



Generating probabilistic forecasts from arbitrary point forecasts using a conditional invertible neural network

Kaleb Phipps¹ · Benedikt Heidrich¹ · Marian Turowski¹ · Moritz Wittig^{1,2} · Ralf Mikut¹ · Veit Hagenmeyer¹

Accepted: 16 February 2024
© The Author(s) 2024

Abstract

In various applications, probabilistic forecasts are required to quantify the inherent uncertainty associated with the forecast. However, many existing forecasting methods still only generate point forecasts. Although methods exist to generate probabilistic forecasts from these point forecasts, these are often limited to prediction intervals or must be trained together with a specific point forecast. Therefore, the present article proposes a novel approach for generating probabilistic forecasts from arbitrary point forecasts. In order to implement this approach, we apply a conditional Invertible Neural Network (cINN) to learn the underlying distribution of the data and then combine the uncertainty from this distribution with an arbitrary point forecast to generate probabilistic forecasts. We evaluate our approach by generating probabilistic forecasts from multiple point forecasts and comparing these forecasts to six probabilistic benchmarks on four data sets. We show that our approach generally outperforms all benchmarks with regard to CRPS and Winkler scores and generates probabilistic forecasts with the narrowest prediction intervals whilst remaining reasonably calibrated. Furthermore, our approach enables simple point forecasting methods to rank highly in the Global Energy Forecasting Competition 2014.

Keywords Probabilistic forecasting · Uncertainty quantification · Conditional invertible neural networks · Machine learning · Normalising flows

1 Introduction

Probabilistic forecasts are required to quantify the inherent uncertainty associated with any prediction of the future [23, 45]. These probabilistic forecasts are crucial for many applications such as stabilising energy systems [11], managing congestion in traffic systems [39], or sizing servers of web applications to cope with a certain number of daily visits [36]. Despite this necessity for probabilistic forecasts, many modern forecasting methods still generate point forecasts [44]. Although many recent machine learning libraries offer support for probabilistic loss functions to simplify the generation of probabilistic forecasts, this may not be possible if an existing point forecast model that cannot easily be modified or retrained is already in use.

One solution to overcome this challenge is to generate probabilistic forecasts based on these existing point forecasts. For many years, such forecasts have been generated

by analysing the residual errors of the point forecast. Based on these errors' standard deviation or quantiles, prediction intervals can be calculated to generate probabilistic forecasts [29, 56]. Moreover, such probabilistic forecasts can be generated by using machine learning methods exploiting the residual errors [5, 54], by applying the Bayesian theory of probability to a point method [37], or by considering Monte-Carlo sampling methods [3]. Although these methods may be effective, they also have various limitations. For example, the prediction interval-based approaches can only generate prediction intervals as probabilistic forecasts, while machine learning methods depend on the point forecast and must be retrained if the point forecast is altered. Ideally, such probabilistic forecasts should be generated directly from arbitrary point forecasts and should not require retraining if the point forecast changes.

Therefore, in the present article, we present an approach that generates probabilistic forecasts from arbitrary point forecasts by using a Conditional Invertible Neural Network (cINN) to learn the underlying distribution of the time series data. Since time series have an inherent component of randomness [29], we propose using this uncertainty

✉ Kaleb Phipps
kaleb.phipps@kit.edu

Extended author information available on the last page of the article

within the distribution of the time series data to generate probabilistic forecasts. However, the underlying system responsible for this uncertainty typically generates observations of an unknown probability distribution. Therefore, with our approach, we first map this unknown probability distribution of the underlying time series data to a known and tractable distribution by applying a cINN. Then, we use the output of a trained arbitrary point forecast method as an input to the trained cINN and consider the representation of this forecast in the known and tractable distribution. We then analyse the neighbourhood of this representation in the known and tractable distribution to quantify the uncertainty associated with the representation. Finally, we use the backward pass of the cINN to convert this uncertainty information into the forecast. In our approach, the cINN is trained independently of the point forecast and must not be retrained when the point forecast is altered.

Thus, the main contribution of the present article is twofold. First, we provide a novel approach for generating probabilistic forecasts from arbitrary point forecasts whose training is independent of the point forecast. Second, we empirically evaluate the approach using different data sets from various domains. In this empirical evaluation, we compare our approach to six probabilistic benchmarks, evaluate multiple metrics, and recreate the Global Energy Forecasting Competition 2014 (GEFCOM2014) competition setting.

The remainder of our article is structured as follows. First, we present related work and highlight the research gap that the present article addresses in Section 2. In Section 3, we then explain our approach in detail and highlight how we use a cINN to generate probabilistic forecasts from an arbitrary point forecast. We detail the experimental setup in Section 4, before presenting our results in Section 5. In Section 6 we discuss our evaluation and key insights. Finally, we conclude and suggest possible directions for future work in Section 7.

2 Related work

Our article is closely related to two research fields: previous work that generates probabilistic forecasts based on point forecasts and previous work focusing on probabilistic forecasts using a cINN. Table 1 presents an overview of the identified related articles, and in this section, we present and discuss these articles in more detail and highlight the research gap the present article addresses.

Generating probabilistic forecasts from point forecasts

Determining the uncertainty associated with a point prediction is one of the key research areas of uncertainty quantification [52]. Many methods focus on generating probabilistic prediction intervals from existing point forecasts

by using the residual errors between the point forecast and the true value [29]. These prediction intervals can be generated by assuming a Gaussian distribution of the errors [29], using the empirical distribution of the errors [56], or considering nonconformity errors [8, 53, 59]. While effective, these methods are designed to generate prediction intervals rather than approximate the full probability distribution, which may be a limitation. Furthermore, if the point forecaster used is changed, new residual or nonconformity errors must be calculated to apply these methods. Although this calculation is not a retraining process, it does require additional effort.

Similar approaches also use residual errors in combination with further machine learning algorithms. [5], for example, train a neural network to forecast the standard deviation of the residual errors and generate probabilistic forecasts as realisations of a Gaussian distribution centred around the original point forecasts. Similarly, [54] use the residual errors from a point forecast to train a Generative Adversarial Network (GAN). This trained GAN is then used to generate multiple residual scenarios, which are combined with the point forecast to form probabilistic forecasts. The main limitation of both approaches is that the additional machine-learning models used to predict the uncertainty (i.e. standard deviation or residual scenarios) depend on the selected point forecast [5, 54]. Therefore, these machine-learning models must be retrained whenever the point forecast is altered.

Further approaches include a Bayesian method involving assumed priors [32], integrating uncertainty into the prediction via an ensemble of predictions [10], and considering uncertainty through Monte Carlo sampling approaches or similar [4]. The main limitation of these approaches, apart from the assumption regarding the Bayesian prior, is the computational complexity resulting from sampling or generating a large ensemble pool.

Probabilistic forecasts using cINNs

To generate probabilistic forecasts, cINNs, also referred to as normalising flows [1], are combined with other machine learning methods. [2], for example, apply normalising flows to learn the parameters of Bernstein polynomials, which are, in turn, used to generate a probabilistic forecast. Moreover, [46] combine normalising flows with recurrent neural networks to generate probabilistic forecasts. Normalising flows are also combined with quantile regression networks and copulas [55], or used to generate a conditional approximation of a Gaussian mixture model [31] to improve the accuracy of the resulting probabilistic forecasts. Whilst these methods are all effective, normalising flows are used to enrich existing complex probabilistic forecasting methods, but not to provide probabilistic forecasts themselves.

An alternative method that directly uses normalising flows in the context of probabilistic forecasts is to learn multi-

dimensional distributions of electricity price differences to predict the trajectory of intraday electricity prices [9]. Similarly, normalising flows may be applied multiple times to generate scenario-based probabilistic forecasts [15, 21, 60], or to generate a proxy for weather ensemble prediction systems based on numerical weather prediction models [18]. These methods use the generative nature of normalising flows to generate multiple predictions drawn from the same distribution. However, the forecasts are only probabilistic as an ensemble, with each individual forecast still being a point forecast. Furthermore, these forecasts assume that the underlying learned distribution remains constant and only partly considers external features. Finally, these methods all focus on directly generating probabilistic forecasts. Therefore, such methods cannot be applied to generate probabilistic forecasts from existing, well-designed point forecasts.

Research gap

As shown in Table 1, we identify a lack of existing work that directly generates probabilistic forecasts from arbitrary point forecasts without the training process being dependent on this point forecast or being limited to only generating prediction intervals. In the present article, we aim to fill this research gap by presenting an easy-to-use approach described in the following section.

3 Generating probabilistic forecasts with a cINN

To generate probabilistic forecasts from arbitrary point forecasts, we directly apply the uncertainty in the underlying time series. This uncertainty usually reflects the inherent randomness or unpredictability of the measured underlying system. However, this underlying system typically generates observations of an unknown distribution. Although this data is not random, the distribution is still unknown, and it is challenging to include the corresponding uncertainty directly in a forecast.

To solve this challenge, we aim to find a bijective mapping from the unknown distribution to a known and tractable distribution. Since many time series are affected by exogenous features such as weather, this bijective mapping should also be able to consider such exogenous features, as shown in Fig. 1. If such a mapping g exists, we will be able to map a point forecast from the unknown distribution to its representation in a known and tractable distribution. In the known and tractable distribution, we could then analyse the neighbourhood of this representation and gain information about its uncertainty. Finally, we could map this uncertainty information back to the unknown distribution using the inverse mapping g^{-1} to generate probabilistic forecasts.

Table 1 An overview of previous research related to the present article. None of the identified articles proposes methods capable of generating probabilistic forecasts from existing point forecasts without being limited to only generating prediction intervals or involving a training process that is dependent on the training of the point forecast

Article	Probabilistic Forecast from Point Forecasts	Not Limited to Prediction Interval	Training Independent of Point Forecast
Chernozhukov et al [8]	✓	✗	(✓)
Hyndman and Athanasopoulos [29]	✓	✗	(✓)
Stankeviciute et al [53]	✓	✗	(✓)
Williams and Goodman [56]	✓	✗	(✓)
Zaffran et al [59]	✓	✗	(✓)
Camporeale et al [5]	✓	✓	✗
Kaplan and Huang [32]	✓	✓	✗
Wang et al [54]	✓	✓	✗
Camporeale et al [4]	(✓)	✓	✗
Cramer et al [10]	(✓)	✓	✗
Arpogaus et al [2]	✗	✓	✓
Cramer et al [9]	✗	✓	✓
Dumas et al [15]	✗	✓	✓
Fanfarillo et al [18]	✗	✓	✓
Ge et al [21]	✗	✓	✓
Jamgochian et al [31]	✗	✓	✓
Rasul et al [46]	✗	✓	✓
Wen and Torkkola [55]	✗	✓	✓
Zhang and Zhang [60]	✗	✓	✓

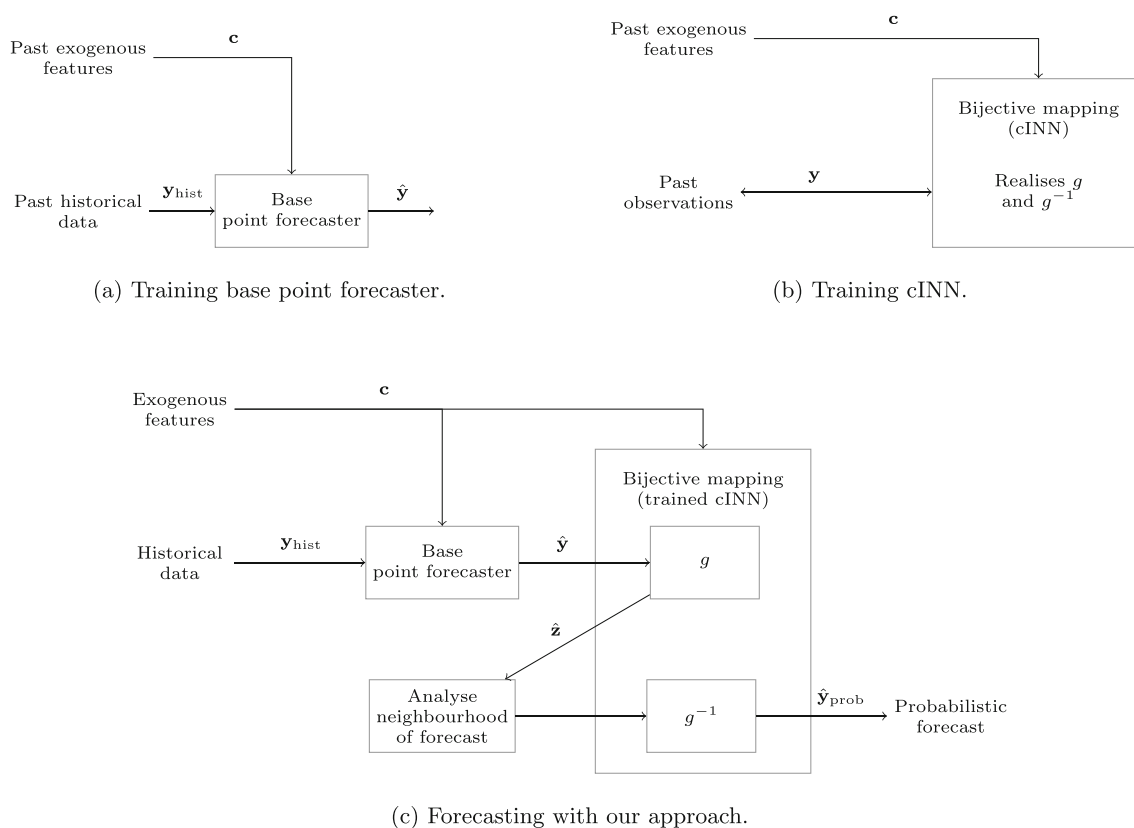


Fig. 1 Overview of the proposed approach. In the first step, an arbitrary point forecaster (a) and cINN (b) are trained independently, both considering exogenous features, past observations, or past historical data. To generate a probabilistic forecast, as shown in (c), the arbitrary point forecaster first generates a point forecast based on historical data and exogenous features. The resulting point forecast is combined with the exogenous features as inputs to a bijective mapping realised by the trained cINN. This mapping generates a representation of the forecast in a known and tractable distribution. We analyse the neighbourhood of this known and tractable representation to gain information about its uncertainty. Finally, we map this representation back to the unknown distribution to generate a probabilistic forecast

In this section, we demonstrate that this mapping g does exist under certain conditions, and we show how this mapping can be used to generate probabilistic forecasts. We then explain how to apply this approach with a cINN, starting with the training of this cINN, before describing how we generate probabilistic forecasts using arbitrary point forecasts.

