Energy Informatics

RESEARCH Open Access

Boost short-term load forecasts with synthetic data from transferred latent space information

Benedikt Heidrich*, Lisa Mannsperger, Marian Turowski, Kaleb Phipps, Benjamin Schäfer, Ralf Mikut and Veit Hagenmeyer

From The 11th DACH+ Conference on Energy Informatics 2022
Freiburg, Germany. 15-16 September 2022

*Correspondence:
benedikt.heidrich@kit.edu

Karlsruhe Institute of Technology, Institute for Automation and Applied Informatics, Hermann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany
Full list of author information is available at the end of the article

Abstract

Sustainable energy systems are characterised by an increased integration of renewable energy sources, which magnifies the fluctuations in energy supply. Methods to to cope with these magnified fluctuations, such as load shifting, typically require accurate short-term load forecasts. Although numerous machine learning models have been developed to improve short-term load forecasting (STLF), these models often require large amounts of training data. Unfortunately, such data is usually not available, for example, due to new users or privacy concerns. Therefore, obtaining accurate short-term load forecasts with little data is a major challenge. The present paper thus proposes the latent space-based forecast enhancer (LSFE), a method which combines transfer learning and data augmentation to enhance STLF when training data is limited. The LSFE first trains a generative model on source data similar to the target data before using the latent space data representation of the target data to generate seed noise. Finally, we use this seed noise to generate synthetic data, which we combine with real data to enhance STLF. We evaluate the LSFE on real-world electricity data by examining the influence of its components, analysing its influence on obtained forecasts, and comparing its performance to benchmark models. We show that the Latent Space-based Forecast Enhancer is generally capable of improving the forecast accuracy and thus helps to successfully meet the challenge of limited available training data.

Keywords: Short-term load forecasting, Generative methods, Synthetic data

Introduction

The transition to a sustainable energy system includes the increasing integration of renewable energy sources into the power grid. Since these renewable energy sources depend on fluctuating weather conditions, their increased integration magnifies fluctuations in the energy supply that potentially destabilise the power grid (Kroposki et al.

2017). Methods for coping with these fluctuations and maintaining a stable power grid—e.g. load shifting—typically require accurate short-term load forecasts.

Therefore, in recent years, a large number of machine learning models for short-term load forecasting (STLF) and load forecasting in general have been developed (Upadhaya et al. 2019; González Ordiano et al. 2018). Although these models provide an improved forecasting accuracy, their increasing complexity is also associated with a growing need for training data (Hippert et al. 2001; Hastie et al. 2009), often initially multiple years, e.g., 1 year (Wu and Shahidehpour 2010), 2 years (Yona et al. 2008), or 3 years (Mabel and Fernandez 2008). Unfortunately, training data, e.g. for buildings, is often not available (Do and Cetin 2018), also known as cold-start problem (Moon et al. 2020). This lack of data may arise, for example, from manual data collection in non-advanced metering infrastructures (Fan et al. 2022), poor quality of the collected data (Fan et al. 2021), or newly constructed buildings (Fan et al. 2022; Ribeiro et al. 2018; Hooshmand and Sharma 2019). However, with limited data, machine learning-based STLF models cannot provide accurate load forecasts required for load shifting, which severely limits the potential of demand side flexibility and potentially affects grid stability.

Therefore, obtaining accurate forecasts from STLF models with limited available training data is a major challenge. The two most common approaches to meet this challenge are transfer learning (TL) and data augmentation (DA). TL applies previously acquired knowledge from one problem to solve another, similar problem. With regards to STLF, TL typically involves pre-training a machine learning model with similar existing data, before fine-tuning that model to the target data. In case of buildings, this pre-training could be performed on public data for similar buildings (Hooshmand and Sharma 2019), data from buildings showing a high correlation with the target building (Ozer et al. 2021; Gomez-Rosero et al. 2021; Tian et al. 2019; Lin and Wu 2021), or information-rich buildings (Li et al. 2021). Instead of using separate data sets for TL, data from multiple buildings can also be combined for the pre-training and individual buildings can be used for the fine-tuning (Voß et al. 2018). Furthermore, to counteract negative transfer and improve learning performance, specific source selection algorithms (Moon et al. 2020; Zhang and Luo 2015) or time series decomposition (Xu and Meng 2020) can be applied.

Whilst TL focuses on transferring information through the trained model, DA aims to increase the amount of available training data. Regarding STLF, one way to increase the amount of training data is to create slightly modified copies of existing data. Examples are applying noise to the existing data (Maalej and Rebai 2021) and simple data manipulation techniques such as rotation, permutation, jittering, and scaling (Fan et al. 2022). Another way is to create similar synthetic data with generative methods such as a conditional Variational Autoencoder (cVAE) (Fan et al. 2022) and a bidirectional generative adversarial network (BiGAN) (Zhou et al. 2020).

Although TL and DA provide promising results, both approaches have limitations. TL relies on the assumption that the information in the target data is similar to information in the source data and thus the information transfer is suitable (Goodfellow et al. 2016). At the same time, given a very limited amount of data, the performance of DA is severely restricted. DA can only create slightly modified additional data with noise and simple data manipulation techniques or, when using generative methods, no additional data at all due to an insufficient amount of available training data. DA is also only advantageous if the data used for augmentation is sufficiently diverse, for example, covering a complete year in case of yearly seasonalities. As a result, using TL and DA currently does not enable accurate forecasts from STLF models when little or no training data is available.

Therefore, the main contribution of the present paper is the latent space-based forecast enhancer (LSFE). The LSFE combines transfer learning and data augmentation to enhance STLF when training data is limited. The LSFE first trains a generative method conditioned with external information similar to the target data to capture relevant temporal patterns. In the present paper, we only include calendar information as an external feature. Using the resulting trained generative model, the LSFE then maps the available target data to a normally distributed latent space, before applying a seed noise sampling strategy to the latent space representation of the target data. Afterwards, the LSFE inputs the generated seed noise to the generative model to create synthetic target data. Lastly, the LSFE combines the synthetic and the previously existing real target data and uses this data to train the STLF model. By using a generative model to create additional training data, the LSFE is able to take advantage of source data similar to the target data whilst still considering the structure of the target data in the data generation process.

To evaluate the LSFE, we first examine the influence of its components, before analysing the influence of the LSFE on the obtained forecasts. We finally compare the performance of the LSFE to that of benchmark models. For the evaluation, we use real-world electricity data and apply our method with two state-of-the-art generative methods, three seed noise sampling strategies, and three data combination strategies.

The rest of the paper is structured as follows. We first present the LSFE in detail, including the possible generative methods, seed noise sampling strategies, and data combination strategies. We then introduce the experimental setting, comprising the used data, generative models, forecasting models, and metrics. The subsequent sections examine the influence of the LSFE's components, analyse the influence of the LSFE on obtained forecasts, and benchmark it to assess its performance. We finally discuss the LSFE and conclude the paper.

Latent space-based forecast enhancer

This section introduces the Latent Space-based Forecast Enhancer (LSFE) for enhancing STLF when available training data is limited. It comprises the three components generative method, seed noise sampling strategy, and data combination strategy. Before we introduce these components in detail, we shortly describe how the LSFE works using these components.

