Received: 14 July 2023 | Revised: 2 March 2024 | Accepted: 2 April 2024

IET Smart Grid

DOI: 10.1049/stg2.12169

ORIGINAL RESEARCH

Using conditional Invertible Neural Networks to perform mid-term peak load forecasting

Benedikt Heidrich | **Matthias Hertel** | **Oliver Neumann** | **Veit Hagenmeyer** | **Ralf Mikut**

Karlsruhe Institute of Technology, Institute for Automation and Applied Informatics, Eggenstein-Leopoldshafen, Germany

Correspondence

Benedikt Heidrich.
Email: benedikt.heidrich@kit.edu

Funding information

Helmholtz-Gemeinschaft, Grant/Award Numbers: Energy System Design, Helmholtz.AI

Abstract

Measures for balancing the electrical grid, such as peak shaving, require accurate peak forecasts for lower aggregation levels of electrical loads. Thus, the Big Data Energy Analytics Laboratory (BigDEAL) challenge—organised by the BigDEAL—focused on forecasting three different daily peak characteristics in low aggregated load time series. In particular, participants of the challenge were asked to provide long-term forecasts with horizons of up to 1 year in the qualification. The authors present the approach of the KIT-IAI team from the Institute for Automation and Applied Informatics at the Karlsruhe Institute of Technology. The approach to the challenge is based on a hybrid generative model. In particular, the authors use a conditional Invertible Neural Network (cINN). The cINN gets the forecast of a sliding mean as representative of the trend, different weather features, and calendar information as conditioning input. By this, the proposed hybrid method achieved second place overall and won two out of three tracks of the BigDEAL challenge.

KEYWORDS

artificial intelligence and data analytics, load forecasting, neural nets

1 | INTRODUCTION

Due to an increasing share of renewable energy sources in the electricity grid, the energy supply becomes more volatile, and consequently, balancing the grid and avoiding grid overload becomes more difficult. Thus, countermeasures such as peak shaving are needed to balance the grid. These countermeasures require accurate forecasts. Additionally, depending on the countermeasure, different characteristics of the forecast are of particular interest, such as accurate forecasts of the daily peak load, the peak timing and the peak shave. Thus, the BigDEAL challenge—organised by the Big Data Energy Analytics Laboratory (BigDEAL)—focused on forecasting these three characteristics.

Regarding peak load forecasting, various papers focus on forecasting the peaks directly. For example, there exist regression models with additional transformation to improve peak forecasting [1], Generalised Adaptive Models [2], Long Short-Term Memory networks (LSTMs) in combination with Convolutional Neural Networks [3], LSTMs with attention mechanisms [4], methods that cluster different load curves and apply specific regression on each group separately to handle the high variances between different load curves [5], a combination of multivariate empirical mode decomposition, Support Vector Regression (SVR), and particle swarm optimisation to provide accurate peak load forecasts [6], combining multi-resolution with forecasters (generalised additive models and neural networks) to predict the magnitude and the timing of peaks [7]. In contrast to the direct peak forecasting approach, extracting the peak characteristics from time series forecasts is also possible. Thus, we provide a short overview of existing energy time series forecasting methods in the following.

Benedikt Heidrich, Matthias Hertel and Oliver Neumann contributed equally.

Veit Hagenmeyer and Ralf Mikut contributed equally.

Regarding energy time series forecasting in general, in the Global Energy Forecasting Competitions (GEFCom) [8–10] leaderboards, we observe that often statistical or non-deep learning-based forecasting methods are used. In the load forecasting task of the GEFCom 2012, the top three teams used multiple linear regression [11], gradient boosting and Gaussian processes [12], and splines and ensembles of these models [13]. In the load forecasting task of GEFCom 2014, the top three teams used robust additive models [14], semi-parametric regression models [15], and a combination of multiple different forecasting models [16]. Finally, for GEFCom 2017, the top 3 teams used different quantile regression and generalised additive models [17], quantile gradient boosting regression trees, and an ensemble consisting of tree-based methods and neural networks [18].

Regarding neural networks for time-series forecasting, many recent papers propose new architectures. These architectures are often Recurrent Neural Networks (RNNs), which can capture temporal dependencies well and are often used for energy time-series forecasting [19–22]. However, RNNs are computationally expensive. Thus, transformers are proposed using the self-attention mechanism instead of recurrent layers [23]. Popular transformers for time series forecasting are the temporal fusion transformer [24], Informers [25], and Autoformers [26]. Also the first time series foundation model, which can be used for time series forecasting is based on transformers [27]. Furthermore, they are also applied for energy time-series forecasting [28–31]. Regarding neural network-based time series forecasting, hybrid approaches are especially promising. A popular hybrid network is DeepAR, which integrates autoregressive ideas into an RNN [32]. Furthermore, methods that exploit time series decomposition to improve forecasting are also promising. Examples of such methods are N-BEATS [33] and N-HiTS [34], which use an architecture that learns an additive decomposition of a time series, and Profile Neural Network (PNN) [35] and Probabilistic PNN [36], which use an additive decomposition into noise, trend, and periodicities, whereby the periodicities are modelled using rolling statistics. Finally, a hybrid neural network-based forecasting model is also the winner of the M4 [37] competition—the Exponential Smoothing RNN [38].

Thus, we also assume that hybrid neural networks may be beneficial for the BigDEAL challenge. However, due to the long forecast horizon (up to 1 year in the qualification), the historical data probably has no impact on most values. Nonetheless, historical values are an important input of the mentioned forecasting methods. Considering the long forecast horizon and the observation that the peak shape forecasting task rewards methods that create time series indistinguishable from real time series, we find that the BigDEAL challenge setting is similar to the time series generation task under exogenous variables.

In time series generation, popular approaches are often generative adversarial network [39] based, for example, the TimeGAN [40] and the COT-GAN [41]. However, according to ref. [42], these generation methods can not control the generation process that we require to create time series with the correct trend, seasonality, and dependence on the weather; thus, we use the conditional Invertible Neural Network (cINN) proposed in ref. [42] for time series forecasting.

The resulting contribution of the present paper is twofold. First, to the best of the authors' knowledge, we are the first that apply a cINN on peak forecasting. Second, we make this approach hybrid by conditioning the cINN with forecasted rolling averages to control the generation process and generate time series anticipating the trend and seasonal effects of the load time series. The results show that this method is promising. In particular, this method ranked second in the overall ranking of the BigDEAL challenge and won the timing and shape tasks of the challenge.

In the following, we introduce our method, including the cINN, our pre- and post-processing techniques, and our ensemble approach. Afterwards, we evaluate our models by introducing the experimental setup and describing the results. Finally, we discuss the results and conclude our work.

2 | cINN-BASED PEAK LOAD FORECASTING

To compete in the BigDEAL challenge, we used an ensemble whose main component is a cINN. As mentioned in the introduction, we selected a cINN due to the following three reasons:

1. In energy time series, values in the near future are correlated with historical values. However, the BigDEAL challenge requires forecasts with a forecasting horizon of up to 1 year. For such a forecasting horizon, we assume that historical values have almost no impact on most values. Furthermore, long forecasting horizons either require multi-step forecasting models with a large output dimension, leading to many parameters or autoregressive forecasting, where the error terms increase with an increasing forecasting horizon. To avoid both drawbacks, we applied a generative model that is non-autoregressive and has a fixed output length of only 24 values.
2. The shape task of the BigDEAL challenge is comparing the shape of the values around the peak without considering the magnitude. Thus, the shape of the forecasts should be indistinguishable from real data. This indistinguishability is a frequently asked requirement for generative models. Thus, we assume that generative models are well suited for the shape task.
3. The challenge organisers provided different exogenous features. Besides that, further exogenous information is easily extractable from the provided time series. Thus, the forecasting task in the BigDEAL challenge is comparable with time series generation for a specific scenario, which is —in our cases—defined by the exogenous features.