3.1 Including uncertainty from the underlying distribution of the data

This section demonstrates that a bijective mapping from an unknown distribution in a known and tractable distribution exists. Given the existence of this mapping, we highlight the equivalence of the uncertainty in the image and the inverse image of the considered mapping. Finally, we describe how this mapping is realised with a cINN

Bijection mapping

To introduce the bijective mapping, let us consider a times series $\mathbf{y} = \{y_t\}_{t \in T}$ consisting of T observations as realisations of a random variable $Y \sim f_Y(\mathbf{y})$ with a probability

density function (PDF) $f_Y(\mathbf{y})$ in the realisation space \mathbb{Y} . Furthermore, we consider a bijective mapping $g : \mathbb{Y} \rightarrow \mathbb{Z}$ from the realisation space \mathbb{Y} to the space of the tractable distribution \mathbb{Z} where $\mathbf{y} \mapsto g(\mathbf{y}, \circ) = \mathbf{z}$, and g is a continuously differentiable function.¹ To calculate the PDF $f_Z(\mathbf{z})$ in terms of $f_Y(\mathbf{y})$, we can apply the change of variables formula [12, 42], i.e.

$$f_Z(\mathbf{z}) = f_Y(g^{-1}(\mathbf{z}, \circ)) \left| \det \left(\frac{\partial g^{-1}}{\partial \mathbf{z}} \right) \right|, \quad (1)$$

where $\frac{\partial g^{-1}}{\partial \mathbf{z}}$ is the Jacobian matrix. Since g is bijective, this equation describes a bijective mapping from the unknown distribution $f_Y(\mathbf{y})$ to the known and tractable distribution $f_Z(\mathbf{z})$. Therefore, the change of variable formula provides us with the required mapping.

¹ The function g can include further parameters apart from \mathbf{y} , such as exogenous information. These further parameters are indicated via \circ .

Equivalence of uncertainty

After introducing the bijective mapping, we need to show the equivalence of the uncertainty in the unknown distribution and known tractable distribution when applying (1). More specifically, we show the equivalence of quantiles in both the realisation space and the tractable distribution space since quantiles serve as a non-parametric representation of the uncertainty.

To show this equivalence, we first consider the cumulative distribution function (CDF) of the random variable $Z = g(Y, \circ) \sim f_Z(\mathbf{z})$, defined as

$$F_Z(\mathbf{z}) = \int_{-\infty}^{\mathbf{z}} f_Z(\mathbf{u}) d\mathbf{u}. \tag{2}$$

If we use the expression for $f_Z(\mathbf{z})$ from the change of variables formula (1) in the definition of the CDF (2), we obtain

$$F_Z(\mathbf{z}) = \int_{-\infty}^{\mathbf{z}} f_Y(g^{-1}(\mathbf{u}, \circ)) \left| \det \left(\frac{\partial g^{-1}}{\partial \mathbf{u}} \right) \right| d\mathbf{u}, \tag{3}$$

describing the CDF of $F_Z(\mathbf{z})$ in terms of the CDF $F_Y(\mathbf{y})$. Since g is, per definition, a continuously differentiable function, we can apply integration by substitution for multiple variables to rewrite (3) as

$$F_Z(\mathbf{z}) = \int_{-\infty}^{g^{-1}(\mathbf{z})} f_Y(\mathbf{v}) d\mathbf{v} = F_Y(g^{-1}(\mathbf{z}, \circ)),$$

which is simply the CDF of Y evaluated at the inverse of g . Further, the quantiles \mathbf{z}_α of Z are defined by the inverse of the CDF, i.e.

$$\begin{aligned} \mathbf{z}_\alpha &= F_Z^{-1}(\alpha) = \inf \mathbf{z} \mid F_Z(\mathbf{z}) \geq \alpha \\ &= \inf \mathbf{z} \mid F_Y(g^{-1}(\mathbf{z}, \circ)) \geq \alpha \end{aligned}$$

where \inf refers to the infimum, the smallest value of \mathbf{z} that fulfils the condition, and $\alpha \in [0, 1]$ is the considered quantile. Consequently, if we know that the α quantile of F_Z is \mathbf{z}_α , then we can also calculate the α quantile of F_Y as $g^{-1}(\mathbf{z}_\alpha, \circ) = \mathbf{y}_\alpha$. From this follows an equivalence between the quantiles of Z and the quantiles of Y , which implies an equivalence in the uncertainty.

Given the mathematical equivalence of the uncertainty in the two considered distributions, we can include uncertainty in a tractable and known distribution $f_Z(\mathbf{z})$ and use the inverse mapping $g^{-1} : \mathbb{Z} \rightarrow \mathbb{Y}$ to map this uncertainty to the original distribution $f_Y(\mathbf{y})$.

Realising the bijective mapping

To realise this bijective mapping g , we use a cINN [1, 27]. A cINN is a neural network that consists of multiple specially designed conditional affine coupling blocks [1]. As shown by [1], these coupling blocks ensure that the mapping

$g : \mathbb{Y} \rightarrow \mathbb{Z}$ learnt by the cINN is bijective. Furthermore, with the conditional information, the cINN is able to consider additional information, such as exogenous features, extracted statistical features from the time series, or calendar information, when learning the mapping [1, 27]. As a result, the cINN is designed to learn an approximation of $f_Z(\mathbf{z})$ and a mapping g , which is per definition bijective, thus ensuring we can apply (1) as described previously. Since cINNs are specifically designed to efficiently calculate the inverse of a function [1], a well-trained cINN should be capable of learning the bijective mapping g , even if this mapping is non-trivial.

3.2 Applying our approach

In the following, we describe how we realise the inclusion of uncertainty with a cINN.² We first detail how we train a cINN that learns the distribution of the underlying data. Second, we describe how we use this trained cINN to generate probabilistic forecasts.

Training

To apply our approach, firstly an arbitrary base point forecaster $\mathcal{F}(\circ, \psi)$ with trainable parameters ψ must be trained, as shown in Fig. 1a. However, since the cINN is trained on historical time series realisations and not with the point forecasts, the cINN is trained completely independently of the point forecaster and must not be retrained if the point forecast is altered. Furthermore, the cINN in our approach can be applied to any arbitrary point forecast, including a point forecast that has been previously trained and implemented. Since these arbitrary point forecasters may have very different training procedures, we refrain from a more detailed description of the training of the base point forecaster in the present article. As a result of this training, however, the base point forecaster $\mathcal{F}(\circ, \hat{\psi}_{\text{OPT}})$ has optimised parameters $\hat{\psi}_{\text{OPT}}$.

We use a cINN to realise the continuous differentiable function g described previously. In addition to the original realisation \mathbf{y} , we also consider conditional information \mathbf{c} as an input to the function g . This conditional information always includes calendar features such as time of the day, and day of the week, but depending on the time series may also include additional exogenous features that are available for the forecast period. Thereby, the calendar information extracted from the time series is necessary conditional information to account for the temporal dependencies of the time series, whilst the exogenous features are optional. Furthermore, statistical features extracted from the time series can also be included as conditional information. We train the cINN using past exogenous features and past observations,

² The implementation is available via <https://github.com/KIT-IAI/ProbabilisticForecastsFromArbitraryPointForecasts>.

Algorithm 1 Forecasting with our approach.

Input: $\mathcal{F}(\circ, \hat{\psi}_{\text{OPT}})$, $g(\circ, \hat{\theta}_{\text{OPT}})$, σ ▷ Trained base point forecaster, cINN, and selected σ
 $\hat{\mathbf{y}} \leftarrow \mathcal{F}(\mathbf{y}_{\text{hist}}, \mathbf{c}, \hat{\psi}_{\text{OPT}})$
 $\hat{\mathbf{z}} = g(\hat{\mathbf{y}}, \mathbf{c}, \hat{\theta}_{\text{OPT}})$
 $\hat{\mathcal{Y}}^\sigma \leftarrow$ Initialise empty set to store samples
for $i \in \{1, \dots, I\}$ **do**
 $\mathbf{r}_i \sim \mathcal{N}(0, \sigma) \leftarrow$ Initialise random noise for sampling
 $\tilde{\mathbf{z}}_i \leftarrow \hat{\mathbf{z}} + \mathbf{r}_i$
 $\tilde{\mathbf{y}}_i \leftarrow g^{-1}(\tilde{\mathbf{z}}_i, \mathbf{c}, \hat{\theta}_{\text{OPT}})$
 $\hat{\mathcal{Y}}^\sigma \leftarrow \hat{\mathcal{Y}}^\sigma \cup \tilde{\mathbf{y}}_i$
end for
 $\hat{\mathbf{y}}_{\text{prob}} \leftarrow \text{CalculateQuantiles}(\hat{\mathcal{Y}}^\sigma)$
return $\hat{\mathbf{y}}_{\text{prob}}$

as shown in Fig. 1b. The aim of the training is to ensure that the cINN learns the function g , so that resulting realisations $\mathbf{z} = g(\mathbf{y}, \mathbf{c})$ follow a known and tractable latent space distribution $f_Z(\mathbf{z})$. In our approach, we define this known and tractable latent space distribution as a multi-dimensional Gaussian distribution, where the number of dimensions is equal to the forecast horizon. Therefore, we apply the change of variables formula to derive the loss function

$$\mathcal{L}_{\text{cINN}} = \mathbb{E} \left[\frac{\|g(\mathbf{y}; \mathbf{c}, \theta)\|_2^2}{2} - \log |J| \right] + \lambda \|\theta\|_2^2,$$

where $J = \det(\partial g / \partial \mathbf{y})$ is the determinant of the Jacobian, θ is the set of all trainable parameters, and $\lambda \|\theta\|_2^2$ is an L2 regularisation [1, 27].³ Training a cINN with this loss function results in a network with the optimised parameters $\hat{\theta}_{\text{OPT}}$ and ensures that the realised latent space distribution $f_Z(\mathbf{z})$ achieves the best possible approximation of the desired multi-dimensional Gaussian distribution [1].

Forecasting

The process of generating probabilistic forecasts with our approach is shown in Algorithm 1. The process begins with the output of the trained base point forecaster

$$\hat{\mathbf{y}} = \mathcal{F}(\mathbf{y}_{\text{hist}}, \mathbf{c}, \hat{\psi}_{\text{OPT}}).$$

We then combine this output with the associated conditional information \mathbf{c} and pass it through the trained cINN to obtain a latent space representation of the output, i.e.

$$\hat{\mathbf{z}} = g(\hat{\mathbf{y}}, \mathbf{c}, \hat{\theta}_{\text{OPT}}).$$

Given this latent space representation of the point forecast, we explore the uncertainty in the neighbourhood of the forecast with

$$\tilde{\mathbf{z}}_i = \hat{\mathbf{z}} + \mathbf{r}_i, \quad i \in \{1, \dots, I\}, \quad \mathbf{r}_i \sim \mathcal{N}(0, \sigma). \quad (4)$$

³ Full details on the derivation of this loss function are presented in [1].

Using (4), we select a random noise \mathbf{r}_i from a standard normal distribution with mean 0 and variance σ and add this noise to the realisation $\hat{\mathbf{z}}$. We define the variance used for the sampling process σ as the *sampling hyperparameter*, which must be selected in advance before generating a probabilistic forecast. Due to the equivalence of uncertainty in both spaces shown in Section 3.1, we can process this perturbed sample via a backward pass of the cINN, i.e.

$$\tilde{\mathbf{y}}_i = g^{-1}(\tilde{\mathbf{z}}_i, \mathbf{c}, \hat{\theta}_{\text{OPT}}),$$

to obtain a perturbed sample in the realisation space $\tilde{\mathbf{y}}_i$. Based on the selected σ , we repeat the sampling process I times to obtain multiple realisations of $\tilde{\mathbf{z}}_i$ and, in turn, multiple realisations $\tilde{\mathbf{y}}_i$ that are all similar but not identical to the original forecast. If we combine all these samples in a set $\hat{\mathcal{Y}}^\sigma$, i.e.

$$\hat{\mathcal{Y}}^\sigma = \bigcup_{i \in I} \tilde{\mathbf{y}}_i,$$

then this set of realisations provides a representation of the uncertainty in the neighbourhood of the forecast, as schematically shown in Fig. 2 on the left. Given this set, there are multiple possibilities for generating a probabilistic forecast. We can use all the samples as an *ensemble* forecast, perform a density estimation over the samples to generate a *distribution* forecast, or calculate the quantiles of these samples. In the present paper, we calculate the quantiles of these samples in the original realisation space as schematically shown in Fig. 2 on the right. The resulting quantiles represent a probabilistic forecast $\hat{\mathbf{y}}_{\text{prob}}$, derived from the original arbitrary point forecast.

4 Experimental setup

This section describes the experimental setup we use to evaluate our approach. We first introduce the data used, before explaining the evaluation metrics. Furthermore, we describe the selected base forecasters used to generate the point forecasts, introduce the benchmarks we compare our approach to, and detail the implementation of the used cINN.

4.1 Data

We evaluate our proposed approach on four different openly available data sets. In this section, we briefly introduce each of these data sets before we describe their preprocessing.

The first considered data set is *Electricity*, namely the UCI Electricity Load Dataset⁴ [13]. From this data set, we select

⁴ <https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014>

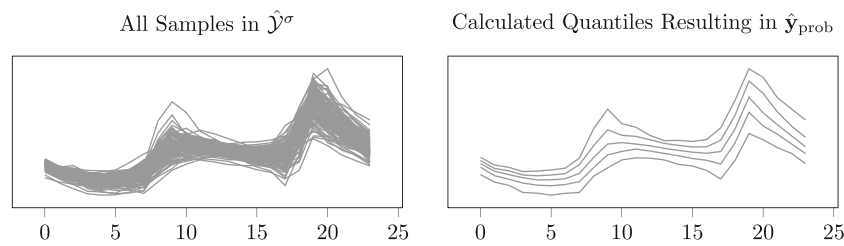


Fig. 2 A schematic representation of the probabilistic forecast generated with our approach. Initially, a set of samples represents the uncertainty. In the present paper, we then calculate the quantiles of these samples to generate the probabilistic forecast \hat{y}_{prob} . However, it would also be possible to consider all samples as an ensemble forecast or to perform density information to obtain a distribution forecast

the time series *MT_158* and resample it to an hourly resolution.