As shown in Fig. 1, the LSFE makes use of data from various sources to train a generative method. It then applies the available real data of the target to the trained generative method to obtain the representation of this data in the normally distributed latent space. Afterwards, the LSFE uses this latent space representation of the target data to determine suitable seed noise based on the selected seed noise sampling strategy. This sampling of seed noise is simplified by the LSFE, since sampling from a normal distribution is easier than from the unknown distribution in the original space. Given the generated seed noise, the LSFE passes it as an input to the generative model to create synthetic target data. Lastly, the LSFE combines the synthetic and the

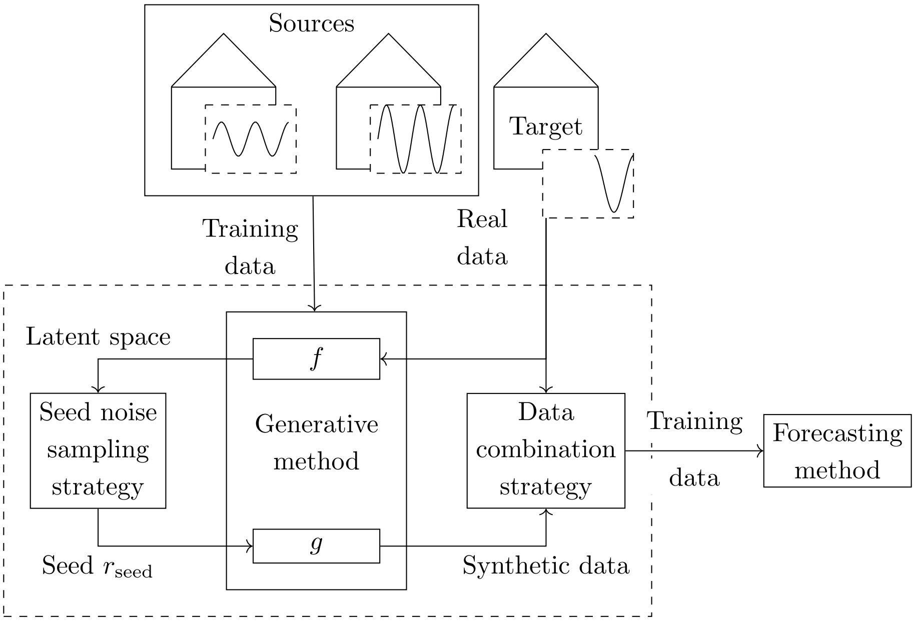


Fig. 1 The LSFE (dashed) consists of three components, i.e. the generative method with the mappings *f* and *g*, the seed noise sampling strategy, and the data combination strategy

previously existing real target data based on a selected data combination strategy. The resulting comparatively large data set serves as training data for a forecasting method.

Generative method

The first component of the LSFE is the generative method. The LSFE applies the generative method to generate synthetic data for the target, e.g., a building. Since the available real target data, such as electrical load data, can greatly vary between working days and weekends and may have strong seasonal profiles (Alrawi et al. 2019), it is highly dependent on calendar information. For this reason, the generative method has to consider calendar information when generating synthetic data for the target. Overall, the generative method performs two tasks within the LSFE.

The first task is to map the available real target data to the latent space so that the seed noise sampling strategy can use the latent space data representation of the available real target data. For this task, the generative model has to realise a mapping from the data space to the latent space, i.e.

$$f : \mathbb{X} \times \mathbb{C} \rightarrow \mathbb{Z},\ \mathbf{x}^i \mapsto f(\mathbf{x}^i; \mathbf{c}, \theta) = \mathbf{z}^i, \tag{1}$$

where $\mathbf{x}^i \in \mathbb{X}$ is a time series segment of fixed length, $\mathbf{c} \in \mathbb{C}$ is the considered calendar information, $\mathbf{z}^i \in \mathbb{Z}$ is the latent space representation of this time series segment, and θ are the trainable parameters.

The second task is to generate synthetic target data based on the seed noise provided. For this, the generative model has to realise a second mapping from the latent space to the data space, i.e.

$$g : \mathbb{Z} \times \mathbb{C} \rightarrow \mathbb{X}, \ \mathbf{z}^i \mapsto g(\mathbf{z}^i; \mathbf{c}, \theta) = \mathbf{x}^i, \tag{2}$$

where $\mathbf{x}^i \in \mathbb{X}$, $\mathbf{c} \in \mathbb{C}$, $\mathbf{z}^i \in \mathbb{Z}$, and θ are defined as above.

To realise both mappings *f* and *g* with consideration of calendar information, one can select cVAEs (Sohn et al. 2015) and conditional Invertible Neural Networks (cINNs) (Ardizzone et al. 2019) from existing generative methods because their architecture supports both mappings and conditional information by design. VAEs comprise a jointly trained encoder and decoder where the encoder realises the mapping *f* and the generator the mapping *g*. INNs are based on the bijective mapping $f^{-1} = g$ that realises both mappings. For VAEs and INNs, conditioning mechanism are available to consider calendar information during the data generation.

Seed noise sampling strategy

The second component of the LSFE is the seed noise sampling strategy. In the following, we present three possible seed noise sampling strategies including their underlying assumption and formal definition.

Random The random seed noise sampling strategy assumes that the latent space representation of the target data is normally distributed. Therefore, this strategy samples the seed noise as $r_{\text{seed}} = \epsilon$, where r_{seed} is the seed noise and $\epsilon \sim \mathcal{N}(0, 1)$ is normally distributed.

Around The around seed noise sampling strategy assumes that the latent space representation of additional target data is similar to the latent space representation of the available target data. In other words, the available target data already sufficiently points to the appropriate location in the latent space. Therefore, this strategy aims to sample the seed noise around the latent space representation of the available target data, i.e. $r_{\text{seed}} = f(\mathbf{x}^i; \mathbf{c}, \theta) + \epsilon$, where r_{seed} is the seed noise, $\epsilon \sim \mathcal{N}(0, \sigma)$ is normally distributed, and $f(\mathbf{x}^i; \mathbf{c}, \theta)$ is the latent space representation of the target data $\mathbf{x}^i$.

Shift The shift seed noise sampling strategy assumes that the latent space representation of the target data has a similar shape as the one of the source data. This strategy therefore aims to find a linear mapping from the latent space representation of the source data to the latent space representation of the target data. To find this mapping, we train a linear regression using source data with the same calendar information as independent variable and the latent space representation of the target data as dependent variable. Afterwards, we apply the linear regression on the source data to obtain the seed noise, i.e. $r_{\text{seed}} = l(f(\mathbf{x}^i; \mathbf{c}, \theta))$, where r_{seed} is the seed noise, l is the linear mapping from the source data to the target data, and $f(\mathbf{x}^i; \mathbf{c}, \theta)$ is the latent space representation of the source data $\mathbf{x}^i$.

Data combination strategy

The third component of the LSFE decides whether and how target data and sampled data are combined for training. In the following, we present three different data combination strategies and their underlying assumption.

Synthetic The synthetic data combination strategy uses only the generated synthetic target data to train the forecasting model.

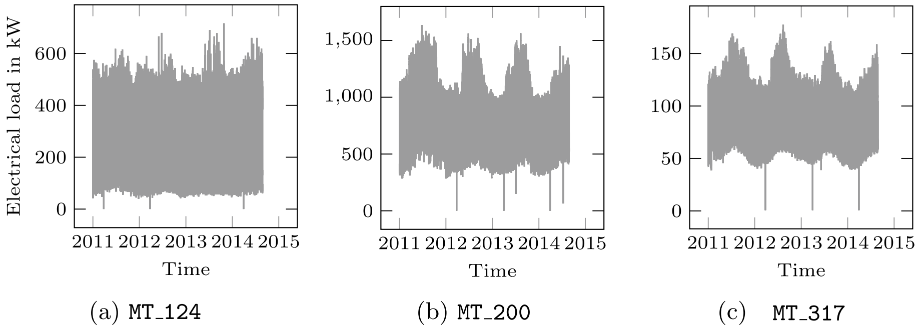


Fig. 2 The time series of the three clients from the UCI Electricity Load Dataset are used as target data in the evaluation

By only considering the synthetic target data, this strategy assumes that existing real target data do not provide any additional information and thus can be ignored.

