 2017 was obtained from the Open Power System Data platform [22] (package version *2019-06-05*). Columns of AL, CS, CY, GB, TR and UA were dropped for incomplete records. The historical data was randomly split into training and test set in blocks of one week with the proportion of 4:1 (35,148 training and 8569 test samples). The training set was min–max normalized before being fed into the CVAE

model and the inverse transformation was applied to generate samples. The context information c is the hour of day. Both total and hourly volumes of the generated data are the same with the training data set, in order to balance them for visual and statistical analysis. However, we emphasize that the purpose of constructing such a generative model is to have the ability to generate limitless non-repeating data, e.g. for reducing the risk of overfitting in downstream machine learning tasks.

The parameters of the generative models were tuned for optimal performance, for both the VAE and the CVAE. The network contained 2 hidden layers in the encoder with dimensions of 24 and 16, respectively; the bottleneck layer had 8 nodes (8-dimensional latent vector). The decoder also had 2 hidden layers with the same dimensions in reverse order. Comparisons against 4-neuron and 16-neuron bottleneck revealed that a smaller bottleneck results in excessive loss, whereas a larger bottleneck insufficiently forces the network to learn features. In the CVAE model, the hourly time-of-day was encoded cyclically using sine/cosine representation.

The ReLU activation function was used, except for the generation of μ and σ leading up to the bottleneck and output layers. The *adaptive moment estimation* (Adam) weight optimizer [23] was utilized with default settings to iteratively optimize the value of weight matrices W and bias vectors b. The batch size and learning rate related parameter α for training was 64 and 10^{-4} respectively and 20,000 training iterations were used. Training and data generation of the model was conducted in Python using `tensorflow` on the Google Colab environment using the GPU option. The code used for this paper is available for download [24, 25].

3.2. Data quality metrics

To test a generative model's ability to reproduce the features of historical data, especially in high dimensions, statistical tests are required. Three tests are put forward to examine different aspects of the generated data set, in comparison with the historical data.

3.2.1. Kolmogorov–Smirnov test for univariate marginal distributions

We used the two-sample Kolmogorov–Smirnov (K–S) test [26] to see whether the generated data was able to reproduce the marginal load distributions for each of the countries in the data set. For a given output dimension (load in a single country), historical and generated data are compared. Under the null hypothesis that historical and generated data are drawn from the same model, the p-values should follow a uniform distribution. In other words, when the historical data is compared against itself, the cumulative distribution of p-values should lie on the diagonal. Thus, for generative models, the closer the cumulative distribution of p-values lies to the diagonal, the higher the similarity between the two distributions.

Clearly, the models are unlikely to exactly reproduce the historical distribution, thus large deviations from the ideal curve will show up for large-sample tests. Nevertheless, to analyse the degree of performance of various models, we use repeated tests on smaller sample sets that result in clear differentiation, as in [3]. In this paper, 0.5% of the data set, i.e. 176 data points out of 35,148, were randomly drawn from training and generated data set, and then a p-value was calculated accordingly. This process was repeated 5,000 times for each country and a curve was constructed from all p-values.

3.2.2. Autoencoder-based point-wise test for multivariate dependencies

Autoencoder (AE) neural networks have been proven to be highly sensitive anomaly detectors [27]. Unlike (C)VAE networks, AEs have no stochastic layers and only minimize the reconstruction loss $r = \sum_i \|x_i - \hat{x}_i\|^2/d$. An AE learns to compress and decompress the data based on properties of the training set. As a result, data points with dependencies that deviate substantially from that in the training set tend to have higher reconstruction errors.
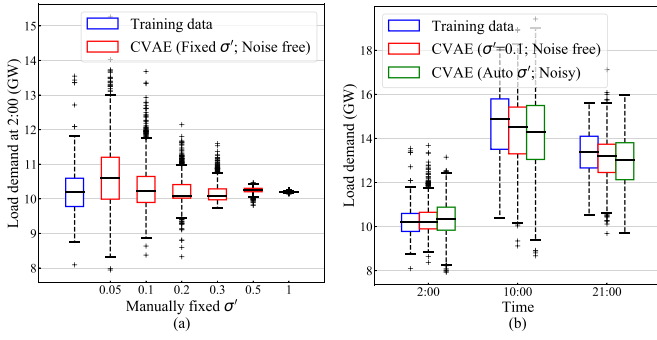
Fig. 2. (a) Box plots of the original and generated load values in the Netherlands, based on 1459 data points at 2:00. Data was generated by CVAE using 'Fixed σ', Noise free' approach with different σ'. (b) Distribution comparison of the original and generated load data in the Netherlands, based on 1,459 (2:00), 1,465 (10:00) and 1,465 (21:00) data points, respectively. Data was generated by CVAE using 'Fixed σ', Noise free' method ($\sigma' = 0.1$) and 'Auto σ', Noisy' scheme.