The second data set, *Price*, contains zonal electricity price data recorded at a single location at an hourly resolution and taken from the electricity price track of the GEFCom2014 [28]. To evaluate our approach on a period longer than a single day, we combine data from all tasks in the GEFCom2014 price track.

Third, we consider a *Solar* data set which contains hourly real-world solar power generation from a solar plant in Australia. This data set is taken from the solar power forecasting track of the GEFCom2014 [28] and, again, we combine data from all tasks to enable evaluation on a period longer than a day.

The fourth considered data set, *Bike*, contains hourly records of rented bikes from the UCI Bikes sharing Dataset [13, 17].⁵

We normalise each of the above data sets before creating separate test, validation, and training subsets for the training and testing of our approach. An overview of these splits and the exogenous variables considered for each data set is presented in Table 15 in Appendix A.

4.2 Evaluation metrics

When evaluating probabilistic forecasts, it is important to consider both sharpness and calibration [23]. According to [23], probabilistic forecasts should aim to maximise sharpness subject to calibration. This aim implies, for example, that a prediction interval should be as narrow as possible while still maintaining coverage close to the nominal coverage rate. Probabilistic forecasts that are too sharp provide misleading information about the uncertainty present, whilst probabilistic forecasts that only focus on calibration may not be sharp enough to deliver any useful information [23]. With these considerations in mind, we aim to evaluate our approach comprehensively, considering both sharpness, calibration, and the trade-off between these two, and therefore

consider multiple evaluation metrics. In the following, we briefly present these metrics in the order they appear in the evaluation.

Continuous ranked probability score

To evaluate the quality of the probabilistic forecasts, we consider the Continuous Ranked Probability Score (CRPS) [41]. The CRPS is a proper scoring rule that measures both the calibration and sharpness of a predictive cumulative distribution function F [22]. The CRPS is defined as

$$\text{CRPS}(F, y) = \int_{\mathbb{R}} (F(z) - \mathbb{1}\{y \leq z\})^2 dz,$$

where $\mathbb{1}\{y \leq z\}$ is the indicator function which is one if $y \leq z$ and otherwise zero. Since our approach and the benchmarks provide samples drawn from a distribution, we use the sample-based variant of the CRPS implemented in the `properscoring` library.⁶

Quantile deviation

To analyse the calibration of our forecasts, we consider the deviation of the forecast quantiles from the theoretical quantiles. We define the quantile deviation for the α -quantile as

$$\text{QD}_\alpha = \left(\frac{1}{n} \sum_{i=1}^n \mathbb{1}\{y_i \leq \hat{y}_{i,\alpha}\} \right) - \alpha,$$

where $\hat{y}_{i,\alpha}$ is the α -quantile forecast, y_i the true value, and $\mathbb{1}$ the indicator function. Ideally, QD_α should be zero for all values of α . However, a positive value indicates the quantile forecast overestimates the theoretical quantile, whilst a negative value indicates that the quantile forecast underestimates the theoretical quantile. To account for the total quantile devi-

⁵ <https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset>

⁶ <https://github.com/properscoring/properscoring>

ation across all considered quantiles $\alpha \in Q$, we calculate the Mean Absolute Quantile Deviation (MAQD) defined as

$$\text{MAQD} = \frac{1}{|Q|} \sum_{\alpha \in Q} \left| \left(\frac{1}{n} \sum_{i=1}^n \mathbb{1}\{y_i \leq \hat{y}_{i,\alpha}\} \right) - \alpha \right|.$$

Normalised prediction interval width

To measure the sharpness of the probabilistic forecasts, we consider the normalised Mean β -PI Width (nMPI(β)). The nMPI(β) is defined as

$$\text{nMPI}(\beta) = \frac{1}{\bar{y}} \left(\frac{1}{n} \sum_{i=1}^n |\hat{y}_{i, \frac{1+\beta}{2}} - \hat{y}_{i, \frac{1-\beta}{2}}| \right),$$

where $\hat{y}_{i, \frac{1+\beta}{2}}$ is the predicted upper quantile, $\hat{y}_{i, \frac{1-\beta}{2}}$ the predicted lower quantile for the forecast value \hat{y}_i , and \bar{y} the mean of the target time series. We consider the nMPI(β) to enable a comparison between different data sets.

Winkler score

To jointly assess calibration and sharpness, we assess the quality of the prediction intervals of our probabilistic forecasts with the Winkler score [57]. As defined by [29], if the $100 \cdot (1 - \alpha)\%$ prediction interval for observation i is given as $[\ell_{\alpha,i}, u_{\alpha,i}]$, then the Winkler score for the α -quantile is defined as

$$W_{\alpha,i} = \begin{cases} (u_{\alpha,i} - \ell_{\alpha,i}) + \frac{2}{\alpha}(\ell_{\alpha,i} - y_i) & y_i < \ell_{\alpha,i} \\ (u_{\alpha,i} - \ell_{\alpha,i}) & \ell_{\alpha,i} \leq y_i \leq u_{\alpha,i} \\ (u_{\alpha,i} - \ell_{\alpha,i}) + \frac{2}{\alpha}(y_i - u_{\alpha,i}) & y_i > u_{\alpha,i}, \end{cases}$$

where y_i is the true value. In this manner, a Winkler score without any violations is simply the width of the prediction interval, whilst true values falling outside the prediction interval are penalised. Therefore, low Winkler scores suggest narrow but reasonably calibrated prediction intervals. In our evaluation, we consider the Mean Winkler (MW) score across all considered quantiles $\alpha \in Q$, defined as

$$\text{MW} = \frac{1}{n |Q|} \sum_{\alpha \in Q} \sum_{i=1}^n W_{\alpha,i}.$$

Pinball loss improvement

To evaluate our approach in the recreated GEFCom2014 competition, we consider the scoring mechanism used in this competition. This mechanism relies on the Pinball Loss (PL), a scoring rule that minimises the loss when issuing a point forecast for the α -quantile [24]. For a set of considered quan-

tiles $Q = [0.01, \dots, 0.99]$, the PL is calculated with

$$\text{PL} = \frac{1}{n |Q|} \sum_{\alpha \in Q} \sum_{i=1}^n \begin{cases} (y_i - \hat{y}_{i,\alpha}) \cdot \alpha & y_i \geq \hat{y}_{i,\alpha} \\ (\hat{y}_{i,\alpha} - y_i) \cdot (1 - \alpha) & \hat{y}_{i,\alpha} > y_i, \end{cases}$$

where y_i is the true value and $\hat{y}_{i,\alpha}$ is the quantile forecast for the quantile α . For the GEFCom2014, the relative improvement of the PL compared to a given baseline forecast is considered, i.e.

$$\text{PL}_{\%} = \frac{\text{PL}_{\text{Forecast}}}{\text{PL}_{\text{Baseline}}} \cdot 100,$$

where $\text{PL}_{\text{Forecast}}$ is the PL for the considered forecast and $\text{PL}_{\text{Baseline}}$ the PL for the baseline provided in the GEFCom2014.

4.3 Selected base forecasters

Our proposed approach can be applied to forecasts from arbitrary point forecasters. Thus, we evaluate our approach on four simple and two state-of-the-art point forecasting methods. As simple base point forecasters we consider a *Linear Regression (LR)*, a *Random Forest (RF)*, a *Feed-Forward Neural Network (NN)*, and the *eXtreme Gradient Boosting (XGBoost)* Regressor. We select these methods due to their robust performance in multiple studies, e.g. [16, 19, 47, 49, 50], and [51]. The two state-of-the-art base point forecasters are *Neural Hierarchical Interpolation for Time Series Forecasting (N-HiTS)* [6] and *Temporal Fusion Transformer (TFT)* [38]. We provide implementation details for each of the selected base point forecasters in Table 16 in Appendix A.

When applying the base forecasters with the cINN to generate probabilistic forecasts, we manually select the sampling parameter σ that minimises the CRPS on the validation data set. An overview of the selected sampling parameters is presented in Table 17 in Appendix A. Furthermore, all selected base point forecasters are implemented in a pipeline with pyWATTS⁷ [26].

4.4 Probabilistic benchmarks

To assess the quality of the probabilistic forecasts generated with our approach, we compare them to multiple probabilistic benchmarks. These benchmarks can be classified into the following two groups: probabilistic forecasts generated from existing point forecasts and directly generated probabilistic forecasts. We focus on a selection of benchmarks that have achieved state-of-the-art performance whilst being relatively

⁷ <https://github.com/KIT-IAI/pyWATTS>

computationally inexpensive and therefore exclude computationally expensive benchmarks that may generalise poorly, such as Bayesian Neural Networks [30]. In the following, we introduce the benchmarks of both groups.

4.4.1 Probabilistic forecasts based on existing point forecasts

The first group of probabilistic benchmarks considers methods that generate probabilistic forecasts from existing point forecasts. All of these benchmarks operate on a similar principle. They consider the empirical errors

$$\epsilon_i = |\hat{y}_i - y_i|,$$

between the point forecasts \hat{y}_i and true values y_i on a validation data set. These empirical errors are then used to generate prediction intervals. The benchmarks differ in how these empirical errors are used to generate prediction intervals.⁸

The first considered benchmark is the *Gaussian Prediction Interval (Gaussian PI)*. In this case, the empirical errors are assumed to be distributed according to a Gaussian distribution and the prediction intervals are calculated based on the standard deviation of these errors [29].

Second, we consider the *Empirical Prediction Interval (Empirical PI)*. This benchmark does not assume any parametric distribution but instead uses the empirical distribution of these empirical errors to calculate the prediction intervals [56].

Finally, we consider a *Conformal Prediction Interval (Conformal PI)*. This benchmark, introduced for multi-horizon time series forecasts by [53], calculates a critical nonconformity score for each of the empirical errors and applies Bonferroni and finite sample correction to ensure temporal dependence between these critical scores across the forecast horizon. These critical nonconformity scores are combined with the point forecast to generate the prediction intervals [53].

4.4.2 Direct Probabilistic Forecasts

The second group of probabilistic benchmarks considers methods that directly generate probabilistic forecasts. The first of these benchmarks is *DeepAR* [48], which is an autoregressive recurrent neural network-based approach for probabilistic forecasting. We implement DeepAR using the PyTorch Forecasting library⁹.

⁸ These benchmarks are selected due to their simplicity and proven performance. Due to computational cost, we explicitly exclude machine learning methods that require retraining for each point forecast, such as [5] and [54].

⁹ https://pytorch-forecasting.readthedocs.io/en/stable/api/pytorch_forecasting.models.deepar.DeepAR.html

The second benchmark method is a *Quantile Regression Neural Network (QRNN)*. It trains a NN to directly forecast selected or multiple quantiles instead of the mean or median [35]. To realise the QRNN, we use a separate simple feed-forward NN to forecast each of the selected quantiles, training each NN with the appropriate pinball loss function. The QRNN is implemented using TensorFlow¹⁰ with the Keras¹¹ library and the pinball loss function.

The third benchmark method uses the *Nearest Neighbour Quantile Filter (NNQF)* proposed by [25]. Similar to the QRNN, this method also forecasts quantiles. However, instead of using a custom quantile loss function to directly learn the quantiles, the NNQF finds similar values for each time step based on similarity in the target variable to determine quantiles in the data. A forecasting method is then trained to predict these calculated quantiles [25]. To realise the NNQF, we use a multi-layer feed-forward NN with one output per quantile, which is implemented using sklearn [43] and pyWATTS [26].

4.5 Used cINN

In the evaluation, we use the same cINN architecture (see Table 18 in Appendix A) for each of the considered data sets. It is based on GLOW coupling layers that consider conditional input [34]. Similar to [1, 27], the conditional input is provided by a fully connected NN, which uses the same exogenous information available to the base forecaster as conditional information (see Table 15). We detail the implementation information for the used cINN in Tables 18, 19 and 20 in Appendix A. When training the used cINN, we apply the Adam optimiser with a maximum of 100 Epochs. Furthermore, when sampling in the latent space to generate probabilistic forecasts, we consider a sample size of 100. We implement the cINN in a pipeline with pyWATTS [26].

5 Evaluation

We evaluate our proposed approach in three steps. First, we compare the probabilistic forecasts generated from our approach when using different base point forecasters. Second, we compare our approach with existing probabilistic benchmarks. For each of these two steps, we first consider an overview of the normalised evaluation metrics for each model across all metrics and all data sets to establish an overview of the results. We then consider the probabilistic forecasts' quality, calibration, sharpness, and prediction intervals separately. Finally, in the third step, we recreate the price track of

¹⁰ <https://www.tensorflow.org/>

¹¹ <https://keras.io/>

GEFCom2014 and compare our approach to the competition winners.

5.1 Comparison of different base point forecasters

In this section, we compare the performance of the different base point forecasts combined with the cINN in our approach. An overview of the normalised evaluation metrics for each base point forecaster when combined with the cINN for all considered evaluation metrics and data sets is shown in Fig. 3.