Because of these three reasons, we based our cINN-based forecaster on the cINN described in ref. [42], since it is designed to control the time series generation for specific

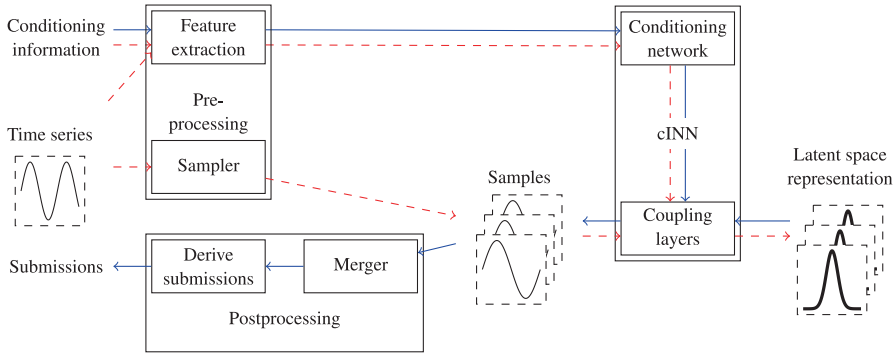

FIGURE 1 The conditional Invertible Neural Network (cINN) provides a bijective mapping between the time series and the latent space. To be able to model a conditional distribution, the cINN gets encoded conditional information via a conditioning network. Moreover, since the cINN works on samples with a fixed length, we need a sampler and use a merger for the reverse operation. The data flow during the training is indicated with the red dashed lines and the data flow during the generation with the blue solid lines.

scenarios. In the following, we explain the cINN before describing the other ensemble members.

2.1 | cINN based forecaster

The used cINN-based forecaster is visualised in Figure 1. It mainly consists of three steps—preprocessing, the cINN itself, and postprocessing. In the following, we explain each of these steps in more detail. Note that this cINN-based forecaster will be made available at sktime.[1]

2.1.1 | Preprocessing

The preprocessing consists of two parts—the feature extraction and the sampler.

Feature extraction

The feature extraction's task is to create and extract features for conditioning the generation process of the cINN. These extracted features are calendar information, statistical information, weather data and additional derived weather features. To perform the extraction, the feature extraction receives as input the time index, exogenous weather features, and during the training also the time series.

The calendar information is extracted from the time index. This information should support the generation process in creating time series with the calendar-driven periodicity structure. Furthermore, it enables us to aggregate the time series samples using the merger since this aggregation requires that each sample corresponds to a controlled and specific date (compare [42]). More specifically, as calendar information, we use the sine and cosine encoded month of the year, day of the week, and hour of the day. Furthermore, we use three flags indicating if the day is a workday, weekend, or a federal US holiday.

[1]Currently, there is an open Pull Request.

The challenge organisers provided temperature information from six weather stations as exogenous weather features. We use this temperature information to extract three kinds of additional weather features. First, we calculate the average, median, standard deviation, minimum and maximum as extra features based on six temperature values of the weather stations for each hour. Second, we transform the temperature time series. Therefore, we search the saddle point between the temperature time series and the target time series for each hour using a fitted polynom between both. Afterwards, we square the difference of the temperature time series and the saddle point for each hour. Third, we use a rolling mean of the average temperature and the linearised temperature time series to capture the heat inertia of the buildings that contribute to the time series.

Finally, we use statistical information to control mid- and long-term trends and periodicities. This information is represented and obtained by sine-based functions. In the present paper, we use two different functions, which are first fitted and then provide a forecast of the statistics. More specifically, we use the sliding mean as a statistic. Thus, during the training to fit this function, these steps require the sliding mean of the time series as target. During the inference, only the index is required to perform the prediction of the statistic. The first function is a simple sine function with three parameters

$$s_1 = \sin\left(\frac{x \cdot 4 \cdot \pi}{365 \cdot 24} + p_1\right) \cdot l_1 + o_2, \tag{1}$$

where x is the linear input ranging from 1 to the length of the training data, and p_1, l_1, and o_1 are the parameters that must be fitted. Thereby, p_1 is the phase of the searched periodicity, l_1 is the amplitude, and o_1 is the offset. We refer to this function in the following as the sine-1 statistic. This function can capture the half-yearly periodicity of the time series, which we observe in the data and is probably caused by electric heating in winter and electric cooling in summer. However, two additional yearly periodicities with opposite trends of their magnitude superimpose this basic half-yearly periodicity. To

capture this, we introduce the following function as a second variant:

$$\begin{aligned} s_2 \quad &= s_1 \\ &+ \left(\sin\left(\frac{x \cdot \pi \cdot 2}{365 \cdot 24} + 365 \cdot 24 \cdot \pi + p_2\right) + 1\right) \cdot a_2 \cdot x \\ &+ \left(\sin\left(\frac{x \cdot \pi \cdot 2}{365 \cdot 24} + p_2\right) - 1\right) \cdot a_3 \cdot x, \end{aligned} \tag{2}$$

where s_1 is described in Equation (1) and models the half-yearly periodicities, and the newly introduced sine terms with the parameters p_2, a_2, and a_3 model the two yearly periodicities. We refer to this function in the following as the sine-3 statistic.

Sampler

The cINN model requires that the input and output size of the samples is equal and fixed. However, the requested forecasts in the challenge have differing lengths. Thus, we create overlapping samples from the time series. Thereby, we define a sample starting at t_i as follows:

$$\mathbf{x}_{t_i} = \left(x_{t_i}, x_{t_{i+1}}, \ldots, x_{t_{i+h-1}}\right), \tag{3}$$

where x_{t_i} the time series' value at index t_i, and h is the length[2]. Note that to obtain a forecast with arbitrary length from these samples, the merger performs the reverse operation by aggregating the generated samples.

2.1.2 | cINN

The used architecture of the cINN is similar to the cINN presented in refs. [42, 43] and thus also implemented using the FrEIa framework[3]. Note that the described hyperparameters are selected by preliminary studies.

We use 15 GLOW coupling layers [44] with fully connected subnetworks. The subnetworks' specification is provided in Table 1. Similar to the previously used cINN, we also use a conditioning network to encode the conditioning information into a vector of size 64. The conditioning network's specification is provided in Table 2.

The cINN should provide a mapping between the time series samples and the normal distributed latent space. Thus, we applied the same loss function as described by the following [42, 43]:

$$\mathscr{L} = \mathbb{E}^i\left[\frac{\left\|f\left(\mathbf{x}^i; \mathbf{c}^i, \theta\right)\right\|_2^2}{2} - \log\left|J^i\right|\right] + \lambda\|\theta\|_2^2, \tag{4}$$

[2]We use $h = 24$

[3]https://github.com/vislearn/FrEIA

TABLE 1 The architecture of the used subnetworks and the conditioning network for the conditional Invertible Neural Network (cINN).

Layer	Unitss	Activation function
1	32	ReLU
2	n_{out}	Linear

TABLE 2 The architecture of the conditioning network for the conditional Invertible Neural Network (cINN).

Layer	Units	Activation
1	128	ReLu
2	64	Linear

where f is the cINN, J^i is the Jacobian corresponding to the ith sample [43], $\mathbf{x}^i$ is the ith time series samples, $\mathbf{c}^i$ the corresponding conditioning information and θ the parameters that should be learnt. The first part of the formula ensures that the cINN learns to map the input to a normal distribution using the change-of-variable formula, and the second part of the formula is a regularisation term. We trained the cINN for 50 epochs using Adam with a weight decay of $1e^{-5}$ and a learning rate of $5e^{-4}$ as an optimiser.