Combined The combined data combination strategy uses the available target data and the generated synthetic target data. By considering both the existing real target data and the synthetic target data, this strategy assumes that—regardless of the amount—both target and synthetic data contain information that is relevant for the training of the forecasting method.

Fine-tune The fine-tune data combination strategy also considers both the existing real target data and the generated synthetic target data. It first trains the forecasting method on the synthetic target data, before fine-tuning the resulting forecasting model on the existing real target data. By using the existing real target data to fine-tune the forecasting model, this strategy assumes that it is beneficial to specialise the forecasting model on this real target data.

Experimental setting

This section describes how we evaluate the LSFE using pyWATTS (Heidrich et al. 2021). We first present the used data, before describing the used generative models and forecasting models. We then present the applied metrics.

Data

For the evaluation, we use the publicly available "ElectricityLoadDiagrams20112014 Data Set"[1] from the UCI Machine Learning Repository (Dua and Graff 2019). This data set contains real-world time series of 370 clients that have a quarter-hourly resolution and mostly cover the period from January 2011 to December 2014. This data set contains clients with different consumption behaviour such as factories and hotels (Rodrigues and Trindade 2018). To use the full period and to avoid negative impacts from concept drifts, we select three clients—namely MT_124, MT_200, and MT_317—for the evaluation (for the selected time series, see Fig. 2). We use the time series of each of these clients as target data and the time series of the other two selected clients as source data, resulting in three different combinations of source and target data.

[1] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

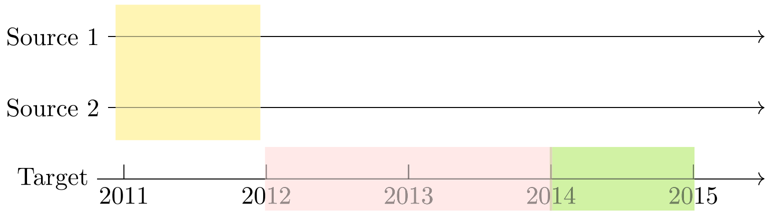


Fig. 3 As training data for the generative method, we choose the first year of source data (yellow). As test data for the forecasting method, we use the last year of the target data (green). As available real target data, we consider up to two years in 2012 and 2013 (red)

Table 1 The architecture of the subnetwork and the conditioning network in the used cINN

	Layers of the subnetwork
1	Dense 64, activation tanh
2	Dense horizon, activation linear
	Layers of the conditioning network
1	Dense 2, activation tanh
2	Dense 1, activation linear

As illustrated in Fig. 3, we select the first year of the respective source data as training data for the generative method. As test data for the forecasting method, we use the last year of the respective target data. From this target data, we additionally consider the second and third year as available real target data, which we use to determine the seed noise and to combine it with the synthetic target data to the training data for the forecasting method. More specifically, we examine the most recent 1, 2, 4, 8, 12, 26, 52, and 104 weeks of the available real target data in our evaluation.

Before using the data for the generative and forecasting methods, we perform the following four preprocessing steps. Firstly, we aggregate the data to an hourly resolution. Secondly, we standardise the data so that it has a mean of 0 and a variance of 1. Thirdly, we extract calendar information, which serve as conditional input for the generative method and as additional input for the forecasting method. Finally, we create overlapping samples of size 48 for the generative method and samples of size 24 for the forecasting method. These overlapping samples are due to our moving window approach, i.e., the first sample consists of the first 48 or 24 values, the second sample consists of the second value and the following 48 and 24 values, etc.

Used generative models

Since cINNs and cVAEs can be applied as the generative method in the LSFE as mentioned above, we evaluate the LSFE with an implementation of each generative method. First, we introduce the used cINN before we describe the used cVAE. Finally, we describe the input data, which is the same for both generative models.

cINN The used cINN consists of 10 coupling layers, each followed by a random permutation. We use GLOW coupling layers (Kingma and Dhariwal 2018) that implement a type of generative flow. Each of the GLOW coupling layers contains a subnetwork that enables the coupling layer to learn. As a subnetwork, we use a fully connected network

Table 2 The architecture of the encoder and decoder in the used cVAE

	Layers of the encoder
Input	[Load data, Calendar information]
1	Dense 32, activation: tanh
2	Dense 16, activation: tanh
3	μ: Dense latent dimension, activation: linear σ: Dense latent dimension, activation: linear
	Layers of the decoder
Input	[Seed noise, Calendar information]
1	Dense 16, activation: tanh
2	Dense 32, activation: tanh
3	Dense horizon, activation: linear

as shown in Table 1. To consider conditional information, we use a conditioning network as suggested by Ardizzone et al. (2019). As conditioning network that processes the conditional information, we also use a fully connected network as proposed in Heidrich et al. (2022) (see Table 1). We implement the cINN using FrEIA[2] and PyTorch (Paszke et al. 2019).

To train the cINN, we use the ADAM optimiser (Kingma and Ba 2015) and 50 epochs. In the training, we apply the maximum likelihood estimation as loss function to ensure that the latent space is normally distributed.

cVAE In the selected cVAE, we use fully connected networks for the encoder and the decoder. Table 2 shows the architecture of both. We implement the cVAE using Keras (Chollet et al. 2015).

To train the cVAE, we use the ADAM optimiser (Kingma and Ba 2015) and 500 epochs. The higher number of epochs compared to the training of the cINN is due to the different architectures. During the joint training of the encoder and the decoder, the decoder aims to reconstruct the input of the encoder by using the normal distributed latent space representation of the input provided by the encoder. To ensure a good reconstruction, we use the Root Mean Squared Error (RMSE) as reconstruction loss. Additionally, we use the Kullback Leibler Divergence to make sure that the latent space is normally distributed.

Input data All selected generative models receive time series segments of size 48 as input since the forecasting models should use the past 24 hours to forecast the next 24 hours. Additionally, the selected generative models consider calendar information about each entry of the time series segment as conditional information. More specifically, as calendar information, we use the sine and cosine encoded month of the year, the sine and cosine encoded hour of the day, a Boolean indicating whether it is a weekend, and a Boolean indicating whether it is a public holiday.

Used forecasting models

In the evaluation of the LSFE, we consider three different neural networks as forecasting models. From each popular architecture, we select one model, i.e. a convolutional

[2] https://github.com/VLL-HD/FrEIA

neural network (CNN), a fully connected neural network (FCN), and a long short-term memory (LSTM) network. In the following, we describe their architecture, training, and input data.

The CNN consists of four layers where the hidden layers are convolutional layers with a kernel size of 3, a ReLu activation function, and different filter sizes. The first hidden layer has a filter size of 5 and the second a size of 2. The output layer is a dense layer with a linear activation function. The FCN comprises three layers. The hidden layer has 32 units and a ReLu activation function whereas the output layer has a linear activation function. The LSTM network consists of three layers. The hidden layer is a LSTM layer with 32 units and the output layer is a dense layer with a linear activation function. To implement all three models, we use Keras (Chollet et al. 2015).