For comparison purposes, the performance in each of the considered metrics is normalised so that the best-performing base point forecaster achieves a score of 0.1 and the worst-performing base point forecaster a score of 1. We observe that the best-performing base forecaster depends on the considered metric and the data set. However, the TFT performs consistently well according to all metrics across all data sets by ranking within the top three in all cases apart from the CRPS and nMPI(β) for both β on the Electricity data set. Furthermore, certain base point forecasters exhibit highly

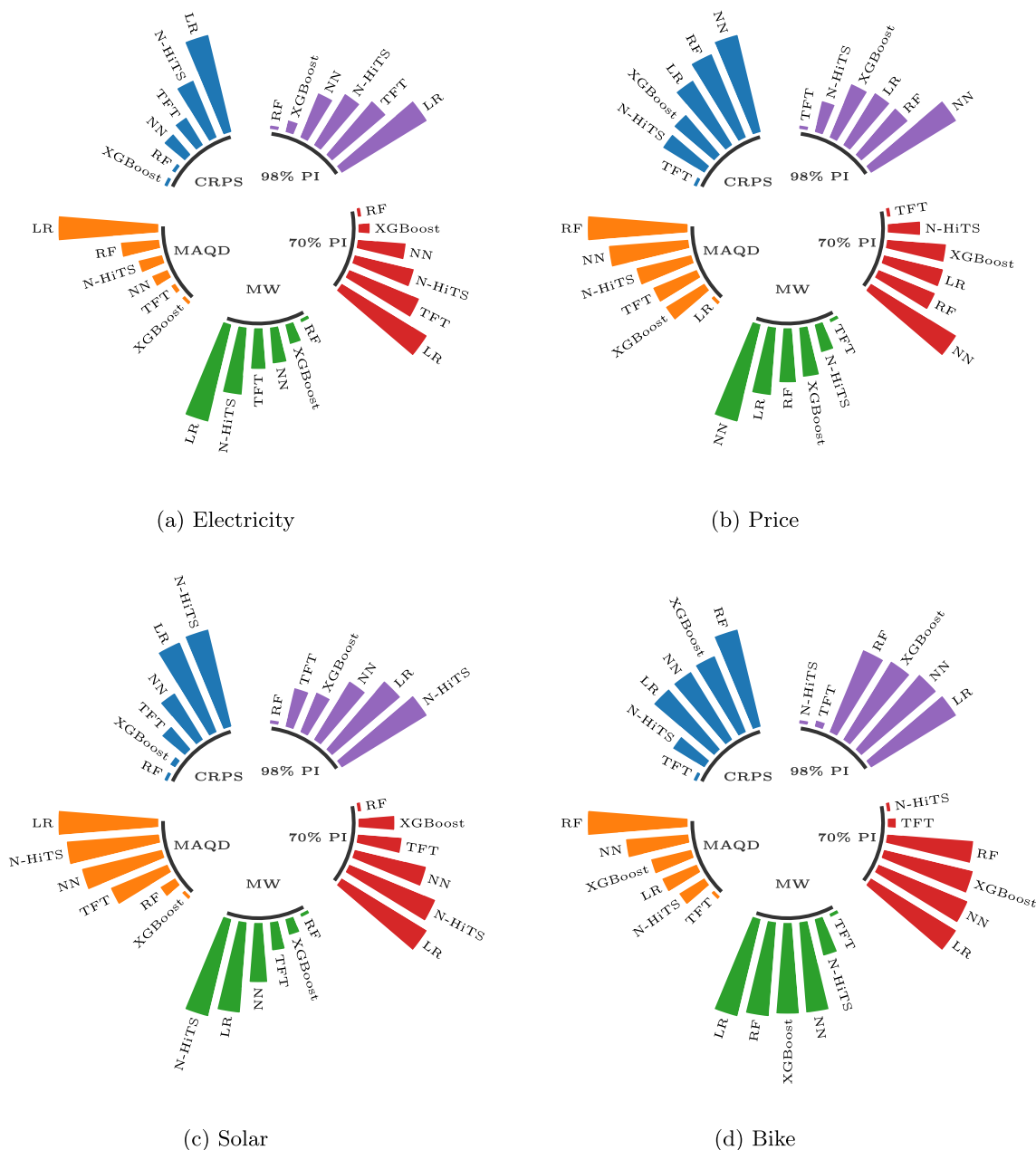


Fig. 3 An overview of the normalised evaluation metrics for each base point forecaster when combined with the cINN for each considered evaluation metric across all data sets. The value of each metric is normalised between 0.1 and 1 for illustrative purposes to facilitate the comparison, with lower values indicating better performance. The base point forecasters are ranked from best to worst for each metric

variable performance. For example, the LR achieves the best performance with regards to MAQD on the Price data set but consistently performs as one of the worst models on all other data sets. In the following, we report the results in more detail by comparing the forecast quality, the calibration, the sharpness, and the prediction intervals.

Quality

For each of the base point forecasters combined with the cINN, we report the average CRPS across five runs in CRPS Table 2.

We observe that the best-performing point forecaster combined with the cINN again depends on the data set considered, although the cINN base point forecaster combined with the cINN results in the lowest CRPS on two of the four data sets. Furthermore, although all base point forecasters combined with the cINN perform similarly on the Price data set, there are noticeable differences in the results of the other data sets. For example, the N-HiTS and TFT point forecasters combined with the cINN result in a noticeably lower CRPS on the Bike data set than the other point forecasters when combined with the cINN.

Calibration

To analyse the calibration of the probabilistic forecasts generated by combining different point forecasters with the cINN, we report the cINN in Table 3.

We observe that the base forecaster that results in the lowest MAQD when combined with the cINN depends on the data set considered. However, XGBoost, when combined with the cINN, achieves the lowest CRPS on two of the four data sets. Furthermore, the MAQD varies noticeably between the data sets. On the Electricity data set, almost all base point forecasters achieve a similar MAQD when combined with the cINN. However, on the Bike data set, only the TFT combined with the cINN result in a MAQD under 0.1.

Table 2 The average CRPS calculated on the test data set for each of the considered base point forecasters combined with the cINN over five runs. The best values for each data set are highlighted in bold

Forecasting Method	Data Set			
	Electricity	Price	Solar	Bike
LR-cINN	0.3180	0.1641	0.1686	0.4481
RF-cINN	0.2339	0.1689	0.1056	0.4992
NN-cINN	0.2542	0.1721	0.1399	0.4493
XGBoost-cINN	0.2337	0.1565	0.1072	0.4561
N-HiTS-cINN	0.2844	0.1552	0.1705	0.3454
TFT-cINN	0.2588	0.1404	0.1233	0.2641

Table 3 The average MAQD between the theoretical and forecast quantiles calculated on the test data set for each of the considered base point forecasters combined with the cINN over five runs. The best values for each data set are highlighted in bold

Forecasting Method	Data Set			
	Electricity	Price	Solar	Bike
LR-cINN	0.1360	0.0721	0.2079	0.1324
RF-cINN	0.1009	0.1246	0.0666	0.2023
NN-cINN	0.0886	0.1136	0.1780	0.1605
XGBoost-cINN	0.0811	0.0948	0.0431	0.1369
N-HiTS-cINN	0.0929	0.1010	0.1953	0.1236
TFT-cINN	0.0817	0.0959	0.1420	0.0945

Sharpness

We evaluate the sharpness of the probabilistic forecasts generated when combining different base learners with the cINN by reporting the average $n\text{MPI}(\beta)$ over five runs in Table 4.

Depending on the considered data set, we observe that different base point forecasters, when combined with the cINN, result in the best $n\text{MPI}(\beta)$. Only the RF, when combined with the cINN, achieves the best $n\text{MPI}(\beta)$ on more than one data set, whilst the TFT as base point forecaster performs best on the Price data set, and N-HiTS as a base forecaster on the Bike data set. We observe varying $n\text{MPI}(\beta)$ s across the data sets, with the largest $n\text{MPI}(\beta)$ of 1.2953 for $\beta = 98\%$ on the Electricity data set and the narrowest $n\text{MPI}(\beta)$ of 0.1329 for $\beta = 70\%$ on the Price data set.

Prediction intervals

To evaluate calibration and sharpness at the same time, we report the average MW score across five runs as a measure of the quality of the prediction intervals generated by different point forecasters in Table 5.

Again, the performance varies depending on the considered data set. Probabilistic forecasts generated when combining the TFT with the cINN result in the lowest MW score for the Price and Bike data sets, and RF as the base point forecaster results in the lowest MW score for the Electricity and Solar data sets. Regarding the MW scores, the performance varies noticeably depending on which base point forecaster is used on all data sets, although this variance is smaller on the Price data set.

5.2 Comparison to benchmarks

In the second step of our evaluation, we compare probabilistic benchmarks with the probabilistic forecasts generated when combining a cINN with the XGBoost, N-HiTS, and TFT base point forecasters. First, we compare the probabilistic

Table 4 The average $nMPI(\beta)$ calculated on the test data set for each of the considered base point forecasters combined with the cINN for $\beta = 98\%$ and $\beta = 70\%$ over five runs. The best values for each data set and β are highlighted in bold

Forecasting Method	Electricity		Price		Solar		Bike	
	98%	70%	98%	70%	98%	70%	98%	70%
LR-cINN	1.2953	0.5653	0.4618	0.2043	0.8989	0.4146	1.1744	0.5085
RF-cINN	0.7795	0.3640	0.4619	0.2046	0.6268	0.2964	1.1179	0.4815
NN-cINN	1.0163	0.4571	0.5785	0.2530	0.8319	0.3821	1.1661	0.5014
XGBoost-cINN	0.8248	0.3803	0.4518	0.2022	0.7510	0.3362	1.1335	0.4915
N-HITS-cINN	1.0811	0.4837	0.3741	0.1690	0.9315	0.4061	0.6906	0.3160
TFT-cINN	1.1376	0.5145	0.2893	0.1329	0.7408	0.3462	0.7051	0.3252

forecasts from our approach to benchmarks that also use these same point forecasters to generate probabilistic forecasts. Second, we compare our approach to methods that directly generate probabilistic forecasts.

5.2.1 Probabilistic forecasts based on existing point forecasts

We report an overview of the normalised evaluation metrics for the TFT as the base point forecaster when combined with the cINN and the other benchmarks based on existing point forecasts for all considered evaluation metrics and data sets in Fig. 4. For comparison purposes, the performance in each of the considered metrics is normalised so that the best-performing model achieves a score of 0.1 and the worst-performing model a score of 1. We first observe that for CRPS, MW, and $nMPI(\beta)$ with $\beta = 98\%$, our approach using the cINN results in the best performance on all data sets. Additionally, our approach also achieves the lowest $nMPI(\beta)$ for $\beta = 70\%$ on the Price and Bike data sets and only performs slightly worse than Conformal PI and Empirical PI on the remaining data sets. The benchmarks only perform better than our approach in terms of MAQD, with our approach never achieving the lowest MAQD. The results for the two remaining base point forecasters, namely XGBoost and N-HITS, are similar and can be found in Appendix C. In the remainder of this section, we analyse quality, calibration,

Table 5 The average MW score calculated on the test data set for each of the considered base point forecasters combined with the cINN over five runs. The best values for each data set are highlighted in bold

Forecasting Method	Data Set			
	Electricity	Price	Solar	Bike
LR-cINN	30.0248	14.6476	10.7391	36.8567
RF-cINN	17.2494	13.9050	6.7742	35.9929
NN-cINN	21.5586	16.4061	9.3110	35.4235
XGBoost-cINN	19.4470	13.6459	7.3523	35.4601
N-HITS-cINN	25.7098	12.5044	11.1579	27.2643
TFT-cINN	22.1819	11.1381	7.9020	22.0296

sharpness, and the prediction intervals for each considered data set in more detail.

Quality

We evaluate the quality of the different probabilistic forecasts by reporting the average CRPS across five runs in Table 6.

First, our approach using a cINN generally performs better or similarly to the benchmarks. The cINN results in probabilistic forecasts with the lowest CRPS for each considered point forecaster on the Price and Solar data sets and for two of the three point forecasters on the Electricity and Bike data sets. In the remaining two cases, the Conformal PI results in the lowest CRPS, however, the CRPS resulting from our approach is similar in all cases. Second, we observe that the Gaussian PI consistently results in the highest CRPS. Finally, we note that, across all data sets and for all considered point forecasters, the Empirical PI generates probabilistic forecasts resulting in almost identical CRPSs to those from the Conformal PI.

Calibration

To evaluate the calibration of the considered probabilistic forecasts, we report the average MAQD for each data set calculated across five runs in Table 7.

We first observe that the results depend strongly on the base point forecaster and the data set considered. Whilst the Conformal PI results in the lowest MAQD for all base point forecasters on the Electricity data set, the results for the other data sets are not as clear. On the Price and Solar data sets, the Conformal PI and Empirical PI achieve the lowest MAQD depending on the considered point forecaster, whilst each of the three considered benchmarks performs best on the Bike data set for one of the applied base point forecaster applied. Our approach using the cINN never achieves the lowest MAQD.

Sharpness

To assess the sharpness of probabilistic forecasts generated from point forecasts, we report the average $nMPI(\beta)$ over five runs in Table 8.

Our approach using a cINN results in the lowest $nMPI(\beta)$ for all base point forecasters, considered values of β and data

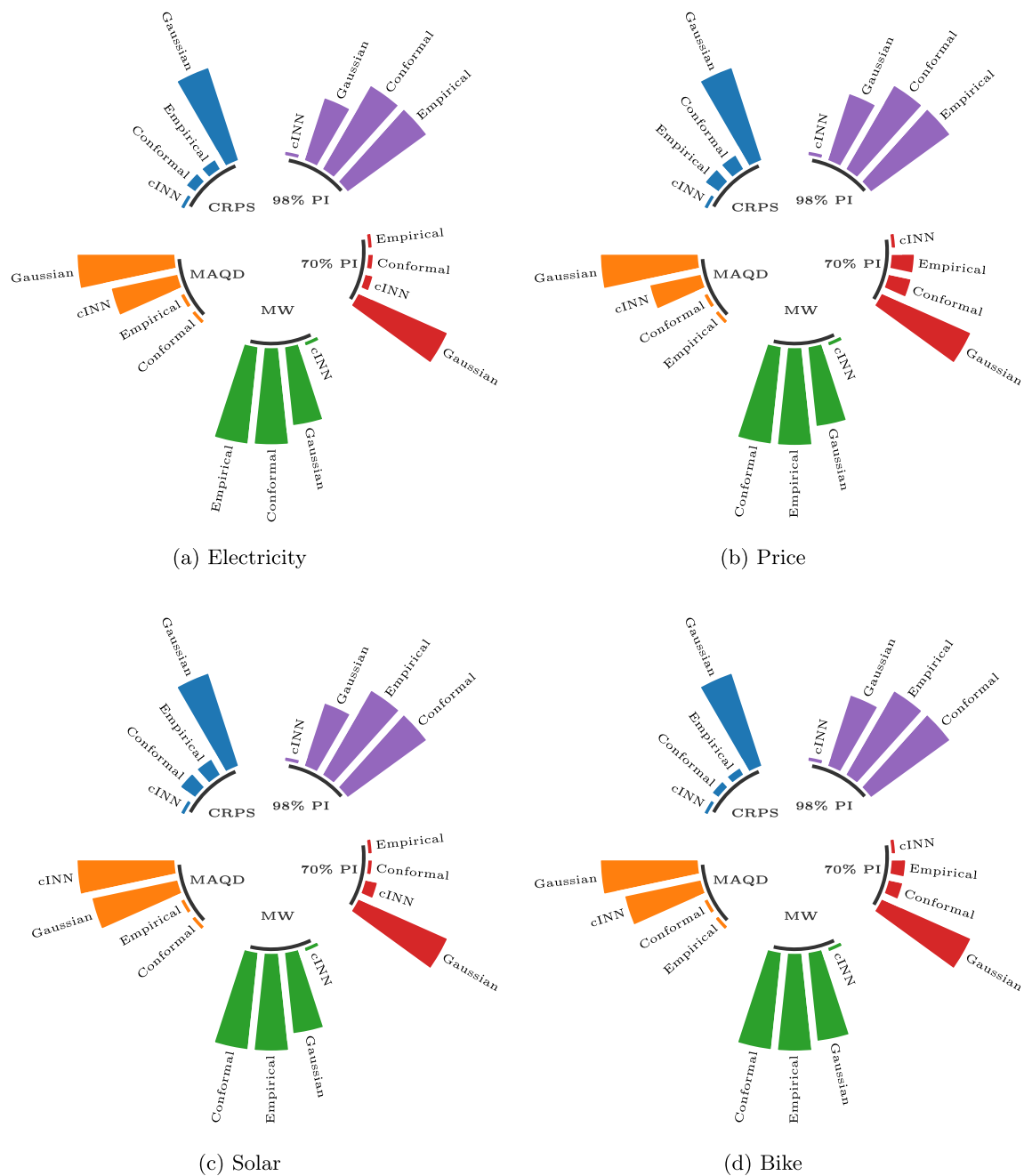


Fig. 4 An overview of the normalised evaluation metrics for the TFT base forecaster when combined with the cINN or used with the benchmarks based on existing point forecasts for each considered evaluation metric across all data sets. The value of each metric is normalised between 0.1 and 1 for illustrative purposes to facilitate the comparison, with lower values indicating better performance. The considered models are ranked from best to worst for each metric

sets almost all the time, with the exceptions being for $\beta = 70\%$ on the Electricity set with the TFT and on the Solar data set with XGBoost and the TFT. Moreover, the $nMPI(\beta)$ from the Empirical PI and Conformal PI are generally the largest for $\beta = 98\%$, and noticeably so. For example, the $nMPI(\beta)$ for $\beta = 98\%$ for Conformal PI on all data sets are often more than double the $nMPI(\beta)$ generated with the cINN. Further,

the $nMPI(\beta)$ for $\beta = 70\%$ are generally the largest with the Gaussian PI. Finally, the $nMPI(\beta)$ vary noticeably depending on the data set and selected point forecaster.