2.1.3 | Postprocessing

The postprocessing comprises two steps—namely the merging of the generated time series samples and the derivation of the submissions.

Merger

To create one time series, we need to merge the overlapping samples created by the cINN. Therefore, we first align the samples temporally. For example, the second entry of a sample generated with the calendar information corresponding to the time t_i is aligned with the first entry of a sample generated with calendar information corresponding to t_{i+1}. After this alignment, we calculate the median for each time step. More formally, we apply the following function:

$$\hat{x}_{t_k} = \text{median}\left(\hat{\mathbf{x}}_{t_i} j | i + j = k\right), \tag{5}$$

where $i + j = k$ corresponds to the alignment process, $\hat{x}_{t_k}$ is the time series values at time t_k and $\hat{\mathbf{x}}_{t_i} j$ is the jth entry of the sample generated with the calendar information that corresponds to t_j.

Derive submissions

Our proposed solution provides a time series forecast instead of directly predicting the required peak information. Thus, we derive this peak information from the time series. To derive the magnitude result, we extract the maximal value of each day. For the shape, we submit the complete forecast. For the timing task, we calculate the argmax for each day.

2.2 | Ensembling

To improve the forecast, we also use an ensemble. The ensemble members are the cINN-based forecasting model and the two models used as benchmarks: a SVR and a Fully Connected Neural Network that are described in more detail in Section 3.1.3. For each task, the ensemble obtains the prediction of each ensemble model. It optimises a linear regression on the training data to form the final forecast. Note that we restrict the linear regression to have only positive factors.

3 | EVALUATION

This section describes the experimental setup and the results.

3.1 | Experimental setup

The organisers of the BigDEAL challenge provide the dataset. It contains three time series with the hourly loads of three local distribution companies (LDCs 1–3) and the hourly ambient temperature values for six nearby weather stations. The ambient temperature values are measurements for the training period, and for the test period, they are forecasts.

The evaluation consists of six rounds with rolling training and test periods. For the first round, the training period is from January 2015 to December 2017, and the test period contains January and February 2018. The consecutive rounds test from March to May 2018 (round 2), June and July 2018 (round 3), August 2018 (round 4), September and October 2018 (round 5) and November and December 2018 (round 6). Note that the test period of each round was added to the training period of the following rounds.

3.1.1 | Metrics

For a given day i, we denote the 24 predicted hourly load values as a vector $\hat{y}_i = \left(\hat{y}_{i,1}, \ldots, \hat{y}_{i,24}\right)$ and the actual hourly load values as a vector $y_i = (y_{i,1}, \ldots, y_{i,24})$. In the following, we present metrics used by the organisers of the BigDEAL challenge to evaluate the performance of the forecasting method regarding the three tasks. Afterwards, we also introduce a further metric that we use for gaining insights into the proposed cINN-based forecaster.

Magnitude

The challenge uses the mean absolute percentage error (MAPE) to evaluate the magnitude predictions. Let n be the size of the test data, $m_i = \max(y_i)$ the ground truth peak magnitude and $\hat{m}_i$ the predicted peak magnitude for day i. The MAPE is defined as follows:

$$\mathrm{MAPE}(\hat{m}, m) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{m}_i - m_i}{m_i}\right| \tag{6}$$

Timing

A weighted and capped mean absolute error (wMAE) is used to evaluate the peak timing predictions. The weight for the error of a day is one if the forecast is off by 1 hour, two if the forecast is off by two to 4 hours, and the error is capped at 10 if the forecast is off by 5 hours or more. Let $p_i = \mathrm{argmax}(y_i)$ be the ground truth peak magnitude and $\hat{p}_i$ the predicted peak magnitude for day i. The wMAE is defined as follows:

$$\mathrm{wMAE} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{weighted}\left(\left|\hat{p}_i - p_i\right|\right) \tag{7}$$

with

$$\mathrm{weighted}(x) = \begin{cases} x & \text{for } x \in \{0, 1\} \\ 2x & \text{for } x \in \{2, 3, 4\} \\ 10 & \text{for } x \in \{5, 6, \ldots, 23\} \end{cases} \tag{8}$$

Note that the values of $x \in \{0, 1, \ldots, 23\}$ are discrete.

Shape

The calculation of the shape metric consists of three steps. First, the predicted loads and ground truth loads are normalised by the daily peaks

$$\overline{y}_i = y_i / \max\left(y_i\right) \tag{9}$$

$$\overline{\hat{y}}_i = \hat{y}_i / \max\left(\hat{y}_i\right). \tag{10}$$

Second, the sum of the absolute errors in the period ranging from 2 hours before the actual peak to 2 hours after the actual peak is computed for every day. Third, the daily sums of errors are averaged across all days to obtain the shape metric:

$$\mathrm{shape}\left(\overline{\hat{y}}, \overline{y}\right) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j=p_i-2}^{p_i+2}\left|\overline{\hat{y}}_{i,j} - \overline{y}_{i,j}\right| \tag{11}$$

where $p_i = \mathrm{argmax}(y_i)$ is the peak timing of day i.

Mean Percentage Error

To gain insights and analyse our method's bias, we use the Mean Percentage Error (MPE) as an additional metric. The MPE measures the average percentage deviation of the forecast from the ground truth. Thereby, values greater than zero mean that the forecast overestimates the ground truth, and values below zero mean that the forecast underestimates the ground truth. Furthermore, we can interpret the distance to zero as the severity of the bias. The MPE is defined as follows:

$$\mathrm{MPE}(\hat{y}, y) = \frac{100}{n} \sum_{i=1}^{n} \frac{\hat{y}_i - y_i}{y_i}. \quad (12)$$

3.1.2 | Considered cINN implementations

In the following evaluation, we use four different variants of the cINN-based forecaster to which we refer as cINN variants. In particular, we use a cINN with the sine-1 statistics (cINN-1) and with the sine-3 statistic (cINN-2). Furthermore, we use also both variants with and without applying transfer learning (TL) to both variants. To transfer learn the cINN variants, we pre-train them on all time series before fine-tuning them on each time series separately.

3.1.3 | Benchmarks

As benchmarks, we use a SVR and a fully connected neural network. As SVR, we use the implementation provided by SKLearn [45] and the default hyperparameters. To implement the fully connected neural network, we use PyTorch [46]. The network comprises three hidden layers with 64, 32, and 16 neurons and uses ReLU as an activation function. Both benchmarks take the same exogenous information as the cINN as input and predict each value of the time series separately. Thus, similar as for the cINN, from the predicted time series, the peak position and timing has to be derived.

3.2 | Results

The present results focus on assessing the performance of the four different versions of the cINN, gaining insights into how the cINN works, comparing the cINNs performance with the benchmarks, and presenting our ranks in the leaderboard of the BigDEAL challenge.

3.2.1 | cINN variants

To assess and compare the different cINN variants, we use two evaluations. First, we evaluate how the performance differs across the six rounds. Finally, we compare their overall performances.

Performance through the rounds

Figure 2 visualises the cINN variants' performance over the six rounds using lineplots for each task and time series. In the following, we discuss the observation for each task separately.

Regarding the shape, we make two observations: First, the forecast quality of all models fluctuates strongly during the six rounds. For each time series and model, the best round achieves a score that is half of the score of the worst round.

Second, the observed fluctuations have seasonal patterns, at least for LDC1 and LDC3, where the performance during the summer months is better. For LDC2, we observe that the performance is constant except for the last round.

Regarding the daily peaks' magnitude, we make two observations again: First, similar to the first task, the forecast performance for the daily peaks' magnitude strongly fluctuates across the six rounds. In contrast to the first task, we cannot observe a clear seasonal pattern.