A separate AE network was trained for this test, with hyperparameters equal to that of the CVAE model, except for the stochastic layers and objective function. Reconstruction errors of all data points (historical or generated) are plotted as cumulative distributions for easy comparison. As a test for overfitting of the autoencoder on the training data, the autoencoder test was performed on the training and test data. The two distributions visually overlapped, suggesting this is not a concern.

3.2.3. Energy test for multivariate dependencies of population

Another two-sample test, the energy test [28], was conducted to examine whether the *multivariate dependencies* of the population were well acquired from the training set. The energy test, computed using the PyTorch library torch-two-sample [29] uses a user-specified number of permutations (200 was used) to calculate a *p*-value. The same as for the K–S test, we used random subsets of 0.5% of the generated population and historical population. We repeated this process 1,000 times to draw a distribution of *p*-values and compare it with the uniform distribution (which would be expected if the data was drawn from the historical distribution).

3.3. Results

3.3.1. Visual comparison of univariate distributions

In this experiment, the CVAE with fixed σ' and no output noise was used to generate 1459 load demands, conditioned on the time 2:00. Results for the Netherlands are shown in Fig. 2a, for various values of σ'. As the output noise assumed in training increases, the variability of the generated points decreases (because noise is not actually added). When $\sigma' = 0.1$, the distribution of generations is the closest to that of historical data. This setting will be used for all further experiments with fixed σ'.[1]

Fig. 2b further compares data generated using the 'Fixed σ'' and 'Auto σ', Noisy' schemes and the training data. Conditioning on 2:00, 10:00 and 21:00 was performed, and results are shown for the Netherlands. Both methods are able to qualitatively reproduce the features of the data.

3.3.2. Multivariate correlations

The top row of Fig. 3 shows the loads of all countries for 10 different snapshots at 2:00, relative to the mean load in those countries
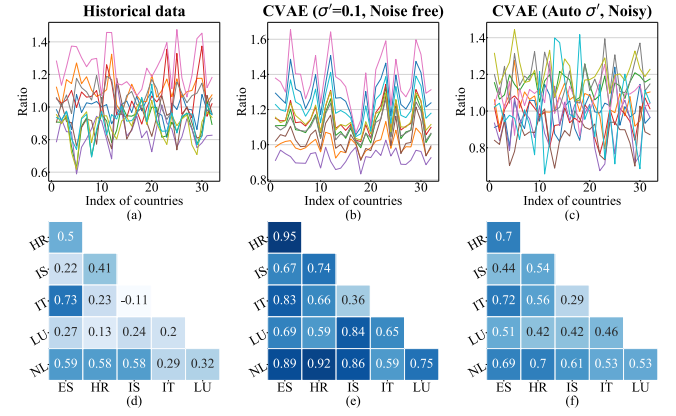
---

[1] Note that we only fix a single parameter in this case. An approximate visual match of the box plots is a necessary condition for a good overall fit, justifying the choice $\sigma' = 0.1$ for this comparison.