Prediction intervals

To simultaneously consider calibration and sharpness, we analyse the prediction intervals of the considered probabilis-

Table 6 A comparison of the average CRPS when generating probabilistic forecasts based on existing point forecasters. The average CRPS is calculated across five runs on the test data set, and the best values for each base point forecaster in each data set are highlighted in bold

Data	Point Forecaster	cINN	Gaussian PI	Empirical PI	Conformal PI
Electricity	XGBoost	0.2337	0.2993	0.2341	0.2339
	N-HiTS	0.2844	0.3630	0.2840	0.2835
	TFT	0.2588	0.3543	0.2658	0.2657
Price	XGBoost	0.1565	0.3496	0.1789	0.1786
	N-HiTS	0.1552	0.2785	0.1713	0.1712
	TFT	0.1404	0.2959	0.1607	0.1608
Solar	XGBoost	0.1072	0.2083	0.1249	0.1250
	N-HiTS	0.1705	0.2713	0.1781	0.1777
	TFT	0.1233	0.2333	0.1398	0.1398
Bike	XGBoost	0.4561	0.6075	0.4857	0.4856
	N-HiTS	0.3454	0.4232	0.3368	0.3363
	TFT	0.2641	0.3597	0.2680	0.2679

tic forecasts by comparing the average MW scores for each data set calculated over five runs in Table 9.

We first note that the probabilistic forecasts generated with the cINN result in the lowest MW scores on all data sets and for all considered point forecaster. Furthermore, the Winkler scores from our approach are noticeably smaller than the benchmarks. Although all the prediction interval-based benchmarks generate probabilistic forecasts with similar Winkler scores, the Gaussian PI results in slightly lower Winkler scores on all data sets, with the difference being most noticeable on the Price data set. Finally, we observe that similar to the CRPS results, the MW scores for the Empirical PI and Conformal PI are almost identical for every data set and each considered point forecaster, with the Conformal PI sometimes performing slightly better.

5.2.2 Direct probabilistic forecasts

An overview of the normalised evaluation metrics for our approach using a cINN combined with three base point forecasters (XGBoost, N-HiTS, TFT) and the three considered benchmarks that directly generate probabilistic forecasts for all considered evaluation metrics and data sets is shown in Fig. 5. For comparison purposes, the performance in each of the considered metrics is normalised so that the best-performing model achieves a score of 0.1 and the worst-performing model a score of 1. We observe that combining an appropriate base point forecaster with the cINN results in the best-performing model for all data sets and across all metrics in all but two cases. The exceptions are the MAQD on the Price data set, where the NNQF performs best, and

Table 7 Comparison of the average MAQD when generating probabilistic forecasts based on existing point forecasters. The average MAQD is calculated using five runs on the test data set, and the best values for each base point forecaster in each data set are highlighted in bold

Data	Point Forecaster	cINN	Gaussian PI	Empirical PI	Conformal PI
Electricity	XGBoost	0.0811	0.1029	0.0221	0.0220
	N-HiTS	0.0929	0.1034	0.0117	0.0112
	TFT	0.0817	0.1088	0.0241	0.0239
Price	XGBoost	0.0948	0.1565	0.0390	0.0387
	N-HiTS	0.1010	0.1467	0.0251	0.0245
	TFT	0.0959	0.1534	0.0349	0.0351
Solar	XGBoost	0.0431	0.1252	0.0124	0.0125
	N-HiTS	0.1953	0.1239	0.0186	0.0194
	TFT	0.1420	0.1290	0.0215	0.0215
Bike	XGBoost	0.1369	0.1169	0.1273	0.1271
	N-HiTS	0.1236	0.1099	0.0286	0.0281
	TFT	0.0945	0.1169	0.0110	0.0111

Table 8 Comparison of the average $nMPI(\beta)$ when generating probabilistic forecasts based on existing point forecasts for $\beta = 98\%$ and $\beta = 70\%$. The average $nMPI(\beta)$ is calculated using five runs on the test data set, and the best values for each base point forecaster and β in each data set are highlighted in bold

Data	Point Forecaster	PI	cINN	Gaussian PI	Empirical PI	Conformal PI
Electricity	XGBoost	98%	0.8248	1.5418	1.8594	1.8527
		70%	0.3803	0.9276	0.4510	0.4529
	N-HiTS	98%	1.0811	1.8780	2.1627	2.1579
		70%	0.4837	1.1278	0.6265	0.6265
	TFT	98%	1.1376	1.8769	2.2586	2.2469
		70%	0.5145	1.1288	0.4886	0.4965
Price	XGBoost	98%	0.4518	1.5271	1.9885	1.9193
		70%	0.2022	0.8810	0.2434	0.2444
	N-HiTS	98%	0.3741	1.1536	1.4098	1.3963
		70%	0.1690	0.6493	0.2352	0.2364
	TFT	98%	0.2893	1.3332	1.7665	1.7534
		70%	0.1329	0.7571	0.2589	0.2592
Solar	XGBoost	98%	0.7510	1.8661	2.4523	2.4564
		70%	0.3362	1.0729	0.2617	0.2626
	N-HiTS	98%	0.9315	2.3938	2.9698	2.9618
		70%	0.4061	1.3934	0.5539	0.5475
	TFT	98%	0.7408	2.0298	2.6828	2.6830
		70%	0.3462	1.1668	0.2651	0.2668
Bike	XGBoost	98%	1.1335	1.8288	1.9887	1.9885
		70%	0.4915	1.1201	0.6329	0.6342
	N-HiTS	98%	0.6906	1.3939	1.5342	1.5299
		70%	0.3160	0.8199	0.4735	0.4749
	TFT	98%	0.7051	1.2546	1.4288	1.4344
		70%	0.3252	0.7332	0.3695	0.3706

Table 9 Comparison of the average MW scores when generating probabilistic forecasts based on existing point forecasts. The average MW score is calculated across five runs on the test data set, and the best values for each base point forecaster in each data set are highlighted in bold

Data	Point Forecaster	cINN	Gaussian PI	Empirical PI	Conformal PI
Electricity	XGBoost	19.4470	35.5141	39.5519	39.4368
	N-HiTS	25.7098	44.5813	47.7180	47.5730
	TFT	22.1819	43.4597	48.7009	48.5111
Price	XGBoost	13.6459	47.6781	56.1564	54.8959
	N-HiTS	12.5044	35.9409	39.0527	39.0557
	TFT	11.1381	40.1158	46.8259	46.8474
Solar	XGBoost	7.3523	25.6594	29.7955	29.8464
	N-HiTS	11.1579	34.2878	38.5077	38.4250
	TFT	7.9020	28.6166	32.9194	32.9338
Bike	XGBoost	35.4601	70.4294	72.7355	72.7596
	N-HiTS	27.2643	51.1508	52.0275	51.8381
	TFT	22.0296	44.1040	46.3018	46.3026

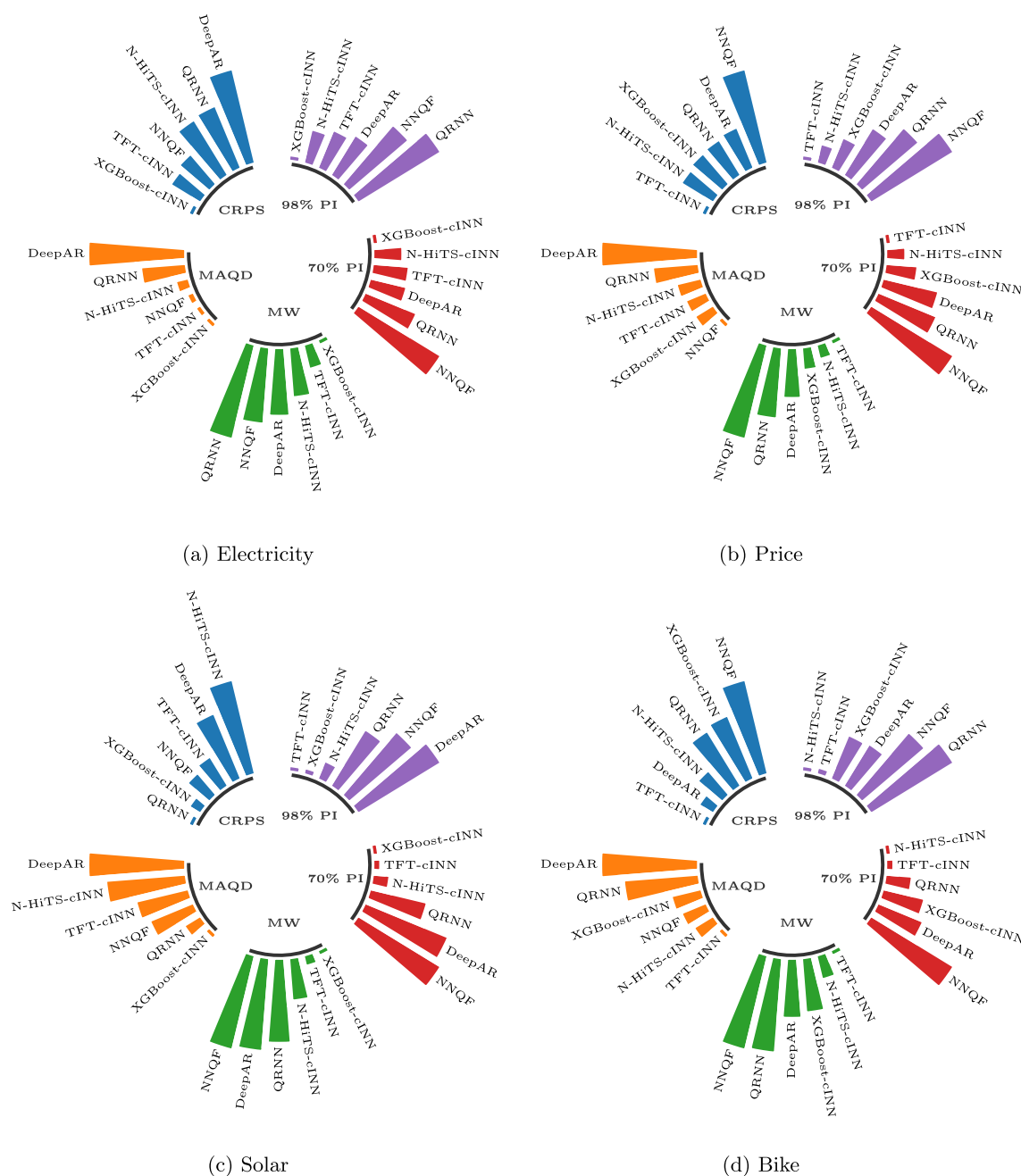


Fig. 5 An overview of the normalised evaluation metrics for our approach combining the cINN with three base point forecasters (XGBoost, N-HiTS, TFT) and the direct probabilistic benchmarks for each considered evaluation metric across all data sets. The value of each metric is normalised between 0.1 and 1 for illustrative purposes to facilitate the comparison, with lower values indicating better performance. The considered models are ranked from best to worst for each metric

the CRPS on the Solar data set, where the QRNN provides the best performance. Furthermore, regarding MW, all three of the base point forecasters, when combined with the cINN, perform better than all of the direct benchmarks on all data sets. Finally, our approach performs consistently across all metrics and data sets, only once achieving the worst ranking, whilst the performance of the direct benchmarks is far more variable, with each of them being ranked worst at least

three times. To evaluate the performance in more detail, we consider the forecast quality, calibration, sharpness and prediction intervals for each of the direct benchmarks and our approach with the cINN on the four considered data sets in the following.

Quality

To analyse the quality of the probabilistic forecasts, we report the average CRPS across five runs for all data sets in Table 10.

Table 10 Comparison of the average CRPS between the probabilistic forecasts from the cINN and the direct probabilistic benchmarks. The average CRPS is calculated over five runs on the test data set, and the best values for each data set are highlighted in bold

Forecasting Method	Data Set			
	Electricity	Price	Solar	Bike
XGBoost-cINN	0.2337	0.1565	0.1072	0.4561
N-HiTS-cINN	0.2844	0.1552	0.1705	0.3454
TFT-cINN	0.2588	0.1404	0.1233	0.2641
DeepAR	0.3115	0.1583	0.1509	0.2985
QRNN	0.2866	0.1571	0.1013	0.4431
NNQF	0.2629	0.1825	0.1191	0.5415

The first observation is that our approach results in the lowest CRPS on three of the four data sets. Thereby, the choice of the base point forecaster is important, with XGBoost combined with the cINN performing best on the Electricity data set, whilst the TFT combined with the cINN performs best on the Price and Bike data sets. On the Solar data set, the QRNN benchmark outperforms all others, although our approach using the XGBoost as a base point forecaster performs similarly. Furthermore, when combined with the cINN, we observe that all base point forecasters outperform all the direct benchmarks on the Price data set, and two of the three base point forecasters outperform all the direct benchmarks on the Electricity data set. In general, the performance of the direct benchmarks is also highly dependent on the considered data set. Of the direct benchmarks, the NNQF performs best for the Electricity data set, the QRNN for the Price and Solar data sets, and DeepAR for the Bike data set.