Second, for LDC1 and LDC3, the fifth round is worse than the other rounds. For LDC2, the fifth round behaves similarly to the other rounds.

Regarding the timing task, we make one observation: Like in the previous tasks, the performance fluctuates strongly through the six rounds. The rounds in the summer are better than the winter months.

Average performance

Table 3 provides each cINN variant's average scores and ranks for each time series. In the following, we report our observations for each of the three tasks:

Regarding the shape task, first, methods with TL achieve the best average rank and score. However, the differences are minor (the maximum percentage difference is 7% and achieved on LDC3). Second, when comparing the performance of the three different time series, the best performance is achieved on LDC3 and the worst on LDC2.

Regarding the magnitude, first, the best model is the cINN-2 with TL. This model is the best for LDC2 and LDC3 but struggles for LDC1. In contrast to the magnitude task, the differences between the best and worst models are higher (The maximal deviation is 20%). Second, when comparing the performance of the different time series, the best scores are achieved for LDC1. The performances on LDC2 and LDC3 are comparable.

Regarding the timing task, first, the methods that use the sine-3 statistic provide better forecasts of the daily peaks' timing. Second, we also observe that the overall performance of the four cINN variants fluctuates strongly, with 15% for the time series LDC3.

Finally, in our last observation for the daily peaks' timing forecast, we observe that the peak timings for LDC3 are the easiest to forecast, followed by LDC2 and LDC1.

Wrapping up from the overall performance of the three tasks, the performance of all cINN variants is similar. However, it also appears that the cINN-2 with TL is slightly better than the other models. Thus, we use this model in the benchmark section.

3.2.2 | Insights

In this subsection, we gain insights into the proposed method's performance by analysing its bias using the MPE and visualising the results.

Mean Percentage Error

We use the MPE to quantify the bias of a forecast. An MPE greater than zero means that the forecast generally

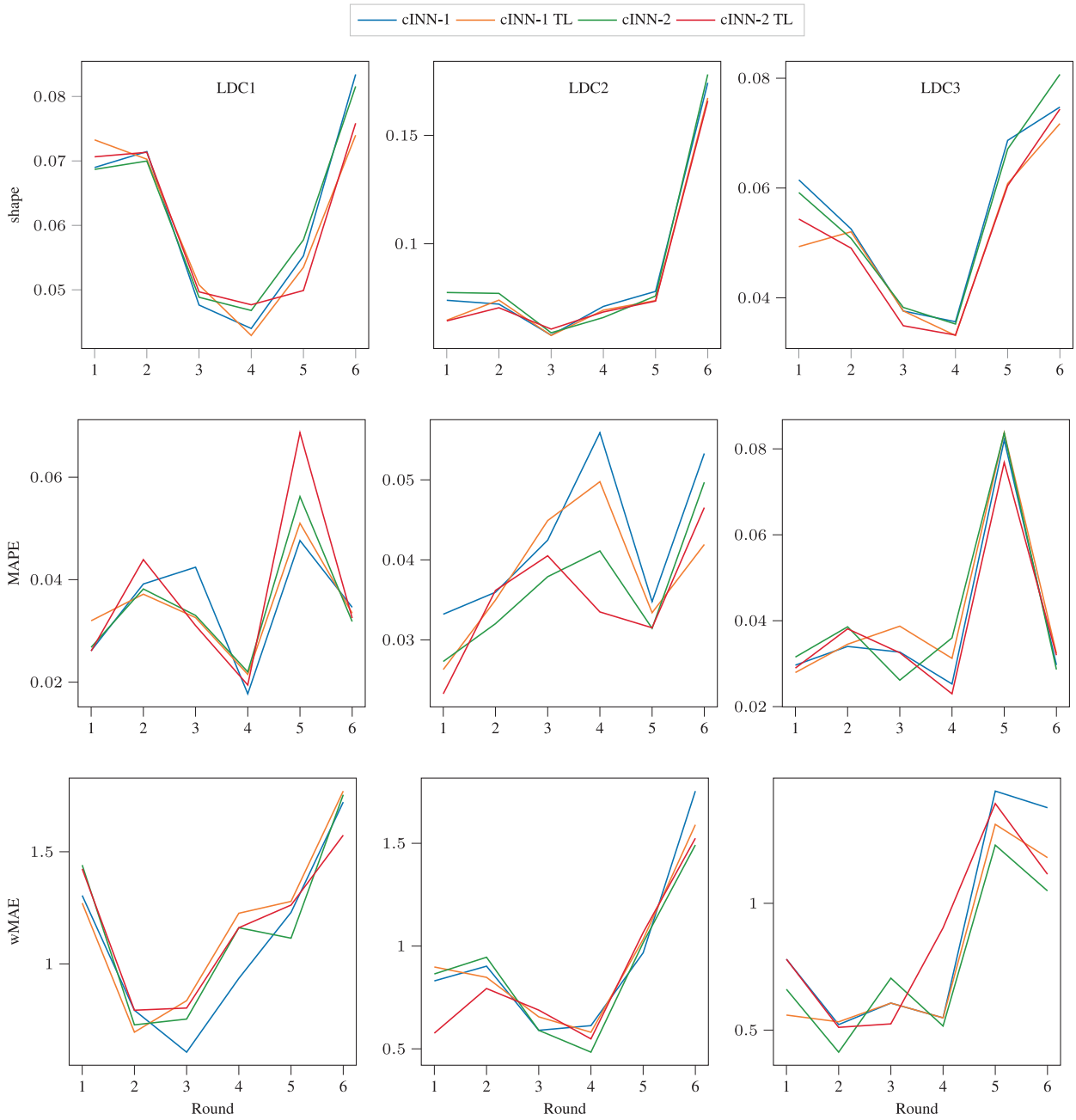

FIGURE 2 For each task and time series, we provide lineplots over the six different rounds that indicate the performance of the four conditional Invertible Neural Network (cINN) variants. (a) The performance of the cINN variants on daily peaks' shape. (b) The performance of the cINN variants on daily peaks' magnitude. (c) The performance of the cINN variants on daily peaks' timing.

underestimates, and an MPE smaller than zero means that the forecast generally overestimates. For the three tasks, Table 4 provides the MPEs.

Regarding the shape, we make three observations: First, the bias of our method is the highest for LDC2, followed by LDC1 and LDC3. Whereby in total, our method underestimates the consumption of LDC2 and LDC3 and overestimates the consumption of LDC1. Second, the bias is comparably high for LDC1 and LDC3 in round 5. In this round, all cINN variants overestimate the consumption. Third, for all three time series and all cINN variants, the bias varies strongly through the different rounds. However, we can not recognise a seasonal correlation of the bias.

Regarding the magnitude task, we make three observations: First, the forecasts for LDC2 have the most substantial bias. Furthermore, similarly to the shape task, the forecasts for LDC2 and LDC3 underestimate the actual

TABLE 3 The average score and rank for all rounds for each conditional Invertible Neural Network (cINN) variant and task.

			cINN-1		cINN-2	
			No TL	**TL**	**No TL**	**TL**
Shape	LDC1	Score	0.062	0.061	0.062	0.061
		Rank	3	1	4	2
	LDC2	Score	0.088	0.084	0.089	0.084
		Rank	3	2	4	1
	LDC3	Score	0.055	0.051	0.055	0.051
		Rank	3	1	4	2
	Average	Score	0.068	0.065	0.069	0.065
		Rank	3	1.33	4	1.67
Magnitude	LDC1	Score	3.462	3.462	3.468	3.694
		Rank	1	2	3	4
	LDC2	Score	4.260	3.855	3.658	3.525
		Rank	4	3	2	1
	LDC3	Score	3.890	4.143	4.076	3.857
		Rank	2	4	3	1
	Average	Score	3.871	3.82	3.734	3.692
		Rank	2.33	3	2.67	2
Timing	LDC1	Score	1.099	1.180	1.159	1.170
		Rank	1	4	2	3
	LDC2	Score	0.943	0.934	0.899	0.866
		Rank	4	3	2	1
	LDC3	Score	0.879	0.790	0.762	0.871
		Rank	4	2	1	3
	Average	Score	0.974	0.968	0.94	0.969
		Rank	3	3	1.67	2.33

values and for LDC1 overestimate them. Second, we see strong biases for LDC1 and LDC3 for round 5. Third, in general, the bias varies between the different rounds. While for LDC2, we cannot recognise a seasonal pattern, we can for LDC1 and LDC3. It seems that for LDC1 and LDC3, the bias during the winter is lower than in the summer.