To train the forecasting models, we use the RMSprop optimiser (Hinton et al. 2012) and the Mean Absolute Error as loss function. We train each model with a maximum of 30 epochs and use a batch size of 64. To avoid overfitting, we apply early stopping with a patience of 5. As validation split, we use 0.2 with a random splitting, since it is the default value of Keras.

As data inputs, all three models receive time series segments of size 24 and the previously described calendar information of the values to be predicted. Each model aims to forecast the next 24 hours.

Metrics

In the evaluation, we use two metrics. One metric measures the accuracy and one metric measures the improvement in the accuracy. For the accuracy, we use the Root Mean Squared Error (RMSE). To measure the improvement in the accuracy, we use the RMSE skill score. The RMSE skill score is defined as $1 - RMSE_{\text{method}}/RMSE_{\text{base}}$, where $RMSE_{\text{method}}$ is the RMSE of the method to be evaluated and $RMSE_{\text{base}}$ the RMSE of a baseline. We consider a persistence model or a STLF model only trained on the available data as baseline. A positive skill score means that the forecast of the method to be evaluated is better than the baseline, while a negative skill score indicates that the baseline is better.

Influence of the LSFE's components

To evaluate the LSFE, we examine the influence of all its components. We first analyse the influence of the seed noise sampling strategy and then the influence of the data combination strategy. Since we always report the results for the cINN and the cVAE in these analyses, we hereby also investigate the influence of the used generative model.

Seed noise sampling strategy

We examine the influence of the seed noise sampling strategy qualitatively using a visualisation of the latent space and quantitatively calculating the accuracy.

Visualisation To qualitatively examine the influence of the different seed noise sampling strategies, we visualise the latent space representation of the target data, the seed noise, and the source data for the three seed noise sampling strategies and

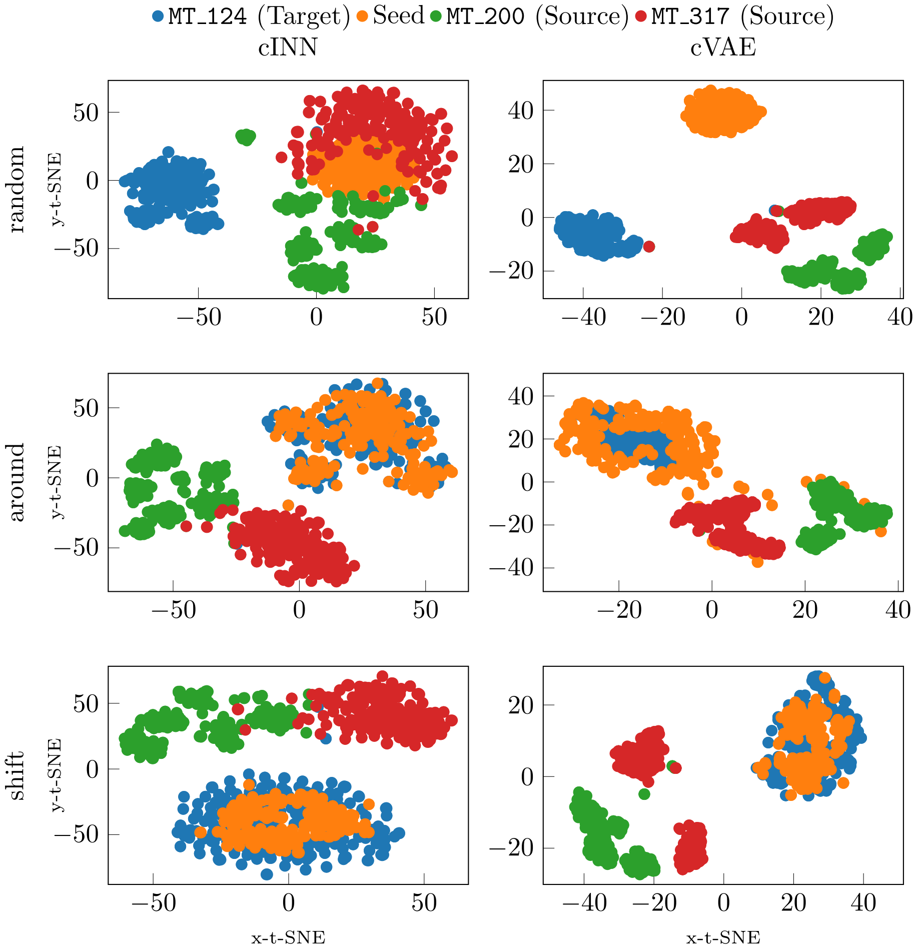


Fig. 4 The t-SNE visualisations of the latent space representation of 300 randomly selected samples from the target data (MT_124), the seed noise r_{seed}, and the source data (MT_200 and MT_317) for the three seed noise sampling strategies (random, around, and shift) and the two generative models (cINN and cVAE). Note that the different subfigures can only be compared qualitatively due to randomly selected start values and different input data

the two generative models. More specifically, we randomly select 300 samples from the target data (MT_124), the seed noise r_{seed}, and the source data (MT_200 and MT_317) for the visualisation but only consider samples with a start time at midnight for a better comprehensibility. To visualise this high dimensional data, we use the t-distributed stochastic neighbour embedding (t-SNE) (van der Maaten and Hinton 2008). The t-SNE maps the data points into a two-dimensional space such that similar data points appear close together and dissimilar data points far apart. We apply the t-SNE implementation from SKLearn (Pedregosa et al. 2011) with the random initialisation method, a perplexity of 20, 1000 iterations, and a learning rate of 600.

In the resulting t-SNE visualisations for the three seed noise sampling strategies and the two generative models in Fig. 4, we make the following three observations. First, for both generative models, we observe that the latent space representations of the target and source data do not overlap. Second, the normally distributed noise neither matches the target data in the latent space generated by the cINN nor that generated by the cVAE.

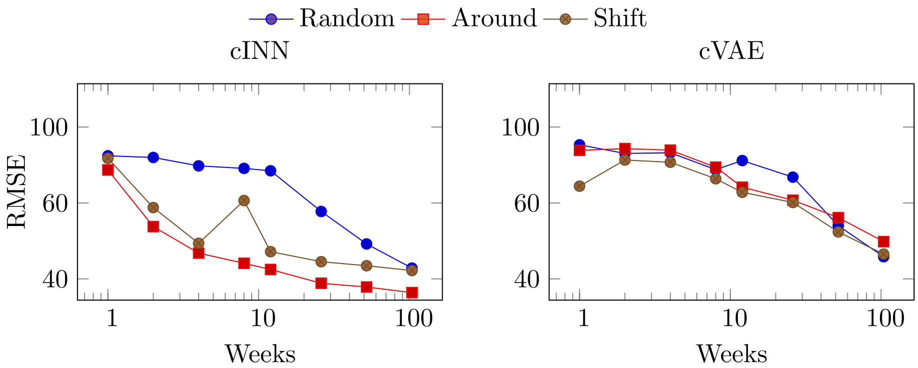


Fig. 5 The RMSE of the three seed noise sampling strategies and two generative models given different amounts of available target data (`MT_124`). The used forecasting model is the FCN and the LSFE is applied with the *combined* data combination strategy

Third, only the *around* and the *shift* seed noise sampling strategies lead to an overlap of the latent space representations of the target data and the seed noise. More specifically, the *shift* strategy leads to the best overlap for the cVAE and the *around* strategy for the cINN.