Calibration

To assess the calibration of the direct probabilistic benchmarks and our approach, we report the average MAQD across five runs in Table 11.

Similar to the CRPS results, our approach using a cINN results in the lowest deviation for three of the four data sets. However, unlike the CRPS results, our approach using the cINN results in the lowest MAQD on the Electricity, Solar, and Bike data sets whilst the NNQF achieves the lowest overall MAQD, on the Price data set. With regards to the direct benchmarks, the NNQF outperforms the other direct benchmarks on all data sets except for the Solar data set, where the lowest MAQD is achieved with the QRNN.

Sharpness

To compare the sharpness of probabilistic forecasts generated with our approach and those from the direct benchmarks, we report the average nMPI(β) over five runs in Table 12.

With regards to the nMPI(β), our approach results in the smallest nMPI(β) for all data sets. Using the XGBoost as

a base point forecaster generates the narrowest prediction intervals for the Electricity data set and the lowest nMPI(β) for $\beta = 70\%$ for the Solar data set. Additionally, using the TFT as a base point forecaster results in the narrowest prediction intervals for the Price data set and the lowest nMPI(β) for $\beta = 98\%$ for the Solar data set. Finally, using N-HiTS as a base point forecaster results in the lowest nMPI(β) for the Bike data set. The width of the prediction intervals for the direct probabilistic benchmark depends on the data set. For the Electricity and Price data sets, DeepAR generates probabilistic forecasts with the lowest nMPI(β). However, for the Solar data set, the nMPI(β) from the QRNN is the smallest. The Bike data set is interesting for the benchmarks since the nMPI(β) with $\beta = 98\%$ is the smallest for DeepAR, but the nMPI(β) with $\beta = 70\%$ is the smallest for the QRNN.

Prediction intervals

To evaluate calibration and sharpness simultaneously, we consider the quality of the prediction intervals generated with our approach and the direct probabilistic benchmarks. For this purpose, we report the average MW score across five runs in Table 13.

We first observe that our approach results in the lowest MW scores for every data set. Furthermore, the MW scores for each point forecaster, when combined with the cINN, are lower than any of the direct benchmarks on all data sets. Regarding the direct probabilistic benchmarks, the best-performing benchmark depends on the data set considered. DeepAR results in the lowest MW scores for the Electricity, Price, and Bike data sets, whilst QRNN results in the lowest MW scores for the Solar data set.

5.2.3 Qualitative analysis

As a final comparison to the benchmarks, we qualitatively compare prediction intervals and calibration for the Price data set to gain further insight into the characteristics of probabilistic forecasts generated by our approach and the

Table 11 Comparison of the average MAQD between the theoretical and forecast quantiles from the cINN and the direct probabilistic benchmarks. The average MAQD is calculated across five runs on the test data set, and the best values for each data set are highlighted in bold

Forecasting Method	Data Set			
	Electricity	Price	Solar	Bike
cINN-XGBoost	0.0811	0.0948	0.0431	0.1369
cINN-N-HiTS	0.0929	0.1010	0.1953	0.1236
cINN-TFT	0.0817	0.0959	0.1420	0.0945
DeepAR	0.2222	0.2139	0.2301	0.2478
QRNN	0.1420	0.1332	0.0708	0.2110
NNQF	0.0835	0.0692	0.1282	0.1303

Table 12 Comparison of the average $nMPI(\beta)$ between forecasts from the cINN and the direct probabilistic benchmarks for $\beta = 98\%$ and $\beta = 70\%$. The $nMPI(\beta)$ is calculated across five runs on the test data set, and the best values for each data set and β are highlighted in bold

Forecasting Method	Electricity		Price		Solar		Bike	
	98%	70%	98%	70%	98%	70%	98%	70%
XGBoost-cINN	0.8248	0.3803	0.4518	0.2022	0.7510	0.3362	1.1335	0.4915
N-HiTS-cINN	1.0811	0.4837	0.3741	0.1690	0.9315	0.4061	0.6906	0.3160
TFT-cINN	1.1376	0.5145	0.2893	0.1329	0.7408	0.3462	0.7051	0.3252
DeepAR	1.1896	0.5204	0.5850	0.2672	1.9875	0.8517	1.1524	0.5204
QRNN	1.6227	0.5990	0.6956	0.2840	1.5715	0.6610	1.6535	0.4150
NNQF	1.4587	0.7790	0.8259	0.3708	1.8035	0.8962	1.5027	0.7468

considered benchmarks. In this analysis, we only consider the Conformal PI from the first group of benchmarks since this method performs overall best compared to the other benchmarks in that group.

We plot the 98%, 70%, and 40% prediction intervals for a single day in the test data set in Fig. 6. Compared to the Conformal PI, our approach generates probabilistic forecasts with the narrowest prediction intervals regardless of the base point forecaster used. In fact, for probabilistic forecasts generated with the N-HiTS or TFT base point forecaster, our approach using a cINN results in the narrowest prediction intervals overall. Furthermore, whilst the 40% and 70% Conformal PIs are only slightly wider than those generated by the cINN, the 98% prediction intervals are by far the widest of all considered benchmarks. The three direct probabilistic benchmarks generate prediction intervals that are generally wider than those generated by the cINN but narrower than the Conformal PIs.

To further analyse the calibration of our forecasts, we plot the forecast quantile coverage against the theoretical quantile coverage as a calibration plot in Fig. 7. We observe that, for all base point forecasters, the Conformal PI provides the most calibrated forecasts, with hardly any deviation

Table 13 Comparison of the average MW score between the probabilistic forecasts from the cINN and the direct probabilistic benchmarks. The average MW score is calculated over five runs on the test data set, and the best values for each data set are highlighted in bold

Forecasting Method	Data Set			
	Electricity	Price	Solar	Bike
XGBoost-cINN	19.4470	13.6459	7.3523	35.4601
N-HiTS-cINN	25.7098	12.5044	11.1579	27.2643
TFT-cINN	22.1819	11.1381	7.9020	22.0296
DeepAR	28.2175	17.6831	16.3384	36.7610
QRNN	32.2938	20.7876	15.4228	46.4754
NNQF	29.4266	24.5482	16.7298	47.0819

from the diagonal. However, our approach using a cINN also results in forecasts that only slightly deviate from the diagonal by slightly overestimating the lower quantiles and slightly underestimating the upper quantiles. From the direct probabilistic benchmarks, the NNQF achieves similar results to our approach using a cINN, whilst the results of DeepAR and the QRNN are noticeably worse.

5.3 GEFCom2014 probabilistic price forecasting

For the final step of our evaluation, we test the proposed approach by retrospectively determining its placement in the GEFCom2014. The GEFCom2014 was a probabilistic energy forecasting competition with four different tracks for load, price, wind, and solar forecasting [28]. For the evaluation, we recreate the setup of the GEFCom2014 price forecasting track in which 14 teams competed and compare the performance of our approach to the leading entrants from the competition. This comparison is based on the scoring mechanism from GEFCom2014 that considers the final twelve of fifteen tasks, each generating 24-hour quantile forecasts. Given the pinball loss improvement $PL_{\%}$ for each task, the final ranking is determined as the average pinball loss improvement across all tasks weighted by the task number [28]. For our approach, we apply all previous base point forecasters, select the best-performing sampling parameter over the first three non-evaluated tasks, and use this for all remaining tasks. The final weighted pinball loss improvement and the resulting rank of our approach are shown in Table 14 (see Table 22 for the results per task).

Overall, our approach with a cINN and various base forecasters performs well. With simple point forecasting methods such as RF regression and LR, we achieve an average weighted pinball loss improvement that would have placed these methods within the top five of the competition. Furthermore, with a more advanced TFT base forecaster, we achieve an average weighted pinball loss improvement that would have resulted in a third-place finish.

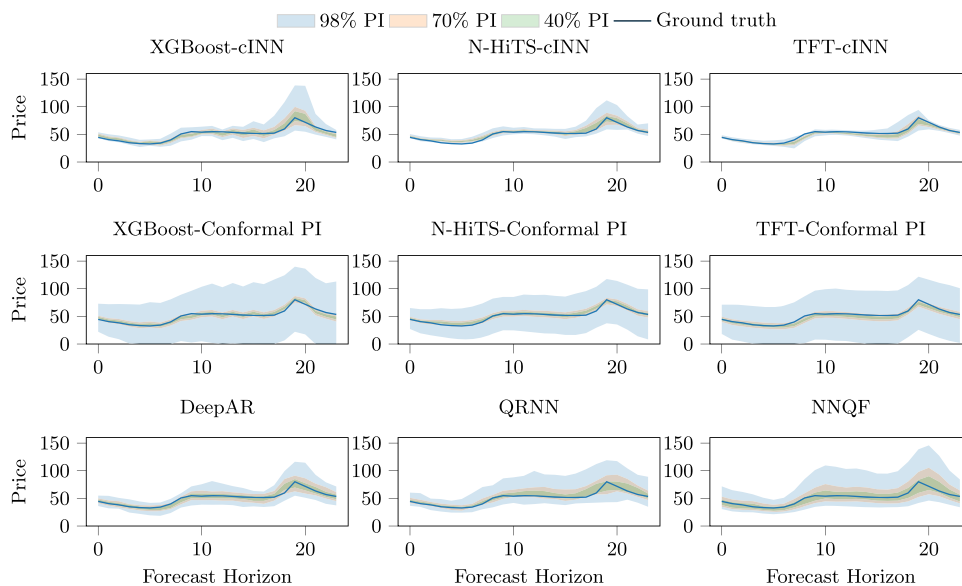


Fig. 6 Exemplary 98%, 70%, and 40% prediction intervals for the Price data set. Probabilistic forecasts are generated by using XGBoost, N-HiTS, and the TFT as base point forecasters and either combining them with our cINN or applying Conformal PI. Further, we compare the three direct probabilistic benchmarks: DeepAR, QRNN, and NNQF

6 Discussion

In this section, we first discuss our results and the implications these have before we highlight some of the key insights gained from the evaluation.

6.1 Results

With regard to our results, we discuss two aspects. First, we focus on the forecasting performance of our approach before considering the GEFCom2014.

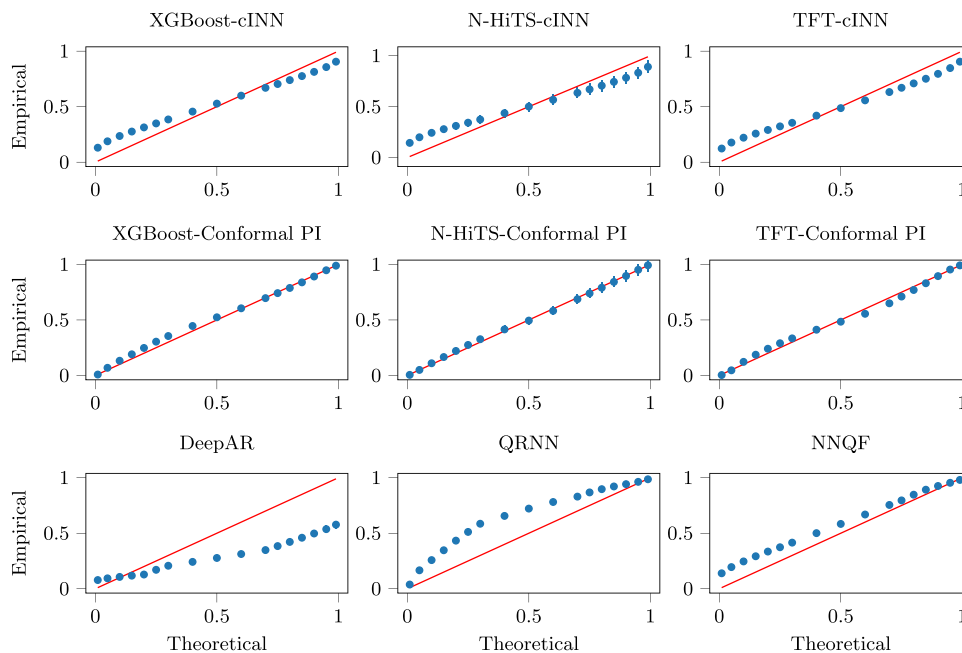


Fig. 7 Exemplary calibration plots comparing the theoretical and forecast quantiles on the Price data set, with the red diagonal indicating zero deviation. We compare probabilistic forecasts generated by using XGBoost, N-HiTS, and the TFT as base point forecasters and either combining them with our cINN or applying Conformal PI. Further, we compare the three direct probabilistic benchmarks: DeepAR, QRNN, and NNQF

Table 14 The overall weighted pinball loss improvement all tasks and final rank in the GEFCom2014 price forecasting challenge for all base forecasters from Section 4.3 combined with the cINN. The weighted pinball loss improvement indicates the improvement of a given method over the GEFCom2014 baseline forecast weighted by the task number. The rank indicates the placement in GEFCom2014 according to the weighted pinball loss improvement

	Average weighted pinball loss improvement	Rank ¹
RF-cINN	65.9	4
LR-cINN	65.7	4
NN-cINN	61.5	9
XGBoost-cINN	65.0	4
N-HiTS-cINN	62.9	6
TFT-cINN	67.4	3
Tololo [20]	71.7	1
Team Poland [40]	67.7	2
GMD [14]	67.1	3

¹This ranking is determined by how each individual method would have placed in the GEFCom2014 price forecasting challenge in 2014 and, therefore, assumes that only one of the presented methods is considered at a time when creating the respective ranking

Forecasting performance

With regard to forecasting performance, we first discuss the performance of our approach with different point forecasters before comparing our approach to other probabilistic benchmarks.

When comparing different point forecasters in our approach, we note that the quality of the point forecasts affects the quality of the probabilistic forecasts when combined with the cINN. This observation is unsurprising since the cINN in our approach includes uncertainty around the initial point forecast, and, therefore, the more accurate the point forecast is, the easier it is to include uncertainty effectively. Additionally, the quality of the point forecasts is influenced by the exogenous features considered. Therefore, it may be useful to consider additional features or factors influencing the forecast, similar to [58].