Regarding the timing task, we make two observations: First, our method estimates the timing of the daily peaks too late for all three time series and all cINN variants. The overestimation is the strongest for LDC3, followed by LDC1 and LDC2. Second, there are substantial fluctuations of bias through the six rounds. For example, for the cINN-1 on LDC3, the MPE fluctuates between 3 and -32. We generally observe that the bias fluctuation is the highest for LDC3, followed by LDC1 and LDC2.

Visualisation

Figure 3 provides a visualisation of the forecasts for the overall predicted time series[4], for the magnitude and the timing. In this visualisation, we make two observations: First, we observe that the prediction is almost narrow to the ground truth (black line) regardless of the task, except for a dip in September for LDC3[5] (last accessed 07.06.2023). Second, we observe that for LDC2 and LDC3, the overall prediction and the magnitude appear almost always underestimated. For the timing task, we do not observe something similar.

3.2.3 | Benchmarking

For benchmarking, we compare the results of the SVR model, the neural network, the cINN, and the ensemble for the three tasks and the three LDCs (Table 5). Thereby, we make two observations:

First, we observe that, in total, the cINN is the best single model for all tasks and LDCs. The ensemble only improves the magnitude results for LDC2 and LDC3. For all other tasks, the cINN is better than the ensemble. Note that sometimes the benchmarks perform better than the cINN for specific rounds. Second, when examining the distance between the cINN's performance and the performance of the benchmarks, we observe that the distances are smaller in the summer than in the winter.

3.2.4 | Computational effort

With regard to the training times (Table 6), we make three observations:

First, the proposed cINN-based method requires more training time than both benchmarks. Second, when analysing the training times of the different cINNs, we observe that the variants with TL need more training time since they use a bigger dataset, and the training time includes the sum of training on all data and the fine-tuning. Comparing cINN-1 and cINN-2, we can not see any differences. Finally, regarding the different rounds, the training time increases, since, with each new round, more training data is available.

3.2.5 | Big Data Energy Analytics Laboratory challenge

Regarding the challenge's final leaderboard, Table 7 shows the best five teams for each task[6]. We want to highlight that,

[4]Note, we use this instead of the shape visualisation since the shape is scaled for each day.
[5]We assume that this dip is caused by Hurricane Florence: https://www.washingtonpost.com/weather/2018/09/20/land-transformed-by-water-north-carolina-before-after-hurricane-florence/
[6]http://blog.drhongtao.com/2022/12/BigDEAL-challenge-2022-final-leaderboard.html

TABLE 4 Mean Percentage Error (MPE) of the conditional Invertible Neural Network (cINN) variants.

			LDC1				LDC2				LDC3			
	Round		cINN-1		cINN-2		cINN-1		cINN-2		cINN-1		cINN-2	
	Round	Time span	No TL	TL	No TL	TL	No TL	TL	No TL	TL	No TL	TL	No TL	TL
Shape	1	01–02.2018	0.46	−0.66	1.30	−0.35	2.02	1.05	−2.20	−0.46	2.11	1.53	0.57	1.23
	2	03–05.2018	−0.76	−0.26	−0.36	0.97	1.14	1.71	−0.26	1.64	2.23	2.65	3.33	3.30
	3	06–07.2018	−4.69	−0.78	−2.20	−1.26	1.60	2.43	1.25	2.84	2.44	3.31	1.76	2.36
	4	08.2018	−0.02	0.47	0.46	−0.08	3.25	3.69	2.04	2.04	2.05	2.96	3.15	1.91
	5	09–10.2018	−4.89	−4.23	−6.77	−7.13	0.51	1.18	0.43	1.54	−6.52	−6.39	−7.30	−7.65
	6	11–12.2018	−1.44	−0.33	0.83	0.61	5.49	3.19	5.03	3.95	0.35	1.77	0.26	2.20
	Total	01–12.2018	−1.89	−0.97	−1.12	−1.21	2.33	2.21	1.05	1.92	0.44	0.97	0.30	0.56
Magnitude	1	01–02.2018	0.39	−0.97	1.48	−0.61	2.08	1.19	−1.14	0.09	1.81	1.27	0.88	0.58
	2	03–05.2018	−0.33	0.20	0.15	1.42	1.40	1.61	−0.03	2.06	2.32	2.57	3.46	3.24
	3	06–07.2018	−3.76	−0.06	−1.23	−0.68	3.27	3.57	2.60	3.09	3.01	3.68	1.99	2.79
	4	08.2018	0.62	1.55	1.27	0.68	5.56	4.93	3.84	3.00	2.13	2.99	3.43	1.82
	5	09–10.2018	−3.25	−3.27	−4.83	−6.12	1.61	1.86	1.29	1.85	−3.08	−2.28	−3.00	−4.10
	6	11–12.2018	−1.39	−0.36	1.02	0.51	4.34	1.81	3.88	3.02	0.71	1.42	−0.15	1.76
	Total	01–12.2018	−1.28	−0.48	−0.35	−0.80	3.04	2.49	1.74	2.19	1.15	1.61	1.10	1.02
Timing	1	01–02.2018	−2.96	−3.02	−1.21	−0.99	3.89	0.39	3.08	2.12	−10.34	3.20	3.78	−9.91
	2	03–05.2018	−3.55	−3.52	−3.57	−3.13	−5.66	−4.28	−3.39	−6.68	−0.70	−3.08	−1.60	−3.14
	3	06–07.2018	−0.73	0.14	−0.48	−0.30	−0.56	−0.58	−0.74	−0.42	−1.17	−1.02	−1.19	−1.09
	4	08.2018	−1.72	−1.58	−2.68	−2.66	0.36	0.21	−0.68	0.02	−1.26	−1.26	−1.07	−2.34
	5	09–10.2018	−5.11	−4.41	−3.09	−4.65	−5.04	−7.01	−5.19	−5.43	−32.54	−32.23	−32.60	−32.27
	6	11–12.2018	−11.99	−8.08	−19.95	−4.94	−6.75	−9.04	−3.00	−7.35	−18.57	−21.15	−18.44	−21.01
	Total	01–12.2018	−4.35	−3.41	−5.16	−2.78	−2.29	−3.38	−1.65	−2.96	−10.77	−9.26	−8.52	−11.63

overall, our method achieves second place. Furthermore, our method wins the most tasks (the shape and the timing task). Regarding the magnitude, our method achieves fifth place from 13 competitors in the final.

4 | DISCUSSION

The results show that cINN as a generative method can be successfully applied to mid-term peak load forecasting. In the following, we discuss the results of the BigDEAL challenge, the insights, the computational effort, and the further potential of cINNs for time series forecasting.

4.1 | Challenge related

The results of the BigDEAL challenge show that cINNs are suited for mid-term daily peak timing and shape forecasting. However, they struggle to forecast the magnitude and can have a bias. Thus, future work can solve these issues using an advanced merger. For example, the merger could try to learn the aggregation function instead of just using the median.