Accuracy To quantitatively evaluate the influence of the different seed noise sampling strategies, we train the FCN with the *combined* data combination strategy on different amounts of available target data and with different seed noise sampling strategies. Afterwards, we calculate the RMSE of the trained FCN on the test data.

Figure 5 shows the amount of available target data in weeks on the x-axis and the resulting RMSE on the y-axis. Based on these results, we make three observations. First, for both generative models, the LSFE achieves the worst results with the *random* seed noise sampling strategy. Second, the *shift* data sampling strategy has a peak in the RMSE at eight weeks of available target data for the cINN and at two weeks for the cVAE. Third, the best seed noise sampling strategy for the cINN is *around*, whereas *shift* is the best seed noise sampling strategy for the cVAE.

Data combination strategy

To examine the influence of the different strategies to combine real and synthetic data, we train the above FCN with the three data combination strategies on different amounts of available target data. As the seed noise sampling strategy, we use the best strategy from the previous experiment, i.e. the *around* seed noise sampling strategy for the cINN and the *shift* seed noise sampling strategy for the cVAE. After the training, we calculate the RMSE of the trained FCN on the test data.

Figure 6 shows the amount of available target data in weeks on the x-axis and the RMSE on the y-axis. In these results, we make two observations. First, for the cINN, all data combination strategies perform similarly. The RMSE decreases until 12 weeks of available target data and is then stable. Second, for the cVAE, the data combination strategies have varying results. The best data combination strategy is *fine-tune* followed by *combined* and *synthetic*.

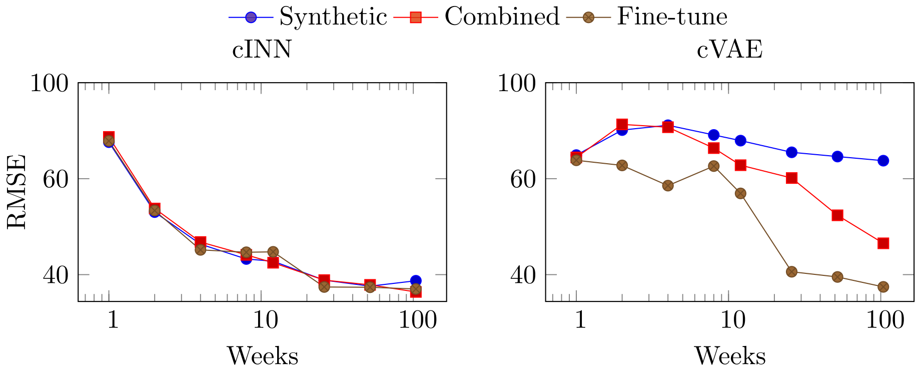


Fig. 6 The RMSE of the three data combination strategies and the two generative models given different amounts of available target data (`MT_124`). The used forecasting model is the FCN and the LSFE is applied with the *around* seed noise sampling strategy for the cINN and the *shift* seed noise sampling strategy for the cVAE

Influence on the forecast

To evaluate the LSFE, we also investigate its influence on the performed forecast. For this, we first examine how the LSFE improves the forecast accuracy. Second, we analyse whether the selected forecasting model influences this accuracy improvement. Third, we analyse whether the initialisation of the generative model influences the accuracy improvement.

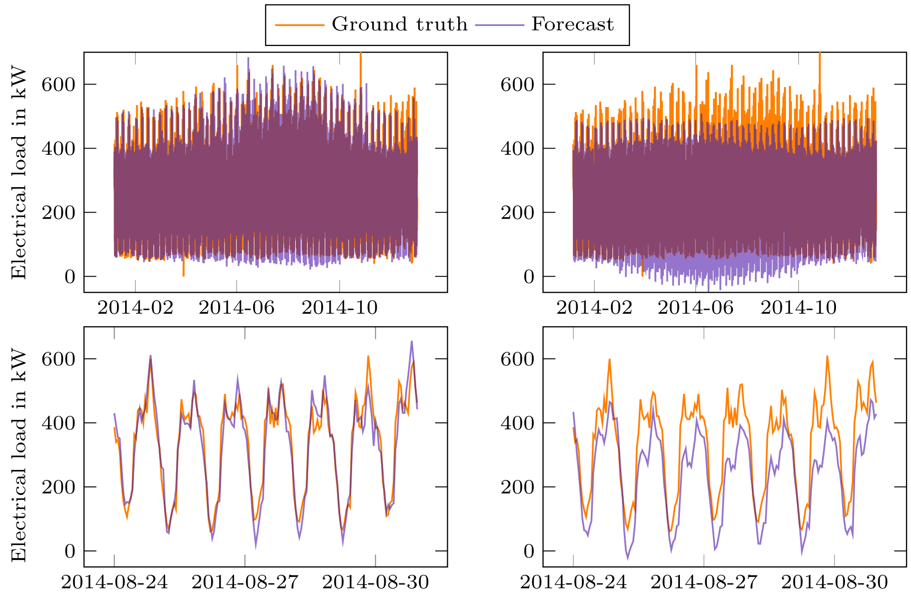
Accuracy improvement

To qualitatively examine how the LSFE improves the forecast accuracy, we compare a 24-hours forecast of the LSFE and a forecasting model only trained on twelve weeks of data. For the LSFE, we select the cINN as generative model, the *around* seed noise sampling strategy, the *combined* data combination strategy, and the FCN as the forecasting model.

Figure 7 shows the 24-hours forecasts of the LSFE and the FCN trained on the last twelve weeks of data from 2013 as well as the ground truth. The top row shows the forecasts for the complete period of the test data, the bottom row shows the forecast for a period of one week. We observe that the LSFE provides better forecasts than the forecasting model trained only on twelve weeks of data. In particular, the LSFE improves the forecasts for periods that differ from the data available in the training (e.g. July and August).

Influence of forecasting model

To examine whether the selected forecasting model influences the accuracy improvement by the LSFE, we apply different forecasting models with the LSFE and determine the improvement in their accuracy using the RMSE skill score with the forecast trained only on real data as $RMSE_{\text{base}}$. As forecasting models, we select the CNN, the FCN, and the LSTM network mentioned above. For the seed noise sampling strategy and the data combination strategy, we use the best strategies determined in the previous experiments, i.e. the *around* seed noise sampling strategy and the *combined* data combination



(a) LSFE

(b) Forecasting model trained on twelve weeks of data

Fig. 7 The 24-hours forecasts of the LSFE and the FCN trained on the last twelve weeks of data from 2013 as well as the ground truth for the complete period of test data (top row) and one week of the test data (bottom row). The considered client is `MT_124`

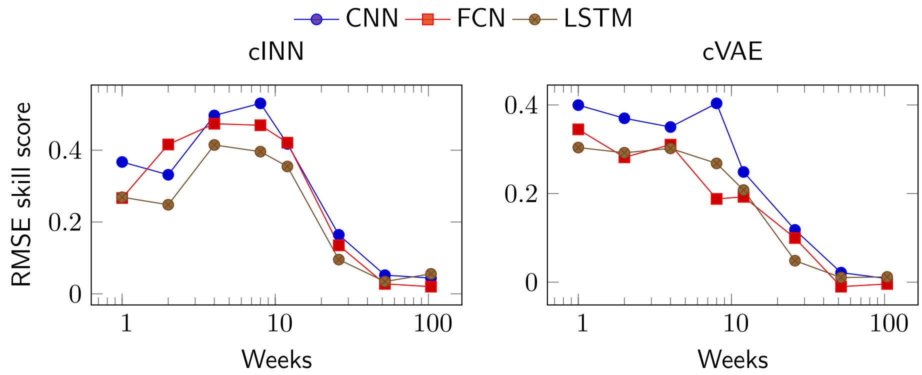


Fig. 8 The RMSE skill score of the three forecasting models applied with the LSFE given different amounts of available target data. The RMSE skill score considers the RMSE of the forecasting model trained using the LSFE as $RMSE_{\text{method}}$ and the RMSE of the forecasting model trained only on the available target data (`MT_124`) as $RMSE_{\text{base}}$. The LSFE is applied with the *around* seed noise sampling strategy and the *combined* data combination method for the cINN, and the *shift* seed noise sampling strategy and the *fine-tune* data combination method for the cVAE

strategy for the cINN as well as the *shift* seed noise sampling strategy and the *fine-tune* data combination method for the cVAE.