Fig. 3. (a), (b) and (c) display 10 typical ratios of 32 countries' historical and generated data to the historical mean values at 2:00. (d), (e) and (f) demonstrate the Pearson correlation coefficient matrices of 6 (out of 32) countries' historical and generated data at 2:00. The horizontal and vertical dimensions in the matrices are Spain (ES), Croatia (HR), Iceland (IS), Italy (IT), Luxembourg (LU) and the Netherlands (NL).

at 2:00. Compared to historical data (a) and the noisy generator (c), samples generated by the noise-free generator clearly show higher correlations between countries. This is confirmed by the correlation analysis between six countries in the bottom row of Fig. 3. By omitting output noise, the noise-free generator generated (too) highly correlated samples.

Sensitive experiments for the multivariate dependencies will be conducted in the following sections using the autoencoder-based pointwise test. The accurate representation of multivariate dependencies will be important for the analysis of supply shortfalls in Section 4.

3.3.3. Influence of noise generation

In this experiment, the influence of the four strategies listed in Table 1 were tested with $\beta = 1$. Results for the statistical tests described in Section 3.2 are shown in the first column of Fig. 4. The K–S test results and autoencoder results show that the inclusion of output noise is essential to improve marginal distributions (Fig. 4a) and increase output variability to the level of the historical data (Fig. 4e). In addition, the autoencoder and energy tests show that automatic tuning of the noise strength (Auto σ') is essential to improve the multivariate dependencies of the generated samples. Together, this experiment shows that the ('Auto σ', Noisy') generator outperforms the other approaches listed in Table 1. This was to be expected given the mathematical theory behind the CVAE (which includes noise), but is at odds with common implementations.

3.3.4. Comparison between conditional and regular VAE

In the second column of Fig. 4, the performance of the CVAE and VAE models (with $\beta = 1$) is compared. The CVAE model slightly outperforms the VAE model in all categories. One possible explanation is that the CVAE model has access to the context c (time of day), which effectively increases the dimension of the latent space. Because of its better performance, we continue using the CVAE model in subsequent experiments, but the results suggest that a VAE model delivers comparable performance, and may be preferable when no natural conditioning variable is available.

3.3.5. β Sensitivity test

The third column of Fig. 4 shows the impact of β (values 1, 3, 10) on the performance of the CVAE (Auto σ', Noisy) model. As β is increased, the performance on the K–S test (Fig. 4c) improves, indicating an improved ability to learn marginal distributions. On the other hand, performance on the autoencoder test (Fig. 4g) worsens, suggesting that points 'outside' of the training point cloud are generated for large β.
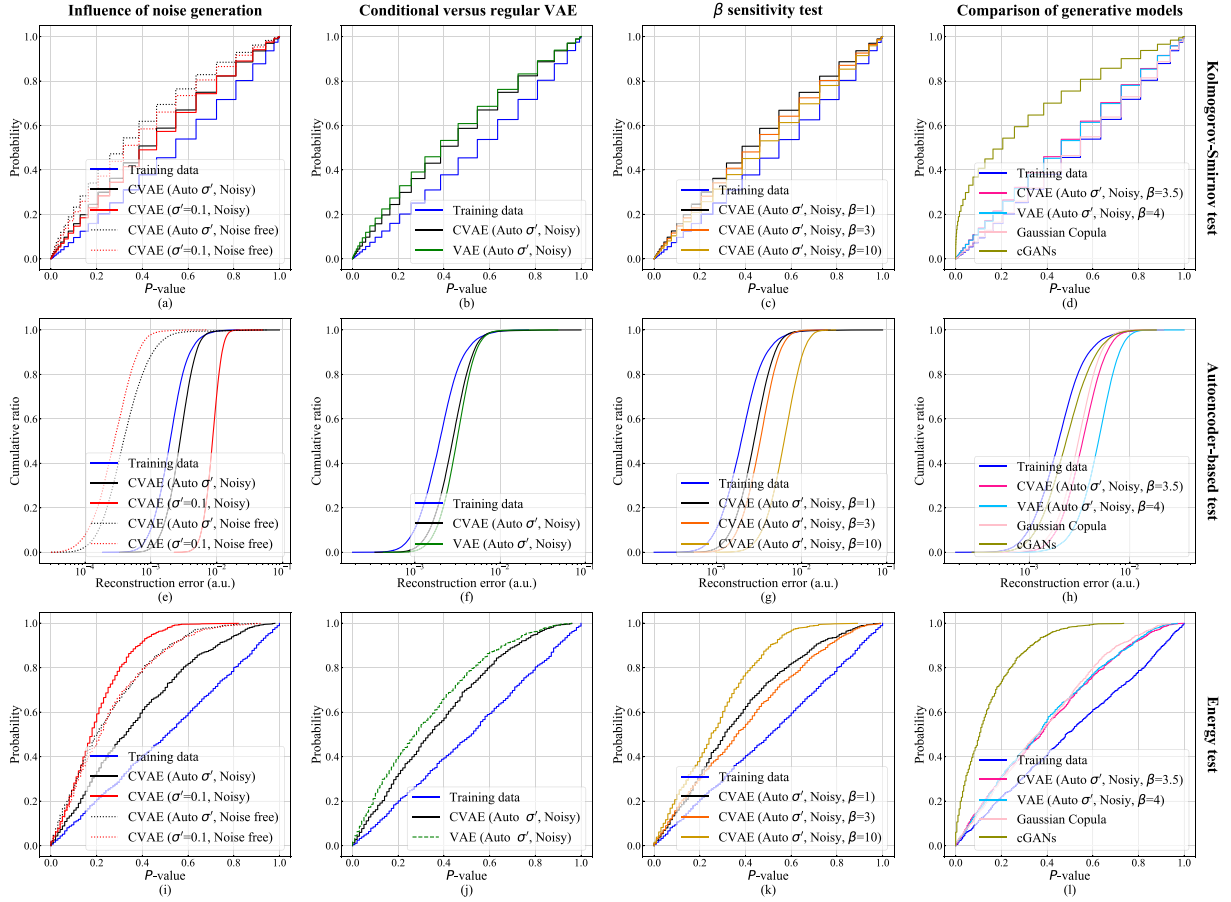
Fig. 4. Results of statistical tests. Each column denotes a set of experiments (noise generation, training condition, value of β and model family). The three rows depict results for the three tests described in Section 3.2.