4.2 | Insight related

Regarding the insights of the cINNs variants, we discuss two aspects: First, the cINN's performance fluctuates strongly across the different rounds. Possibly, this is explainable by the different consumption patterns during different seasons. Thus, future work should try to quantify these fluctuations, for example, by using prediction intervals. Second, we also observe that the results on the different time series differ strongly. A possible explanation could be stochastic smearing effects resulting from the differing numbers of consumers or producers that belong to the time series. Unfortunately, we do not have the information to evaluate this in more detail.

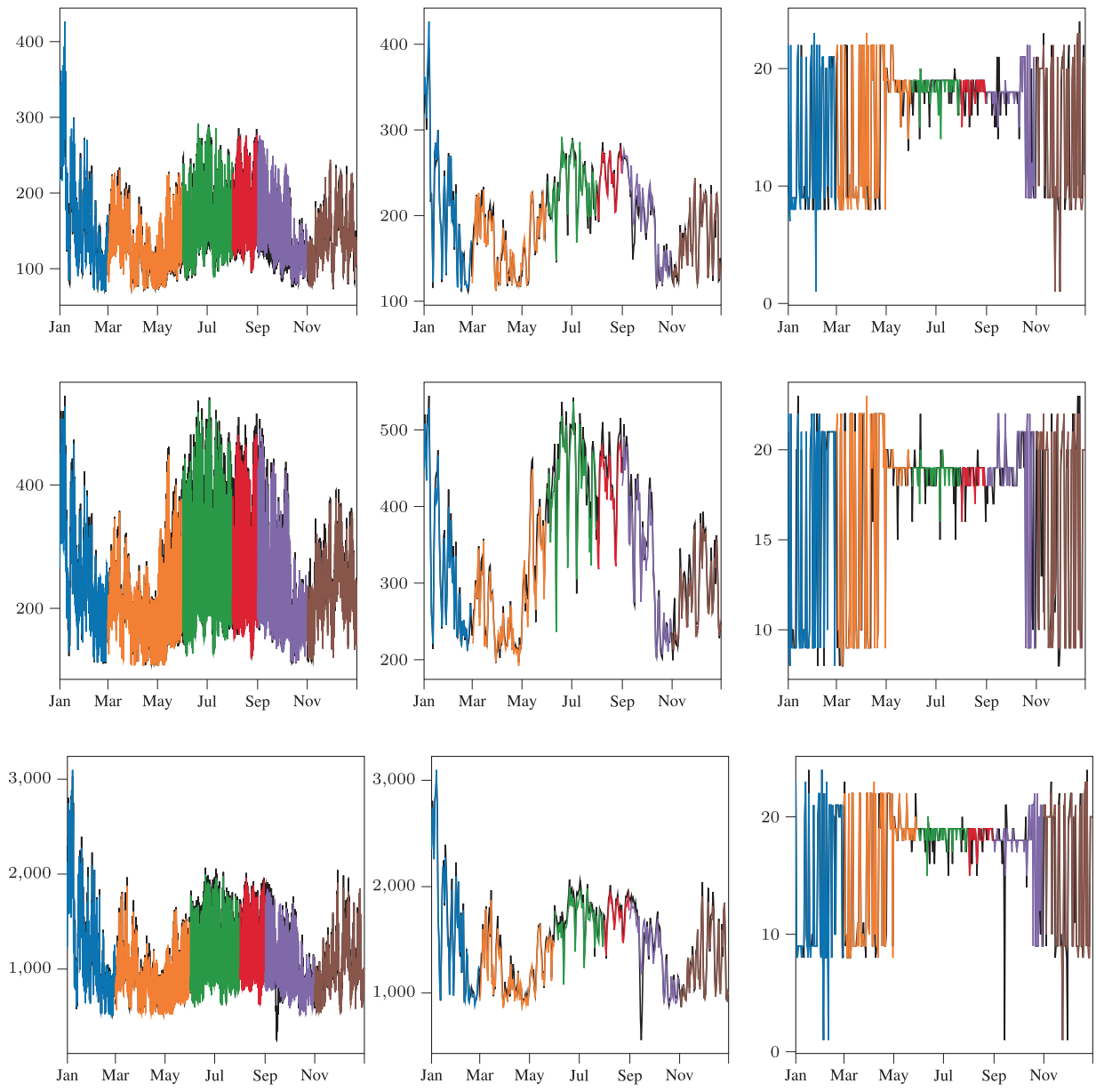

FIGURE 3 The predicted values of the conditional Invertible Neural Network (cINN) using transfer learning (TL) for the three tasks of each time series for the year 2018. To visualise each round separately, we provided a different colour to each. The ground truth is in black. Left [Shape], Middle [Magnitude], Right [Timing]. (a) LDC1. (b) LDC2. (c) LDC3.

4.3 | Computational costs

Regarding the computational costs, we observe that the cINN is more expensive than the benchmarks. However, a disclaimer is that we use different frameworks to implement the benchmarks and the proposed solution, which might cause measurement distortions. Nevertheless, the computational costs must be reduced concerning scalability and the high number of time series in the electrical grid. Perhaps the usage of GPUs instead of CPUs for training can achieve this. Furthermore, using a global forecasting model instead of training separate local cINNs for each time series could save computational costs.

4.4 | Further potential of cINNs

We aim to highlight two aspects regarding the further potential of generative models such as the cINN. First, the architecture of the cINN allows to easily create probabilistic forecasts. Thus, future work can evaluate this approach. Second, the latent space could contain interesting regions or directions. For example, there could be a direction that describes the base load or seasonal variations. Leveraging and exploiting such directions in the latent space can enhance the forecast quality and support the usage of cINNs as global forecasting models.

TABLE 5 Benchmarking. The best values are bold.

			LDC1				LDC2				LDC3			
	Round	Time span	SVR	NN	cINN-2 TL	Ensemble	SVR	NN	cINN-2 TL	Ensemble	SVR	NN	cINN-2 TL	Ensemble
Shape	1	01–02.2018	0.097	0.131	0.073	**0.072**	0.115	0.108	0.065	**0.063**	0.088	0.106	**0.049**	0.057
	2	03–05.2018	0.116	0.108	0.070	**0.068**	0.118	0.099	**0.074**	0.076	0.098	0.088	**0.052**	0.062
	3	06–07.2018	0.078	0.092	**0.051**	**0.051**	0.090	0.077	0.058	**0.056**	0.068	0.066	**0.038**	0.043
	4	08.2018	0.089	0.077	**0.043**	0.044	0.101	0.076	**0.069**	**0.069**	0.076	0.069	**0.033**	0.045
	5	09–10.2018	0.096	0.102	**0.053**	0.056	0.122	0.107	**0.074**	**0.074**	0.123	0.110	**0.061**	0.085
	6	11–12.2018	0.117	0.119	**0.074**	0.083	0.190	0.201	**0.167**	0.169	0.106	0.110	**0.072**	0.096
	Total	01–12.2018	0.099	0.105	**0.061**	0.062	0.123	0.111	**0.084**	0.085	0.093	0.091	**0.051**	0.064
Magnitude	1	01–02.2018	4.789	3.538	3.198	**3.183**	4.322	3.524	2.630	**2.491**	6.189	4.403	**2.794**	2.968
	2	03–05.2018	6.212	4.193	3.717	**3.655**	6.273	4.329	3.498	**3.183**	6.346	4.461	**3.453**	4.131
	3	06–07.2018	5.074	4.023	**3.260**	3.290	8.171	**4.456**	4.491	4.476	2.702	3.722	3.873	**3.423**
	4	08.2018	2.407	**1.770**	2.152	1.780	5.604	4.303	4.977	**3.928**	2.113	2.010	3.124	**2.296**
	5	09–10.2018	6.770	5.564	5.101	**5.448**	5.602	4.744	3.339	**3.660**	8.819	9.133	8.375	**8.062**
	6	11–12.2018	5.444	5.157	**3.346**	3.771	6.102	4.903	**4.192**	4.348	4.940	3.601	**3.240**	3.434
	Total	01–12.2018	5.116	4.041	**3.462**	3.521	6.013	4.376	3.855	**3.681**	5.185	4.555	4.143	**4.052**
Timing	1	01–02.2018	1.797	1.678	**1.271**	1.471	1.475	1.169	**0.898**	1.570	0.847	1.119	**0.559**	0.936
	2	03–05.2018	1.326	1.130	**0.696**	1.253	1.543	**0.815**	0.848	1.544	0.946	0.804	**0.533**	1.012
	3	06–07.2018	0.967	1.098	0.836	**0.821**	0.852	0.721	0.656	**0.586**	0.623	0.852	0.607	**0.587**
	4	08.2018	1.548	1.677	1.226	**1.126**	1.000	0.613	0.581	**0.541**	1.258	1.290	**0.548**	0.693
	5	09–10.2018	1.541	1.787	**1.279**	1.417	1.426	1.164	**1.033**	**1.033**	1.820	1.475	**1.311**	1.653
	6	11–12.2018	2.098	2.459	**1.770**	2.538	2.393	2.197	**1.590**	2.099	1.738	1.590	**1.180**	2.038
	Total	01–12.2018	1.546	1.638	**1.180**	1.438	1.448	1.113	**0.934**	1.229	1.205	1.189	**0.790**	1.153