Figure 8 shows the RMSE skill score for the three forecasting models and different amounts of available target data. Based on these results, we make three observations. First, we observe considerable accuracy improvements with the LSFE for all three forecasting models when only a small amount of target data is available. Second, comparing

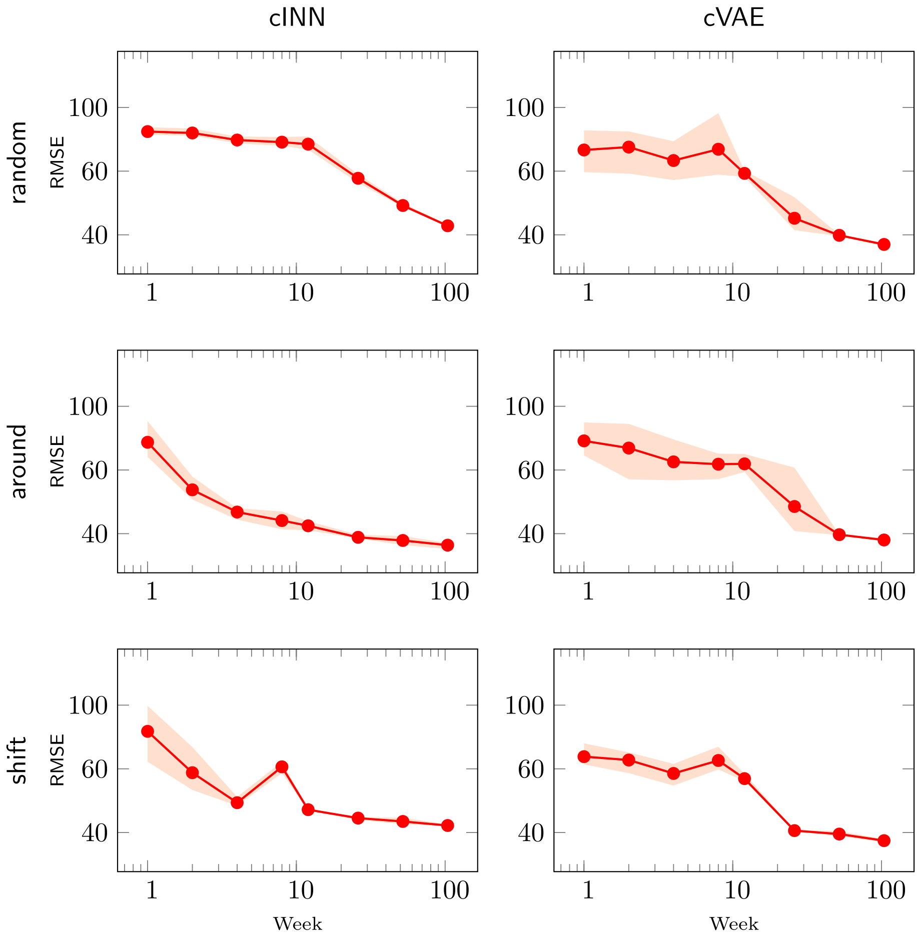


Fig. 9 The mean RMSE and the band comprising the minimum and maximum RMSE for the three seed noise sampling strategies (random, around, and shift), the two generative models (cINN and cVAE), and different amounts of available target data (`MT_124`). The used forecasting model is the FCN and the LSFE is applied with the *combined* data combination strategy for the cINN and the *fine-tune* data combination strategy for the cVAE

the cINN and the cVAE with the best strategies, the improvements in the RMSE of both generative models are similar. Third, the LSFE generally improves the forecasting accuracy regardless of the selected forecasting model but the extent of improvement varies. For example, the cINN achieves the strongest improvement with the FCN and the cVAE with the CNN.

Influence of initialisation

To examine whether the initialisation of the generative model influences the accuracy improvement by the LSFE, we apply the LSFE three times with the *combined* data combination strategy for the cINN and the *fine-tune* data combination strategy for the cVAE.

Figure 9 shows the mean RMSE and the band comprising the minimum and maximum RMSE for the three seed noise sampling strategies, the two generative models, and different amounts of available target data. In these results, we make two observations. First, the width of the band is smaller when using cINNs than cVAEs for each seed noise sampling strategy, except for the *shift* strategy. For this strategy, the bands have a similar

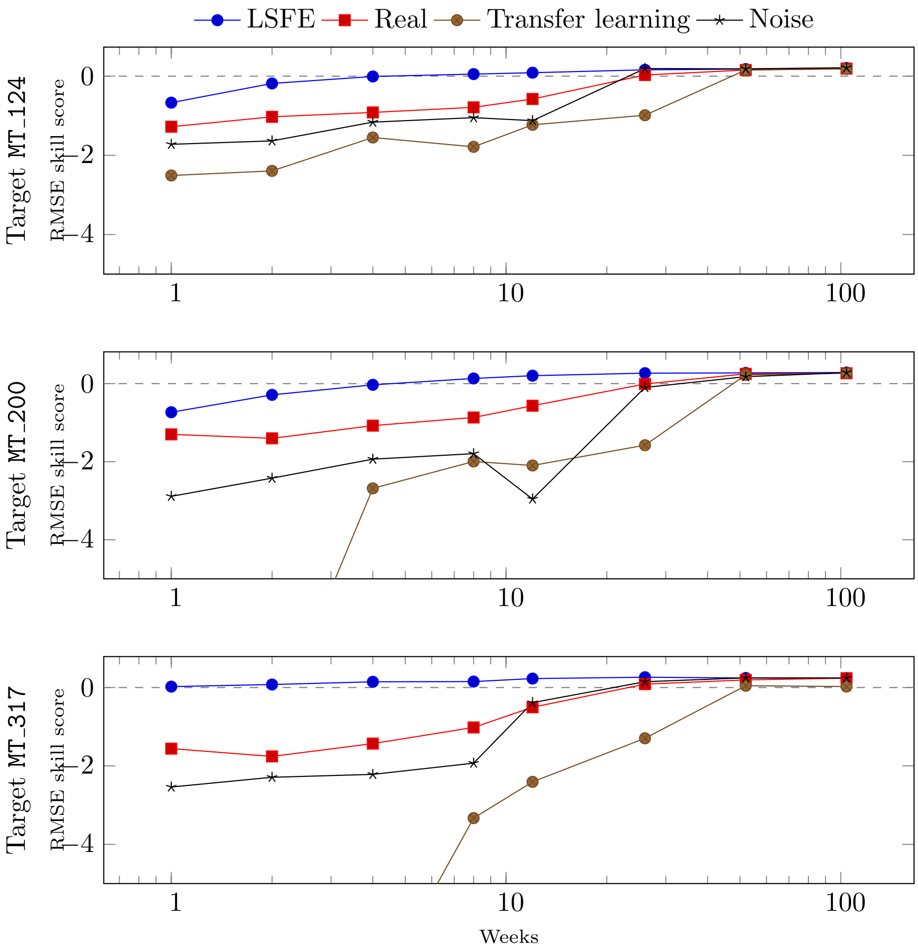


Fig. 10 The RMSE skill score with the persistence forecast as $RMSE_{base}$ of the LSFE using a cINN and the three benchmark models for different amounts of available target data and the three considered target data (`MT_124`, `MT_200`, and `MT_317`

width for the cINN and the cVAE. Second, with an increasing amount of available target data, the width of the band becomes smaller. However, for the *shift* data sampling strategy, there exists a peak at week eight. Note that the band width, which describes the difference between maximum and minimum RMSE, is always greater than zero.