When comparing our approach to the selected benchmarks, we make several observations. First, we observe that our approach generally outperforms all benchmarks regarding CRPS. In the three occasions where our approach does not result in the lowest CRPS, the difference between the best performing Conformal PI or QRNN and our approach is small.

Second, our approach is not optimally calibrated. Although the forecasts generated with our cINN are well calibrated compared to the direct probabilistic benchmarks, the Empirical PI and Conformal PI achieve lower MAQDs on all data sets. However, it is worth noting that prediction interval-based approaches are specifically designed to achieve certain coverage levels, and further, considering the calibration plots in Fig. 7 suggests that the difference in calibration may not be as noticeable as the raw MAQD numbers suggest.

Third, our approach consistently generates the sharpest probabilistic forecasts with the lowest nMPI(β). This obser-

vation is further highlighted by Fig. 6 where the forecasts from the N-HiTS and TFT base point forecaster combined with the cINN are far narrower than those of any other benchmarks.

Fourth, our approach outperforms all considered benchmarks on all data sets with regards to MW scores. Since Winkler scores consider both calibration and sharpness, this result suggests that the poorer calibration from our approach is counteracted by the narrow prediction intervals. Considering again Fig. 6, although the prediction intervals of our approach are narrow, the ground truth is still almost always contained within the interval. In comparison, the other benchmark methods, specifically the prediction interval-based approaches, appear to overestimate the width of the prediction intervals, which adversely affects the Winkler score.

Fifth, we note that both our approach and each of the considered benchmarks have strengths and weaknesses. Our approach results in narrow prediction intervals and low CRPS scores, but this comes at the cost of calibration performance. In contrast, the prediction interval-based benchmarks are highly calibrated but generate far wider prediction intervals, resulting in a worse performance regarding Winkler scores. Therefore, the best probabilistic forecast may vary, depending on the requirements of the situation it is being used for.

Finally, when considering the performance of our approach across all metrics compared to the benchmarks (see Figs. 4 and 5), we conclude that the resulting probabilistic forecasts are high quality and generally outperform all benchmarks. Our approach's small loss in calibration performance results in a far sharper forecast, which is reflected in the generally better performance in CRPS and MW. Therefore, our approach can be considered to deliver state-of-the-art probabilistic forecasts.

GEFCom2014

Not only does our approach perform competitively in the considered track of the GEFCom2014, but several factors undercut its true performance. All top-placing contestants in the competition perform specialised operations to improve forecasting performance, i.e. peak pre-processing [20], filtering methods to weight certain days higher [40], or tailored training data periods to improve performance [14]. In contrast, we consider all available data for training, refrain from complex pre-processing steps, and only use the default hyperparameters for the base forecasters. Furthermore, the true value of our approach is its ability to enable simple base point forecasters, such as a LR or RF, to rank within the top five compared to the original entrants. Finally, different base point forecasters perform better for different tasks. Therefore, we expect an ensemble method that automatically selects the best base forecaster for a given task to deliver even better performance.

6.2 Insights

In addition to the results, there are a few insights regarding the sampling in the latent space and the flexible nature of our approach, which we discuss here.

Sampling in latent space

Our approach includes uncertainty in point forecasts via latent space distribution sampling. Currently, this sampling is performed by adding normally distributed random noise $\mathbf{r}_i \sim \mathcal{N}(0, \sigma)$ to the point forecast. This approach has several limitations. First, the sampling parameter σ is manually selected to generate optimal forecasts according to CRPS. However, by varying this sampling parameter, it is possible to generate different probabilistic forecasts which follow the same general shape but vary in the amount of uncertainty considered. Therefore, it may be interesting to investigate methods to automatically select an optimal sampling parameter given the observed data, a selected base forecaster, and a specific evaluation metric. Such methods would enable the generation of probabilistic forecasts with properties that are specifically designed for a certain application. However, such probabilistic forecasts will only be possible if the requirements of this application can be formulated via a probabilistic loss metric that can be used to select σ .

Second, the current approach to optimise σ is rather rudimentary and based on a single evaluation metric. Therefore, it would be interesting to adapt this optimisation, perhaps by adapting concepts from conformal prediction to calculate the nonconformity scores of the samples. Furthermore, optimising the samples based on the resulting quantiles used as an output of the probabilistic forecast might be interesting.

With such a strategy, Bonferroni correction could possibly be applied to improve the calibration of our approach.

Flexible nature

In the present article, we evaluate the probabilistic forecasting performance of a single selected base point forecaster combined with the cINN. However, our approach is independent of the base point forecast considered, i.e. once the cINN has been trained for a given data set, we can generate probabilistic forecasts from any arbitrary point forecast without retraining. This is advantageous compared to other methods using cINNs or GANs, which require the generative model to be retrained whenever the point forecast is altered. Moreover, such an approach allows us to easily generate an ensemble of probabilistic forecasts based on different point forecasts.

Furthermore, for similar data sets, it may be possible to generate probabilistic forecasts with a generalised cINN that is only trained once on all data sets or a subset thereof. Such a generalised cINN could be beneficial for global forecasting and be applied to scenarios where no data is available, e.g. a new building similar to existing buildings is included in a suburb. However, such a generalised cINN will only be possible if the underlying distribution across the multiple data sets is similar and can be accurately mapped to a single tractable latent space distribution.

Another important aspect is that our approach is not limited to prediction intervals or specific quantiles. Whilst we choose to calculate quantiles based on samples as the output of our approach, the generative nature of the cINN enables us to generate an arbitrary number of samples and either use these directly to form an ensemble forecast or to output an empirical forecast distribution. This is advantageous compared to other probabilistic forecast methods that are, by nature, limited to generating prediction intervals. Specifically, applying an empirical density estimation algorithm to obtain a non-parametric density forecast is simple based on our approach. Such forecasts may be particularly useful if the entire distribution is required, for example, for a stochastic optimisation problem.

7 Conclusion

In the present article, we introduce an approach to generate probabilistic forecasts from arbitrary point forecasts by using a Conditional Invertible Neural Network (cINN) to learn the underlying distribution of the time series data. Our approach maps the underlying distribution of the data to a known and tractable distribution before combining the uncertainty from this known and tractable distribution with an arbitrary point

forecast to generate probabilistic forecasts. Importantly, the cINN is independent of the considered point forecast and must not be retrained when the point forecast is altered.

We evaluate our approach by combining multiple point forecasts with a cINN and comparing the resulting probabilistic forecasts with six probabilistic benchmarks on four data sets. We show that our approach generally outperforms all benchmarks regarding CRPS and Winkler scores. Further, our approach generates probabilistic forecasts with the narrowest prediction intervals whilst maintaining reasonable performance in calibration. Finally, we recreate the GEF-Com2014 and show that our approach enables simple base point forecasters to rank within the top five.

Our approach offers a solution to generate flexible probabilistic forecasts based on arbitrary point forecasts. In future work, this flexibility should be further investigated by developing a more advanced strategy for selecting the sampling parameter to improve the calibration of our probabilistic forecasts. Furthermore, automating the selection of this sampling parameter and considering how different

metrics for optimising this parameter affect the resulting forecasts should be investigated. It would also be interesting to extend our approach to multivariate probabilistic forecasts. Finally, it may be interesting to explore the performance of our approach using a generalised cINN to generate probabilistic forecasts on multiple data sets without retraining.

Appendix A additional implementation details

This Appendix contains the following additional implementation details:

- An overview of the data sets in Table 15.
- An overview of the selected base forecasters in Table 16.
- The selected sampling hyperparameter σ for each base point forecaster and each data set in Table 17.
- Implementation details for the applied cINN in Tables 18 to 19, 20.

Table 15 Overview of the data sets used including the exogenous features considered, and the used train, validation, and test sets

	Target	Exogenous Features	Train	Validation	Test
Electricity	MT_158	Calendar Information ¹	[0, 14716]	[14717, 21023]	[21024, 26280]
Price	Zonal Price	Calendar Information ¹ , Forecast Total Load, Forecast Zonal Load	[0, 14541]	[14542, 20773]	[20774, 25968]
Solar	POWER	Calendar Information ¹ SSRD ² , TCC ³	[0, 11033]	[11034, 15762]	[15763, 19704]
Bike ⁴	cnt	Calendar Information ¹ , Temperature, Humidity, Windspeed, Weather Situation	[0, 9824]	[9825, 14034]	[14035, 17544]

¹Sine- and cosine-encoded time of the day, the sine- and cosine-encoded month of the year, and a Boolean that indicates whether the current day is a weekend day or not

²Surface solar radiation downwards

³Total cloud cover

⁴To create a time index for this data, we merge the columns *dteday* and *hr* and deal with missing values using linear interpolation

Table 16 Overview of the selected base forecasters used to generate point forecasts

Base Forecaster	Classification	Implementation Details	Library Used
LR	Statistical	Default Hyperparameters	SKlearn ¹
RF	Statistical	Default Hyperparameters	SKlearn ¹
NN	Machine Learning	Hidden Layers: 3 Layer Sizes: 90-64-32 Hidden Activation Function: relu Output Activation Function: linear Optimiser: Adam ² Batch Size: 100 Max Epochs: 100	Tensorflow ³ Keras ⁴
XGBoost	Gradient Boosting	Default Hyperparameters	XGBoost ⁵
N-HiTS	Deep Learning	Default Hyperparameters	PyTorch Forecasting ⁶
TFT	Deep Learning	Default Hyperparameters	PyTorch Forecasting ⁶

¹ [43]

² [33]

³ <https://www.tensorflow.org/>

⁴ <https://keras.io/>

⁵ [7]

⁶ <https://pytorch-forecasting.readthedocs.io/en/stable/index.html>

Table 17 The selected sampling parameter for each base point forecaster and each data set used in the evaluation

Data Set	LR	RF	NN	XGBoost	N-HiTS	TFT
Electricity	0.57	0.63	0.49	0.36	0.59	0.72
Price	0.73	0.98	0.92	0.48	0.69	0.76
Solar	0.14	0.77	0.21	0.45	0.22	0.44
Bike	0.54	0.97	0.57	0.46	0.33	0.36
GEFCom Competition	1.05	0.95	0.90	0.50	0.75	1.05

Table 18 The architecture of the used cINN

Parameter	Description
Layers per block	Glow coupling layer and random permutation
Subnetwork in block	Fully connected (see Table 19)
Number of blocks	5
Conditioning network	Fully connected (see Table 20)

Table 19 Implementation details of the subnetwork in the used cINN

Layer	Description
Input	[Output of previous coupling layer, conditional information]
1	Dense 32 neurons; activation: tanh
2	Dense 24 neurons; activation: linear

Table 20 Implementation details of the conditioning network in the used cINN

Layer	Description
Input	[Calendar information, historical information, exogenous forecasts if available]
1	Dense 8 neurons; activation: tanh
2	Dense 4 neurons; activation: linear

Table 21 Comparison of the average RMSE on the test data set for each of the considered base point forecasters. The best values for each data set are highlighted in bold

Base Forecaster	Data Set	Price	Solar	Bike
cline2-5	Electricity			
LR	0.5246	0.4118	0.3331	0.8565
RF	0.4601	0.4253	0.2891	0.9913
NN	0.4894	0.4499	0.3257	0.9227
XGBoost	0.4532	0.4090	0.2966	0.9258
N-HiTS	0.5329	0.4124	0.3686	0.6471
TFT	0.5134	0.3672	0.3366	0.5457

forecasters. To evaluate the quality of the base point forecasters, we consider the Root Mean Squared Error (RMSE). The RMSE is given by

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (\text{B1})$$

with a true value y_i , a forecast value \hat{y}_i , and n observations.

We report the average RMSE (B1) over five runs in Table 21.

In general, we observe that the best-performing point forecast depends on the data set considered, with the performance of most point forecasters varying noticeably across the data sets. However, the TFT performs most consistently, achieving the lowest RMSE on two of the four data sets.

Appendix B point forecast evaluation

Since the quality of the base point forecaster influences the resulting probabilistic forecast when combined with a cINN, we briefly evaluate the point performance of the selected base

Appendix C additional result summaries

For completeness, we report the normalised evaluation metrics overview for the eXtreme Gradient Boosting (XGBoost)

in Fig. 8 and Neural Hierarchical Interpolation for Time Series Forecasting (N-HITS) in Fig. 9.