Finally, the energy test (Fig. 4k) indicates that a moderate value of β can strike a balance between the opposing requirements: the curve for $\beta = 3$ is closest to the desired result. Nevertheless, depending on the application, it may be desirable to choose β larger or smaller.

3.3.6. Comparison of generative models

In the fourth column of Fig. 4, the quality of data sampled from different generative models was investigated. The values of β for CVAE and VAE models (both Auto σ', Noisy) were tuned for optimal performance on the energy test (see previous section). In addition, *Gaussian copula* [30] and cGAN [31] models were included for comparison. The basic cGAN model was modified to use Wasserstein losses [32]. Both its generator and discriminator are deep neural networks; each has two hidden layers of 256 neurons, all activated with LeakyReLU ($\alpha = 0.2$) except in the output layers, where linear and sigmoid activation functions are used for the generator and discriminator. Weights of the neurons are optimized with *root mean square propagation* (RMSprop) available from python package Keras.

The K–S test shows the outstanding ability of the Gaussian copula model to reproduce marginal load distributions (a design feature of copula models [3]). This model also shows competitive performance on the autoencoder and energy tests. However, it will become clear in Section 4 that its tail-performance is worse than that of the (C)VAE models. The cGAN model shows the best performance on the autoencoder test, indicative of its ability to generate samples with realistic features. However, the model significantly underperforms on the K–S and energy tests, which suggests that the generated samples, though 'realistic', are unevenly distributed through the space of possible states. The optimized CVAE and VAE models are competitive on all three tests, with the CVAE model slightly outperforming the VAE model.

4. Multi-area adequacy assessment

Next, we investigated the performance of the load generation mechanisms by using it for a multi-area adequacy assessment study, based on the ENTSO-E Mid-term Adequacy Forecast 2020 (MAF2020) [33]. Multi-area adequacy assessment measures the sufficiency of generating capacity compared with the load on each of the nodes in the power system under transmission constraints. This can be considered a stress test of the generative model, as the outcomes are sensitive to high-load events (tail distributions) *and* their dependencies between countries.