TABLE 6 Training times.

					cINN-1		cINN-2	
	Round	Time span	SVR	NN	No TL	TL	No TL	TL
LDC1	1	01–02.2018	19	257	962	3607	912	3631
	2	03–05.2018	19	96	969	3623	988	3708
	3	06–07.2018	21	128	1059	3814	1085	3909
	4	08.2018	25	108	1336	4089	1327	4156
	5	09–10.2018	25	100	1357	4165	1407	4089
	6	11–12.2018	26	80	1443	3648	1285	3721
LDC2	1	01–02.2018	17	246	962	3685	970	3649
	2	03–05.2018	20	105	1027	3762	1031	3635
	3	06–07.2018	21	101	1056	3431	1016	3656
	4	08.2018	23	71	1286	4020	1310	3880
	5	09–10.2018	22	143	1303	3764	1300	3870
	6	11–12.2018	25	78	1338	4000	1434	4114
LDC3	1	01–02.2018	15	206	753	2637	723	2661
	2	03–05.2018	16	91	79	2722	791	2652
	3	06–07.2018	18	110	818	2705	807	2677
	4	08.2018	19	77	944	2825	925	2767
	5	09–10.2018	20	50	965	2782	953	2841
	6	11–12.2018	21	98	1004	2846	1000	2829

TABLE 7 First five teams on each task.

Position	Overall	Magnitude	Timing	Shape
1	Amperon	Amperon	**KIT-IAI**	**KIT-IAI**
2	**KIT-IAI**	Overfitter	Amperon	Amperon
3	Overfitters	Peaky-finders	BelindaTrotta	Overfitters
4	Peaky-fitters	Team SGEM-KIT	Overfitters	X-Mines
5	X-Mines	**KIT-IAI**	X-Mines	SheenJavan

5 | CONCLUSION

This paper presents a peak load forecasting method based on a cINN. The cINN is conditioned on various exogenous features, including temperature features, forecasts of statistical features, and calendar features. The results of the BigDEAL challenge indicate that this is a promising approach, at least for forecasting the peak's shape and timing. However, the approach must be improved to provide better peak magnitude forecasts.

Thus, future work may focus on improving the peak's magnitude forecast and reducing the forecast bias of the cINN. Furthermore, future work needs to reduce the computational effort to enable better scalability of the proposed method. An interesting approach to improve the scalability would be to extend the proposed method to a global forecasting model. Finally, it would also be very interesting to apply the proposed

method to other domains (e.g. traffic planning) in which peak forecasting is important.

AUTHOR CONTRIBUTIONS

Benedikt Heidrich: Conceptualisation; methodology; software, investigation; writing - original draft; visualisation. **Matthias Hertel:** Conceptualisation; methodology; software; investigation; writing - original draft. **Oliver Neumann:** Conceptualisation; methodology; software; investigation; writing - original draft. **Veit Hagenmeyer:** Funding acquisition; writing - review & editing; supervision. **Ralf Mikut:** Writing - review & editing; supervision.

ACKNOWLEDGEMENTS

This project is funded by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI and the Helmholtz Association under the Program "Energy System Design". Furthermore, we would like to thank Dr. Tao Hong, Shreyashi Shukla, and the rest of Dr. Tao Hong's team for organising the BigDEAL Challenge.

Open Access funding enabled and organized by Projekt DEAL.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have ap-peared to influence the work reported in this paper. This article does not contain any studies involving humans or animals per-formed by any of the authors.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available for the BigDEAL competition. Thus, the organisers of the challenge may share the data.

ORCID

Benedikt Heidrich https://orcid.org/0000-0002-1923-0848
Matthias Hertel https://orcid.org/0000-0002-0814-766X
Oliver Neumann https://orcid.org/0000-0003-4438-300X
Veit Hagenmeyer https://orcid.org/0000-0002-3572-9083
Ralf Mikut https://orcid.org/0000-0001-9100-5496