Benchmarking

To assess the performance of the LSFE, we compare it with other methods in coping with small amounts of available data in STLF. For the comparison, we use the LSFE in the best configuration as determined in the previous experiments, i.e. with a cINN as generative model, the *around* seed noise sampling strategy, the *combined* data combination strategy, and the FCN as forecasting model.

For comparison, we select the following three common benchmark models. The first benchmark is a forecasting model that is trained only on the available real target data. The second benchmark realises the weight initialisation transfer learning approach. This approach first trains the forecasting model on the source data. Afterwards, the model is fine-tuned on the available target data. The last benchmark implements data augmentation using normally distributed noise. For this benchmark, we repeatedly draw noise from $\mathcal{N}(0, 0.1)$ and add it to the available target data until we also have 15000 training data samples.

Figure 10 shows the RMSE skill score of the LSFE and the three benchmark models for different amounts of available target data and the three considered target data. As $RMSE_{\text{base}}$, we use the persistence forecast. In this figure, we make three observations. First, the best performing model is the LSFE using the cINN. For all three selected target data, it outperforms all benchmark models when limited data is available. If not enough data is available all models perform similarly. Moreover, as indicated by the positive RMSE skill score, the LSFE is able to outperform the persistence forecast when at least eight weeks of data are available, much earlier than all benchmarks. When using less than eight weeks of data, the persistence forecast obtains better results for some targets. Second, the performance of the LSFE consistently improves with an increasing amount of available target data, whereas the transfer learning and the noise augmentation benchmark models show an abrupt improvement after a half and one year, respectively.

Discussion

This section discusses the influence of the LSFE's components, the influence of the LSFE on the performed forecast, the benchmark results, and the limitations and benefits of the LSFE.

Influence of the LSFE's components

Regarding the influence of the LSFE's components, we discuss three aspects in the following. First, we find that, depending on the used generative model, the *around* and the *shift* seed noise sampling strategies cover the latent space representation of the target data the best. In line with this finding, the cINN obtains the best quantitative results with the *around* and the cVAE with the *shift* seed noise sampling strategy. These observations indicate that the latent space data representation differs depending on the selected generative method. Therefore, for each generative method, a suitable seed noise sampling strategy has to be determined. Second, we find that, depending on the used generative model, the influence of the data combination strategy differs. While all data combination strategies lead to comparable results of the LSFE for the cINN, the selected data combination strategy influences the results for the cVAE. This finding indicates that the cINN generates synthetic data that are more similar to the real data of the target than the data generated by the cVAE. Third, we assume that the bijective mapping is the reason why the cINN performs better than the cVAE because it ensures that each pattern can be generated from the latent space.

Influence on the forecast

With regard to the influence of the LSFE on the forecast, we discuss four aspects in the following. First, we observe that the LSFE improves the forecasts, especially for periods that differ from the available data. We assume that there are two reasons for this improvement. Using conditional information enables the LSFE to provide data for periods that differ from available data. Additionally, the seed noise sampling strategy ensures that the generated synthetic data matches the target. Second, we find that the

LSFE improves the forecasts of different neural network-based forecasting models. This finding indicates that the LSFE solves the cold-start problem in STLF regardless of the selected forecasting model. Additionally, the good performance of the persistent forecasting model suggests that it could be applied in the first weeks until the LSFE obtains better results. Third, we observe that the results of the LSFE using the cINN with different initialisations are similar because the distances between the best and worst achieved RMSE are small. This observation indicates that the LSFE reliably improves STLF when limited training data is available, which is essential for smart grid applications. Fourth, when analysing the initialisations, we observe that the RMSE peaks slightly around week 8 for both models and multiple sampling strategies. This behaviour is unexpected, and further experiments should be conducted to determine whether it is due to the sampling strategy or an artefact in the data.

Benchmarking

Concerning the benchmarking results, we discuss two properties that we assume to be key contributors to the superior performance of the LSFE. First, the generative model applied in the LSFE learns structural information from multiple source data. Second, the conditioning mechanism enables the generative method to generate synthetic data with specific calendar information. Together with the seed noise sampling strategy, these two properties help the LSFE to generate synthetic data with varying calendar information that matches the target. Noise-based data augmentation strategies, however, only repeat the available target data. Therefore, the resulting synthetic data fits the target but does not have varying calendar information. Additionally, transfer learning strategies may provide training data with varying calendar information but this data often does not fit the target.

Limitations and benefits

Lastly, we discuss three potential limitations and three benefits of the LSFE. The first limitation is that the *around* seed noise sampling strategy requires additional parameters. More precisely, one has to specify σ to sample around existing data points. While the $\sigma = 0.1$ used in the evaluation leads to good results, an optimisation of this parameter could further improve the results. The second limitation is that the *shift* seed noise sampling strategy assumes a linear relationship between the latent space data distribution of the source and the target data. In general, however, this relationship is not necessarily linear. The third limitation is that we only examine a selection of possible seed noise sampling and data combination strategies that require at least some available target data. Therefore, future work should investigate these limitations.

A benefit of the LSFE is that it is not bound to a certain forecasting model. Rather, it can be used to improve the accuracy of any forecasting models. Lastly, the LSFE is also not bound to a specific aggregation level. Consequently, it can also be applied to load forecasting for new building areas and neighbourhood planning.

Conclusion

The present paper proposes a method combining transfer learning and data augmentation to enhance STLF when training data is limited. Using the latent space representation of source data and available target data from a generative method, the LSFE uses a seed noise sampling strategy and calendar information to create synthetic data that fits the target. The resulting synthetic data is then combined with the available real target data based on a data combination strategy and is finally used to train a forecasting model.

The evaluation of the LSFE's components shows that the LSFE using a cINN works best with the *around* seed noise sampling strategy and all of the considered data combination strategies, whereas the cVAE performs best with the *shift* seed noise sampling strategy and the *fine-tune* data combination strategy. The evaluation also shows that the LSFE generally improves the accuracy of short-term load forecasts, regardless of the used generative model, the applied forecasting model, and its initialisation. Moreover, the LSFE outperforms all considered benchmark models.

Future work could incorporate additional information, depending on the application area, such as the size of the building when forecasting individual buildings or the number of buildings in the case of neighbourhoods. Additionally, other generative methods could be tested such as a conditional BiGAN. Lastly, further seed noise sampling strategies and data combination strategies could be analysed.