Monte Carlo simulations were used to estimate *Loss Of Load Expectation* (LOLE [h/year]) and *Expected Energy Not Served* (EENS [MWh/year]). LOLE is the expected number of hours per year during which the supply does not meet demand. EENS is the expected amount of energy demand per year that cannot be supplied. Parameters from the MAF2020 study were used to construct a model for generating capacity and net transfer capacities between countries. They were combined with generated and historical load data to define a probabilistic model for the Monte Carlo simulations. We emphasize that the model thus constructed is not meant to be an accurate representation of the European grid, but a stylized problem that serves as a comparative testing ground for the generative models.

4.1. Multi-area adequacy assessment structure

We consider the network as a directed graph (to allow for asymmetric flow limits) where nodes are zones, edges are connections between zones, and edge capacities are transfer capacities. Each sampled state w is represented by the available generating capacity $\overline{g}_i^w$ and demand d_i^w of each node i. Based on the flow constraints and dispatching policy,

Table 2
Calculated risks of selected countries (with and without interconnection) using historical data and all generative models.

Country	LOLE (h/y)					EENS (MWh/y)				
	Historical data	CVAE	VAE	Gaussian copula	cGAN	Historical data	CVAE	VAE	Gaussian copula	cGAN
AT	0.03(2)	0(0)	0.18(4)	0.32(5)	0.05(2)	4(2)	0(0)	33(11)	42(10)	10(5)
NL	0.74(8)	0.12(3)	0.80(8)	4.6(2)	3.6(2)	119(17)	33(11)	300(45)	1135(67)	1743(129)
UK	37.8(6)	50.6(7)	54.1(7)	50.2(7)	223.6(14)	$5.20(10)\cdot10^4$	$8.77(15)\cdot10^4$	$1.16(2)\cdot10^5$	$7.75(14)\cdot10^4$	$4.99(4)\cdot10^5$
AT (island)	0.74(8)	1.01(9)	0.88(9)	0.80(8)	0.61(7)	221(33)	435(54)	334(45)	273(40)	255(41)
NL (island)	63.8(7)	65.2(8)	69.2(8)	69.4(8)	99.7(9)	$4.13(7)\cdot10^4$	$4.35(7)\cdot10^4$	$4.73(7)\cdot10^4$	$4.61(7)\cdot10^4$	$6.74(8)\cdot10^4$
UK (island)	1026(3)	982(3)	884(3)	1033(3)	1965(4)	$4.28(2)\cdot10^6$	$4.24(2)\cdot10^6$	$3.88(2)\cdot10^6$	$4.38(2)\cdot10^6$	$1.078(3)\cdot10^7$

the consumed power p_i^w and load curtailment c_i^w for each node can be calculated, related by

$$c_i^w = \max(0, d_i^w - p_i^w). \tag{7}$$

We determine c_i^w (and implicitly p_i^w) by solving a quadratic problem with variables $\tilde{c}_i$ (curtailment) and $\tilde{f}_{ij}$ (flows), which aims to minimize the total load curtailments and assumes that curtailments are balanced between areas [34], relative to the demand in that area:

$$\vec{c}^w = \arg\min_{\tilde{f},\tilde{c}} \sum_{i\in\mathcal{N}} \frac{1}{2d_i^w} \tilde{c}_i^2 + \tilde{c}_i \tag{8}$$

$$\underline{f_{ij}} \le \tilde{f}_{ij} \le \overline{f_{ij}}, \qquad \forall(ij)\in\mathcal{L} \tag{9}$$

$$0 \le \tilde{c}_i \le d_i^w, \qquad \forall i \in \mathcal{N} \tag{10}$$

$$d_i^w - \overline{g}_i^w \le \sum_{j<i} \tilde{f}_{ji} - \sum_{j>i} \tilde{f}_{ij} + \tilde{c}_i \le d_i^w, \qquad \forall i \in \mathcal{N} \tag{11}$$

Here, $\mathcal{L}$ and $\mathcal{N}$ are the sets of connections (from i to j with $i < j$) and areas respectively. Constraints on power flow $\tilde{f}_{ij}$ from node i to node j are given in (9); (10) limits curtailment and (11) enforces flow and generating power constraints. The objective function (8) has positive definite structure and the constraints are linear, so this optimization problem is strictly convex and has a unique solution. This optimization problem was solved using the python package quadprog [35].