REFERENCES

1. Haida, T., Muto, S.: Regression based peak load forecasting using a transformation technique. IEEE Trans. Power Syst. 9(4), 1788–1794 (1994). https://doi.org/10.1109/59.331433
2. Berrisch, J., Narajewski, M., Ziel, F.: High-resolution peak demand estimation using generalized additive models and deep neural networks. Energy AI 13, 100236 (2023). https://doi.org/10.1016/j.egyai.2023.100236
3. Rubasinghe, O., et al.: A novel sequence to sequence data modelling based CNN-LSTM algorithm for three years ahead monthly peak load forecasting. IEEE Trans. Power Syst. 39(1), 1932–1947 (2024). https://doi.org/10.1109/tpwrs.2023.3271325
4. Zhu, K., et al.: LSTM enhanced by dual-attention-based encoder-decoder for daily peak load forecasting. Elec. Power Syst. Res. 208, 107860 (2022). https://doi.org/10.1016/j.epsr.2022.107860
5. Goia, A., May, C., Fusai, G.: Functional clustering and linear regression for peak load forecasting. Int. J. Forecast. 26(4), 700–711 (2010). https://doi.org/10.1016/j.ijforecast.2009.05.015
6. Huang, Y., et al.: Multivariate empirical mode decomposition based hybrid model for day-ahead peak load forecasting. Energy 239, 122245 (2022). https://doi.org/10.1016/j.energy.2021.122245
7. Amara-Ouali, Y., et al.: Daily peak electrical load forecasting with a multi-resolution approach. Int. J. Forecast. 39(3), 1272–1286 (2023). https://doi.org/10.1016/j.ijforecast.2022.06.001
8. Hong, T., Pinson, P., Fan, S.: Global energy forecasting competition 2012. Int. J. Forecast. 30(2), 357–363 (2014). https://doi.org/10.1016/j.ijforecast.2013.07.001
9. Hong, T., et al.: Probabilistic energy forecasting: global energy forecasting competition 2014 and beyond. Int. J. Forecast. 32(3), 896–913 (2016). https://doi.org/10.1016/j.ijforecast.2016.02.001
10. Hong, T., Xie, J., Black, J.: Global energy forecasting competition 2017: hierarchical probabilistic load forecasting. Int. J. Forecast. 35(4), 1389–1399 (2019). https://doi.org/10.1016/j.ijforecast.2019.02.006
11. Charlton, N., Singleton, C.: A refined parametric model for short term load forecasting. Int. J. Forecast. 30(2), 364–368 (2014). https://doi.org/10.1016/j.ijforecast.2013.07.003
12. Lloyd, J.R.: GEFCom2012 hierarchical load forecasting: gradient boosting machines and Gaussian processes. Int. J. Forecast. 30(2), 369–374 (2014). https://doi.org/10.1016/j.ijforecast.2013.07.002
13. Nedellec, R., Cugliari, J., Goude, Y.: GEFCom2012: electric load forecasting and backcasting with semi-parametric models. Int. J. Forecast. 30(2), 375–381 (2014). https://doi.org/10.1016/j.ijforecast.2013.07.004
14. Gaillard, P., Goude, Y., Nedellec, R.: Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting. Int. J. Forecast. 32(3), 1038–1050 (2016). https://doi.org/10.1016/j.ijforecast.2015.12.001
15. Dordonnat, V., Pichavant, A., Pierrot, A.: GEFCom2014 probabilistic electric load forecasting using time series and semi-parametric regression models. Int. J. Forecast. 32(3), 1005–1011 (2016). https://doi.org/10.1016/j.ijforecast.2015.11.010
16. Xie, J., Hong, T.: GEFCom2014 probabilistic electric load forecasting: an integrated solution with forecast combination and residual simulation. Int. J. Forecast. 32(3), 1012–1016 (2016). https://doi.org/10.1016/j.ijforecast.2015.11.005
17. Kanda, I., Veguillas, J.M.Q.: Data preprocessing and quantile regression for probabilistic load forecasting in the GEFCom2017 final match. Int. J. Forecast. 35(4), 1460–1468 (2019). https://doi.org/10.1016/j.ijforecast.2019.02.005
18. Smyl, S., Hua, N.G.: Machine learning methods for GEFCom2017 probabilistic load forecasting. Int. J. Forecast. 35(4), 1424–1431 (2019). https://doi.org/10.1016/j.ijforecast.2019.02.002
19. Abdel-Nasser, M., Mahmoud, K.: Accurate photovoltaic power forecasting models using deep LSTM-RNN. Neural Comput. Appl. 31(7), 2727–2740 (2019). https://doi.org/10.1007/s00521-017-3225-z
20. Huang, X., et al.: Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and Bi-LSTM. Energy 246, 123403 (2022). https://doi.org/10.1016/j.energy.2022.123403
21. Panapongpakorn, T., Banjerdpongchai, D.: Short-term load forecast for energy management systems using time series analysis and neural network method with average true range. In: First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), pp. 86–89 (2019). https://ieeexplore.ieee.org/document/8646068
22. Zheng, W., Chen, G.: An accurate GRU-based power time-series prediction approach with selective state updating and stochastic optimization. IEEE Trans. Cybern. 52(12), 13902–13914 (2022). https://doi.org/10.1109/tcyb.2021.3121312
23. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017). https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

24. Lim, B., et al.: Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. Int. J. Forecast. 37(4), 1748–1764 (2021). https://doi.org/10.1016/j.ijforecast.2021.03.012
25. Zhou, H., et al.: Informer: beyond efficient transformer for long sequence time-series forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 11106–11115 (2021). https://ojs.aaai.org/index.php/AAAI/article/view/17325
26. Wu, H., et al.: Autoformer: decomposition transformers with auto-correlation for long-term series forecasting. In: Advances in Neural Information Processing Systems, vol. 34, pp. 22419–22430 (2021). https://proceedings.neurips.cc/paper_files/paper/2021/file/bcc0d400288793e8bdcd7c19a8ac0c2b-Paper.pdf
27. Garza, A., Mergenthaler-Canseco, M.: TimeGPT-1 (2023). arXiv preprint arXiv:231003589
28. Hertel, M., et al.: Evaluation of transformer architectures for electrical load time-series forecasting. In: Proceedings 32. Workshop Computational Intelligence (2022). https://publikationen.bibliothek.kit.edu/1000154155/149936789
29. Hertel, M., et al.: Transformer training strategies for forecasting multiple load time series. Energy Infor. 6(S1), 20 (2023). https://doi.org/10.1186/s42162-023-00278-z
30. Giacomazzi, E., Haag, F., Hopf, K.: Short-Term Electricity Load Forecasting Using the Temporal Fusion Transformer: Effect of Grid Hierarchies and Data Sources (2023). arXiv preprint arXiv:230510559
31. Huy, P.C., et al.: Short-term electricity load forecasting based on temporal fusion transformer model. IEEE Access 10, 106296–106304 (2022). https://doi.org/10.1109/access.2022.3211941
32. Salinas, D., et al.: DeepAR: probabilistic forecasting with autoregressive recurrent networks. Int. J. Forecast. 36(3), 1181–1191 (2020). https://doi.org/10.1016/j.ijforecast.2019.07.001
33. Oreshkin, B.N., et al.: N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (2019). arXiv preprint arXiv:190510437. https://arxiv.org/abs/1905.10437
34. Challu, C., et al.: N-HiTS: Neural Hierarchical Interpolation for Time Series Forecasting (2022). arXiv preprint arXiv:220112886. https://arxiv.org/abs/2201.12886
35. Heidrich, B., et al.: Forecasting energy time series with profile neural networks. In: Proceedings of the Eleventh ACM International Conference on Future Energy Systems, pp. 220–230 (2020). https://dl.acm.org/doi/abs/10.1145/3396851.3397683
36. Heidrich, B., et al.: ProbPNN: Enhancing Deep Probabilistic Forecasting with Statistical Information (2023). arXiv preprint arXiv:230202597. https://arxiv.org/abs/2302.02597
37. Makridakis, S., Spiliotis, E., Assimakopoulos, V.: The M4 Competition: 100,000 time series and 61 forecasting methods. Int. J. Forecast. 36(1), 54–74 (2020). M4 Competition. https://doi.org/10.1016/j.ijforecast.2019.04.014
38. Smyl, S.: A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. Int. J. Forecast. 36(1), 75–85 (2020). M4 Competition. https://doi.org/10.1016/j.ijforecast.2019.03.017
39. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27, pp. 4089–4099 (2014). https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf
40. Yoon, J., Jarrett, D., van der Schaar, M.: Time-series generative adversarial networks. In: Advances in Neural Information Processing Systems, vol. 32, pp. 5508–5518 (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf
41. Xu, T., et al.: COT-GAN: generating sequential data via causal optimal transport. In: Advances in Neural Information Processing Systems, vol. 33, pp. 8798–8809 (2020)
42. Heidrich, B., et al.: Controlling non-stationarity and periodicities in time series generation using conditional invertible neural networks. Appl. Intell. 53(8), 1–18 (2022). https://doi.org/10.1007/s10489-022-03742-7
43. Ardizzone, L., et al.: Guided Image Generation with Conditional Invertible Neural Networks (2019). arXiv preprint arXiv:190702392. https://arxiv.org/abs/1907.02392
44. Kingma, D.P., Dhariwal, P.: Glow: generative flow with invertible 1x1 convolutions. In: Advances in Neural Information Processing Systems, vol. 31, pp. 10215–10224 (2018). https://proceedings.neurips.cc/paper_files/paper/2018/file/d139db6a236200b21cc7f752979132d0-Paper.pdf
45. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830 (2011). https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf
46. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems, vol. 32, pp. 8024–8035 (2019). https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

How to cite this article: Heidrich, B., et al.: Using conditional Invertible Neural Networks to perform mid-term peak load forecasting. IET Smart Grid. 1–13 (2024). https://doi.org/10.1049/stg2.12169