Author contributions

BH Conceptualisation, Investigation, Methodology, Software, Validation, Visualisation, Writing—original draft. LM: Conceptualisation, Investigation, Software, Writing—original draft. MT: Conceptualisation, Writing—original draft, Writing—review and editing. KP: Conceptualisation, Writing—original draft, Writing—review and editing. BS: Conceptualisation, Writing—review and editing. RM: Funding acquisition, Supervision, Writing—review and editing. VH: Funding acquisition, Supervision, Writing—review and editing. All authors read and approved the final manuscript.

About this supplement

This article has been published as part of Energy Informatics Volume 5 Supplement 1, 2022: Proceedings of the 11th DACH+ Conference on Energy Informatics. The full contents of the supplement are available online at https://energyinformatics.springeropen.com/articles/supplements/volume-5-supplement-1.

Funding

This project is funded by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI, the Helmholtz Association under the Program "Energy System Design", the Helmholtz Association under grant no. VH-NG-1727, and the German Research Foundation (DFG) as part of the Research Training Group 2153 "Energy Status Data: Informatics Methods for its Collection, Analysis and Exploitation".

Availability of data and materials

The source code of the LSFE is publicly available GitHub at https://github.com/KIT-IAI/LatentForecastEnhancer. The data from the "ElectricityLoadDiagrams20112014 Data Set" used for the evaluation is openly available in the UCI Machine Learning Repository at https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014.

Declarations

Competing interests

The authors declare that they have no competing interests.

Published: 7 September 2022

References

Ardizzone L, Lüth C, Kruse J, Rother C, Köthe U (2019) Guided image generation with conditional invertible neural networks. arXiv:1907.02392

Alrawi O, Bayram IS, Al-Ghamdi SG, Koc M (2019) High-resolution household load profiling and evaluation of rooftop PV systems in selected houses in Qatar. Energies 12(20):3876

Chollet F et al. (2015) Keras . https://keras.io

Do H, Cetin KS (2018) Residential building energy consumption: a review of energy data availability, characteristics, and energy performance prediction methods. Curr Sustain/Renew Energy Rep 5(1):76–85

Dua D, Graff C (2019) UCI machine learning repository. http://archive.ics.uci.edu/ml

Fan C, Chen M, Wang X, Wang J, Huang B (2021) A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. Front Energy Res 9:652801

Fan C, Chen M, Tang R, Wang J (2022) A novel deep generative modeling-based data augmentation strategy for improving short-term building energy predictions. Build Simul 15(2):197–211

Gomez-Rosero S, Capretz MAM, Mir S (2021) Transfer learning by similarity centred architecture evolution for multiple residential load forecasting. Smart Cities 4(1):217–240

González Ordiano JÁ, Waczowicz S, Hagenmeyer V, Mikut R (2018) Energy forecasting tools and services. Wiley Interdiscipl Rev Data Mining Knowl Discov 8(2):1235

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge

Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning, 2nd edn. Springer, New York

Heidrich B, Bartschat A, Turowski M, Neumann O, Phipps K, Meisenbacher S, Schmieder K, Ludwig N, Mikut R, Hagenmeyer V (2021) pyWATTS: Python workflow automation tool for time series. arXiv:2106.10157

Heidrich B, Turowski M, Phipps K, Schmieder K, Süß W, Mikut R, Hagenmeyer V (2022) Controlling non-stationarity and periodicities in time series generation using conditional invertible neural networks. Appl Intell

Hinton G, Srivastava N, Swersky K (2012) Neural networks for machine learning lecture: lecture 6a overview of mini-batch gradient descent. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

Hippert HS, Pedreira CE, Souza RC (2001) Neural networks for short-term load forecasting: a review and evaluation. IEEE Trans Power Syst 16(1):44–55

Hooshmand A, Sharma R (2019) Energy predictive models with limited data using transfer learning. In: The Tenth ACM International Conference on Future Energy Systems (e-Energy 2019), pp. 12–16

Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations (ICLR 2015)

Kingma DP, Dhariwal P (2018) Glow: generative flow with invertible 1x1 convolutions. In: Bengio S, Wallach H, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R (eds) Advances in neural information processing systems, vol. 31, pp. 10215–10224

Kroposki B, Johnson B, Zhang Y, Gevorgian V, Denholm P, Hodge B-M, Hannegan B (2017) Achieving a 100% renewable grid: operating electric power systems with extremely high levels of variable renewable energy. IEEE Power Energy Mag 15(2):61–73

Li A, Xiao F, Fan C, Hu M (2021) Development of an ANN-based building energy model for information-poor buildings using transfer learning. Build Simul 14:89–101

Lin W, Wu D (2021) Residential electric load forecasting via attentive transfer of graph neural networks. In: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21), pp. 2716–2722

Maalej A, Rebai C (2021) Sensor data augmentation strategy for load forecasting in smart grid context. In: 2021 18th International Multi-Conference on Systems, Signals & Devices (SSD), pp. 979–983

Mabel MC, Fernandez E (2008) Analysis of wind power generation and prediction using ANN: a case study. Renew Energy 33(5):986–992

Moon J, Kim J, Kang P, Hwang E (2020) Solving the cold-start problem in short-term load forecasting using tree-based methods. Energies 13(4):886

Ozer I, Efe SB, Ozbay H (2021) A combined deep learning application for short term load forecasting. Alex Eng J 60(4):3807–3818

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. In: Wallach H, Larochelle H, Beygelzimer A, D'Alché-Buc F, Fox E, Garnett R (eds) Advances in neural information processing systems, vol 32, pp. 8024–8035

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830

Ribeiro M, Grolinger K, ElYamany HF, Higashino WA, Capretz MAM (2018) Transfer learning with seasonal and trend adjustment for cross-building energy forecasting. Energy Build 165:352–363

Rodrigues F, Trindade A (2018) Load forecasting through functional clustering and ensemble learning. Knowl Inf Syst 57(1):229–244

Sohn K, Yan X, Lee H (2015) Learning structured output representation using deep conditional generative models. In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R (eds) Advances in neural information processing systems, vol. 28, pp. 3483–3491

Tian Y, Sehovac L, Grolinger K (2019) Similarity-based chained transfer learning for energy forecasting with big data. IEEE Access 7:139895–139908

Upadhaya D, Thakur R, Singh NK (2019) A systematic review on the methods of short term load forecasting. In: 2019 2nd International Conference on Power Energy, Environment and Intelligent Control (PEEIC), pp. 6–11

van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(86):2579–2605

Voß M, Bender-Saebelkampf C, Albayrak S (2018) Residential short-term load forecasting using convolutional neural networks. In: 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)

Wu L, Shahidehpour M (2010) A hybrid model for day-ahead price forecasting. IEEE Trans Power Syst 25(3):1519–1530

Xu X, Meng Z (2020) A hybrid transfer learning model for short-term electric load forecasting. Electr Eng 102(3):1371–1381

Yona A, Senjyu T, Saber AY, Funabashi T, Sekine H, Kim CH (2008) Application of neural network to 24-hour-ahead generating power forecasting for pv system. In: 2008 IEEE Power and Energy Society General Meeting—Conversion and Delivery of Electrical Energy in the 21st Century

Zhang Y, Luo G (2015) Short term power load prediction with knowledge transfer. Inf Syst 53:161–169

Zhou D, Ma S, Hao J, Han D, Huang D, Yan S, Li T (2020) An electricity load forecasting model for Integrated Energy System based on BiGAN and transfer learning. Energy Rep 6:3446–3461

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.