4.2. Power system model

A European adequacy assessment model was developed, based on the target year 2025 data from the ENTSO-E MAF2020 [33]. The net transfer capacities between countries are defined as the summation of transfer capacities between their constituent zones, as reported in the MAF2020. Since details of generators and unit capacities are not reported in the released dataset, we model the total generating capacity and the unit capacities in each country as follows. The assumed generating capacity of each country is a summation of conventional generating capacity in its zone(s) plus 5% of nameplate wind power capacity. Unit sizes are set per country as the closest value under 500 MW that is a divisor of the generating capacity; a unit availability of 83% is used. Cyprus has no connection to other countries, so a unit capacity of 95 MW is used to avoid excessive outages.

4.3. Multi-area adequacy assessment results

To compare the CVAE, VAE, Gaussian Copula, and cGAN generators, they were trained on historical load data from 2017 and 2018 for 35 countries, retrieved from the Open Power System Data Platform ([22]; columns for CS, IS and UA were omitted). Each model was used to generate 100,000 random load samples. The 'Auto σ', Noisy' setting was utilized for the CVAE and VAE models, and β was set to 10 for improved reproduction of the marginal distributions. For each model, 1,000,000 Monte Carlo generation samples were drawn and combined with random load samples to estimate the LOLE in each country. Fig. 5 depicts the estimated LOLE values for all generative methods and historical data. The area of each sector of the disc represents the LOLE obtained using a particular load states generating model. Numerical results for three countries with low (Austria, AT), medium
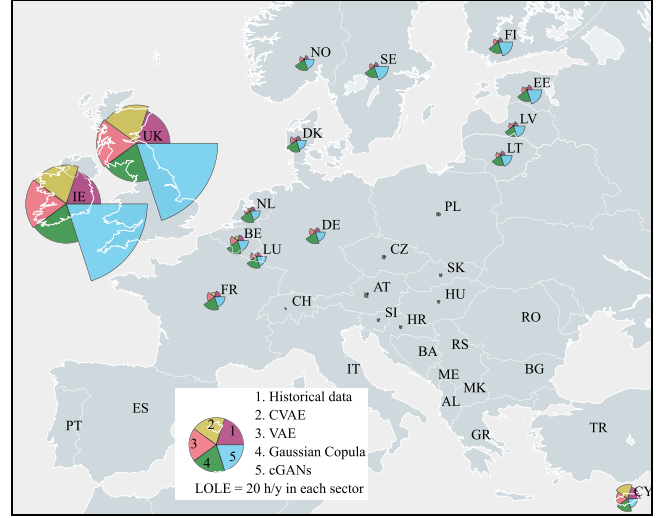

Fig. 5. Comparison of LOLE estimates using historical load data and generative load. The area of each sector of the disc represents the LOLE of the corresponding model (20 h/y shown for scale in legend).

(The Netherlands, NL) and high (UK) risk levels are shown in Table 2. Standard errors for the least significant digits are shown in parentheses. Moreover, to investigate the beneficial effect of interconnection – and therefore the importance of accurate multivariate modelling – risks for these countries are also reported in the absence of interconnection ('island').

By design, the Gaussian copula reproduces the marginal distributions of the historical data. Therefore, the calculated risk for *islanded* systems is consistent with those for historical data. However, the results demonstrate that this model tends to overestimate risks for interconnected nodes (countries). The cGAN generative model tends to cause an overestimation of risks with *both* islanded and interconnected nodes, sometimes very significantly (e.g. the LOLE values for the UK and Ireland). In comparison, both the VAE and CVAE models generate data that results in risk estimates that are closer to those observed using historical data, although deviations exist from country to country. This suggests both models are able to substantially represent the multivariate tail distribution of the historical data.

The capability of generating load conditioned on hours is an additional advantage of CVAE in comparison with VAE in the adequacy assessment context. Load curtailments usually accrue during high load hours. So, time of day could be used as a control variable for an importance sampling Monte Carlo scheme that preferentially samples load states at high load hours and compensates for the resulting bias by sample re-weighting.

5. Conclusions and future work

In this paper, we have investigated the performance of CVAE- and VAE-based models to generate multivariate load states. Our inclusion

of (1) sample noise in the generator and (2) co-optimized output noise parameters results in generated samples that show better marginal distributions and dependencies, when compared with common CVAE implementations (fixed noise parameter, noise omitted from the generator). A loss weighting factor β (hyperparameter) can be used to tune the performance of the model. Performance was tested using three statistical tests and in a Monte Carlo generation adequacy study. The (C)VAE based models significantly outperformed Gaussian copula and cGAN models on at least one of these tests and were competitive on all others.

With access to contextual information, the CVAE model slightly outperformed the VAE model. Moreover, such information can be used for targeted analysis, e.g. as part of a Monte Carlo importance sampling scheme that selects specific hours of the day.

In future work, we will further investigate the universality of our proposed load generation scheme by applying it to lower load aggregation levels, such as household-level data. At this level, the privacy-preserving nature of synthetic data becomes very beneficial and should be carefully tested in addition to the distributional aspects.

CRediT authorship contribution statement

Chenguang Wang: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Visualization. **Ensieh Sharifnia:** Methodology, Software, Data curation, Writing – original draft, Visualization. **Zhi Gao:** Methodology, Software, Writing – review & editing. **Simon H. Tindemans:** Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision. **Peter Palensky:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition.

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.

Acknowledgements

References

[1] H. Bloomfield, D. Brayshaw, A. Troccoli, C. Goodess, M. De Felice, L. Dubus, P. Bett, Y.-M. Saint-Drenan, Quantifying the sensitivity of European power systems to energy scenarios and climate change projections, Renew. Energy 164 (2021) 1062–1075.

[2] P. Panciatici, G. Bareux, L. Wehenkel, Operating in the fog: Security management under uncertainty, IEEE Power Energy Mag. 10 (5) (2012) 40–49.

[3] I. Konstantelos, M. Sun, S.H. Tindemans, S. Issad, P. Panciatici, G. Strbac, Using vine copulas to generate representative system states for machine learning, IEEE Trans. Power Syst. 34 (1) (2019) 225–235.

[4] M.-S. Kang, C.-S. Chen, Y.-L. Ke, C.-H. Lin, C.-W. Huang, Load profile synthesis and wind-power-generation prediction for an isolated power system, IEEE Trans. Ind. Appl. 43 (6) (2007) 1459–1464.

[5] T. Zia, D. Bruckner, A. Zaidi, A hidden Markov model based procedure for identifying household electric loads, in: IECON 2011-37th Annual Conference of the IEEE Industrial Electronics Society, IEEE, 2011, pp. 3218–3223.

[6] F.B. dos Reis, R. Tonkoski, T.M. Hansen, Synthetic residential load models for smart city energy management simulations, IET Smart Grid 3 (3) (2020) 342–354.

[7] S. Theodoridis, Machine Learning: A Bayesian and Optimization Perspective, second ed., Academic Press, ISBN: 978-0-12-818803-3, 2020, pp. 67–120.

[8] D.P. Kingma, M. Welling, Auto-encoding variational bayes, 2013, arXiv preprint arXiv:1312.6114.

[9] X. Gong, B. Tang, R. Zhu, W. Liao, L. Song, Data augmentation for electricity theft detection using conditional variational auto-encoder, Energies 13 (17) (2020) 4291.

[10] Z. Pan, J. Wang, W. Liao, H. Chen, D. Yuan, W. Zhu, X. Fang, Z. Zhu, Data-driven EV load profiles generation using a variational auto-encoder, Energies 12 (5) (2019) 849.

[11] R. Yu, A tutorial on VAEs: From Bayes' rule to lossless compression, 2020, arXiv preprint arXiv:2006.10273.

[12] S. Lin, S. Roberts, N. Trigoni, R. Clark, Balancing reconstruction quality and regularisation in elbo for vaes, 2019, arXiv preprint arXiv:1909.03765.

[13] O. Rybkin, K. Daniilidis, S. Levine, Simple and effective VAE training with calibrated decoders, in: International Conference on Machine Learning, PMLR, 2021, pp. 9179–9189.

[14] C. Doersch, Tutorial on variational autoencoders, 2016, arXiv preprint arXiv:1606.05908.

[15] C. Mylonas, I. Abdallah, E. Chatzi, Conditional variational autoencoders for probabilistic wind turbine blade fatigue estimation using supervisory, control, and data acquisition data, Wind Energy 24 (2021) 1122–1139.

[16] Y. Qi, W. Hu, Y. Dong, Y. Fan, L. Dong, M. Xiao, Optimal configuration of concentrating solar power in multienergy power systems with an improved variational autoencoder, Appl. Energy 274 (2020) 115124.

[17] M. Brégère, R.J. Bessa, Simulating tariff impact in electrical energy consumption profiles with conditional variational autoencoders, IEEE Access 8 (2020) 131949–131966.

[18] D.P. Kingma, M. Welling, An introduction to variational autoencoders, Found. Trends Mach. Learn. 12 (4) (2019) 307–392.

[19] J. Xu, G. Durrett, Spherical latent spaces for stable variational autoencoders, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 4503–4513, [Online]. Available: https://aclanthology.org/D18-1480.

[20] W. Joo, W. Lee, S. Park, I.-C. Moon, Dirichlet variational autoencoder, Pattern Recognit. 107 (2020) 107514.

[21] C.P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β-VAE, 2018, arXiv preprint arXiv:1804.03599.

[22] J. Muehlenpfordt, Time series, in: Open Power System Data, 2019, Accessed on: Oct. 3, 2021. [Online]. Available: https://data.open-power-system-data.org/time_series/2019-06-05.

[23] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[24] C. Wang, Generating multivariate load states using a (conditional) variational autoencoder, 2022, Accessed on: Apr. 13, 2022. [Online]. Available: https://github.com/ChenguangWang-Sam/PSCC2022-CVAE.

[25] E. Sharifnia, Code release: System adequacy case study for CVAE load generation, PSCC 2022, 2022, Accessed on: Apr. 15, 2022. [Online]. Available: https://github.com/ensieh-sharifnia/MC-PSCC2022.

[26] F.J. Massey Jr., The Kolmogorov-Smirnov test for goodness of fit, J. Amer. Statist. Assoc. 46 (253) (1951) 68–78.

[27] C. Wang, S. Tindemans, K. Pan, P. Palensky, Detection of false data injection attacks using the autoencoder approach, in: 2020 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS), IEEE, 2020, pp. 1–6.

[28] G.J. Székely, M.L. Rizzo, Energy statistics: A class of statistics based on distances, J. Statist. Plann. Inference 143 (8) (2013) 1249–1272.

[29] J. Djolonga, A PyTorch library for differentiable two-sample tests, 2017, Accessed on: Oct. 3, 2021. [Online]. Available: https://github.com/josipd/torch-two-sample/blob/master/docs/index.rst.

[30] H. Liu, F. Han, M. Yuan, J. Lafferty, L. Wasserman, High-dimensional semi-parametric Gaussian copula graphical models, Ann. Statist. 40 (4) (2012) 2293–2326.

[31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 27 (2014).

[32] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 214–223.

[33] ENTSO-E, Mid-term adequacy forecast 2020, 2020, Accessed on: Oct. 1, 2021. [Online]. Available: https://www.entsoe.eu/outlooks/midterm/.

[34] M.P. Evans, S.H. Tindemans, Assessing energy storage requirements based on accepted risks, in: 2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe), IEEE, 2020, pp. 1109–1113.

[35] R.T. McGibbon, Quadprog 0.1.8 - PyPI, 2021, Accessed on: Aug. 15, 2021. [Online]. Available: https://pypi.org/project/quadprog.