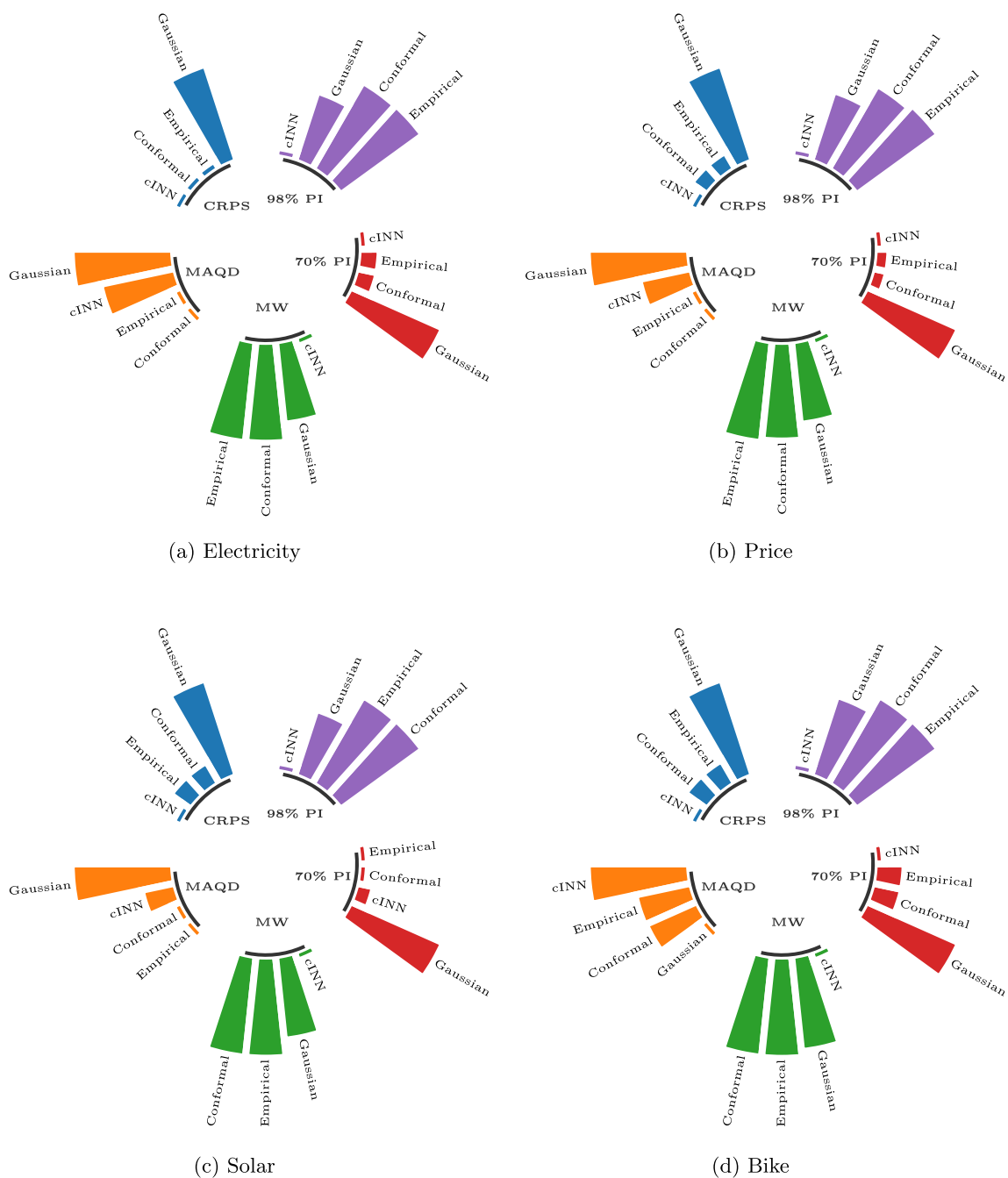


Fig. 8 An overview of the normalised evaluation metrics for the XGBoost base forecaster when combined with the cINN or used with the benchmarks based on existing point forecasts for each considered evaluation metric across all data sets. The value of each metric is normalised between 0.1 and 1 for illustrative purposes to facilitate the comparison, with lower values indicating better performance. The considered models are ranked from best to worst for each metric

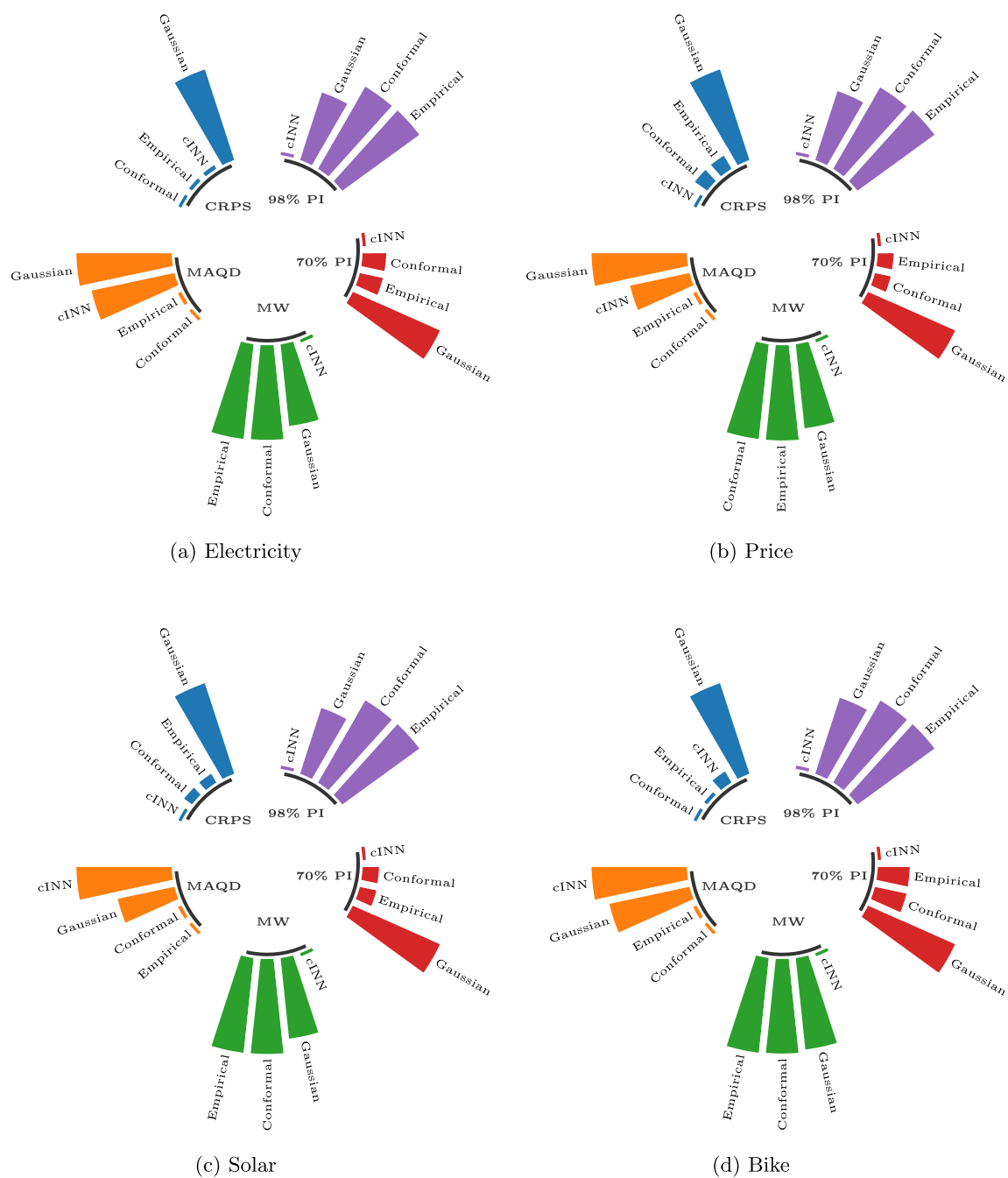


Fig. 9 An overview of the normalised evaluation metrics for N-HiTS base forecaster when combined with the cINN or used with the benchmarks based on existing point forecasts for each considered evaluation metric across all data sets. The value of each metric is normalised between 0.1 and 1 for illustrative purposes to facilitate the comparison, with lower values indicating better performance. The considered models are ranked from best to worst for each metric

Appendix D Full GEFCom2014 Results

Table 22 GEFCom2014 results for each task. The performance of each base forecaster from Section 4.3 combined with the cINN is reported, as well as the performance of the best three entrants from 2014. For each task, the pinball loss improvement $PL\%$ compared to the GEFCom2014 baseline is calculated. The final results are a linear weighted average of the pinball loss improvement in each task, weighted by the task number

Model	Task 1		Task 2		Task 3		Task 4		Task 5		Task 6			
	PL	$PL\%$	PL	$PL\%$	PL	$PL\%$	PL	$PL\%$	PL	$PL\%$	PL	$PL\%$		
RF-cINN	1.10	72.67	3.46	56.61	2.93	48.64	7.13	41.35	5.24	86.34	6.45	85.41		
LR-cINN	1.76	56.25	2.95	62.97	1.92	66.36	2.97	75.53	8.59	77.60	12.14	72.55		
NN-cINN	1.16	71.14	4.38	45.01	2.02	64.57	4.31	64.53	7.26	81.05	7.10	83.96		
XGBoost-cINN	1.70	57.69	3.53	55.76	2.74	51.90	5.21	57.12	6.08	84.14	7.65	82.71		
N-HiTS-cINN	1.70	57.87	2.45	69.21	1.05	81.57	3.75	69.15	8.03	79.06	10.08	77.21		
TFT-cINN	1.22	69.80	2.50	68.64	1.16	79.67	3.08	74.65	6.30	83.57	9.20	79.21		
Tololo ²	1.71	57.62	1.45	81.80	1.10	80.65	2.02	83.40	9.16	76.11	4.68	89.41		
Team Poland ³	1.97	50.98	1.82	77.19	1.19	79.11	2.82	76.77	7.56	80.28	4.21	90.49		
GMD ⁴	3.73	7.48	1.78	77.63	0.92	83.84	5.09	58.12	6.21	83.79	3.83	91.35		
	Task 7		Task 8		Task 9		Task 10		Task 11		Task 12		Overall	
	PL	$PL\%$	PL	$PL\%$	PL	$PL\%$	PL	$PL\%$	PL	$PL\%$	PL	$PL\%$	Total Weighted $PL\%$	Rank ¹
RF-cINN	7.55	58.54	2.51	92.06	2.16	94.97	1.95	31.89	1.79	44.03	5.69	74.56	65.9	4
LR-cINN	12.66	30.53	1.85	94.14	2.39	94.44	1.50	47.36	1.66	48.33	7.54	66.32	65.7	4
NN-cINN	6.69	63.29	2.83	91.03	1.62	96.21	1.88	34.33	1.70	46.99	15.38	31.30	61.5	9
XGBoost-cINN	7.64	58.05	2.43	92.30	1.47	96.58	2.03	28.87	1.83	42.97	7.05	68.50	65.0	4
N-HiTS-cINN	11.70	35.79	1.40	95.58	1.28	97.01	2.31	19.17	2.06	35.72	6.37	71.53	62.9	6
TFT-cINN	8.53	53.21	1.67	94.71	1.43	96.67	2.42	15.41	1.66	48.18	4.95	77.88	67.4	3
Tololo ²	1.60	91.24	0.75	97.61	2.46	94.27	2.96	-3.70	1.35	57.98	3.56	84.10	71.7	1
Team Poland ³	2.60	85.75	1.05	96.68	1.24	97.11	4.06	-42.17	1.08	66.15	3.07	86.31	67.7	2
GMD ⁴	4.93	72.92	1.48	95.32	1.66	96.14	2.06	27.82	2.12	33.72	6.85	69.41	67.1	3

¹ This ranking is determined by how each individual model would have placed in the GEFCom2014 price forecasting challenge in 2014 and therefore assumes that the other models introduced in this article are not included

² [20]

³ [40]

⁴ [14]

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".

Author Contributions Kaleb Phipps: Conceptualisation, Methodology, Software, Investigation, Writing - Original Draft, Visualisation Benedikt Heidrich: Conceptualisation, Methodology, Software, Investigation, Writing - Original Draft Marian Turowski: Conceptualisation, Methodology, Investigation, Writing - Original Draft Moritz Wittig: Conceptualisation, Methodology, Software, Investigation, Writing - Review Ralf Mikut: Funding acquisition, Writing - Review & Editing, Supervision Veit Hagenmeyer: Funding acquisition, Writing - Review & Editing, Supervision

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability Statement The Electricity and Bike data sets analysed during the current study are available in the UCI repository via <https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014> and <https://archive.ics.uci.edu/dataset/275/bike+sharing+dataset> respectively. The Price and Solar data sets are available as a part of the GEFCom2014 forecasting challenge via [28]. The probabilistic forecasts analysed in our article can be recreated via code provided in GitHub upon acceptance of the paper.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.







References

1. Ardizzone L, Lüth C, Kruse J, et al (2019) Guided image generation with conditional invertible neural networks, [arXiv:1907.02392](https://arxiv.org/abs/1907.02392)
2. Arpogaus M, Voss M, Sick B et al (2023) Short-term density forecasting of low-voltage load using Bernstein-polynomial normalizing flows. *IEEE Transactions on Smart Grid* 14(6):4902–4911
3. Caffisch RE (1998) Monte Carlo and quasi-Monte Carlo methods. *Acta Numer* 7:1–49
4. Camporeale E, Agnihotri A, Rutjes C (2017) Adaptive selection of sampling points for uncertainty quantification. *Int J Uncertain Quantif* 7(4):1–22
5. Camporeale E, Chu X, Agapitov O et al (2019) On the generation of probabilistic forecasts from deterministic models. *Space Weather* 17(3):455–475
6. Challu C, Olivares KG, Oreshkin BN, et al (2023) N-HiTS: Neural hierarchical interpolation for time series forecasting. In: *Proceedings of the AAAI conference on artificial intelligence*, pp 6989–6997
7. Chen T, Guestrin C (2016) XGBoost: A scalable tree boosting system. In: *ACM SIGKDD International conference on knowledge discovery and data mining*. ACM, pp 785–794
8. Chernozhukov V, Wüthrich K, Zhu Y (2021) Distributional conformal prediction. *Proc Natl Acad Sci* 118(48):e2107794118
9. Cramer E, Witthaut D, Mitsos A et al (2023) Multivariate probabilistic forecasting of intraday electricity prices using normalizing flows. *Appl Energy* 346:121370
10. Cramer EY, Ray EL, Lopez VK et al (2022) Evaluation of individual and ensemble probabilistic forecasts of Covid-19 mortality in the US. *Proc Natl Acad Sci* 119(15):e2113561119
11. Dannecker L (2015) Energy time series forecasting: efficient and accurate forecasting of evolving time series from the energy domain, 1st edn. Springer Vieweg, Wiesbaden, Germany
12. De La Vallée Poussin C (1915) Sur l'intégrale de Lebesgue. *Trans Am Math Soc* 16(4):435–501
13. Dua D, Graff C (2019) UCI machine learning repository. <http://archive.ics.uci.edu/ml>, (Accessed 10 March 2022)
14. Dudek G (2016) Multilayer perceptron for GEFCom2014 probabilistic electricity price forecasting. *Int J Forecast* 32(3):1057–1060
15. Dumas J, Wehenkel A, Lanaspéze D et al (2022) A deep generative model for probabilistic energy forecasting in power systems: normalizing flows. *Appl Energy* 305:117871
16. Elbeltagi A, Srivastava A, Deng J et al (2023) Forecasting vapor pressure deficit for agricultural water management using machine learning in semi-arid environments. *Agric Water Manag* 283:108302
17. Fanaee-T H, Gama J (2014) Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence* 2:113–127
18. Fanfarillo A, Roozitalab B, Hu W et al (2021) Probabilistic forecasting using deep generative models. *GeoInformatica* 25(1):127–147
19. Fraccanabbia N, da Silva RG, Ribeiro MHD, et al (2020) Solar power forecasting based on ensemble learning methods. In: *2020 International joint conference on neural networks (IJCNN)*, IEEE, pp 1–7
20. Gaillard P, Goude Y, Nedellec R (2016) Additive models and robust aggregation for GEFCom2014 probabilistic electric load and electricity price forecasting. *Int J Forecast* 32(3):1038–1050
21. Ge L, Liao W, Wang S et al (2020) Modeling daily load profiles of distribution network for scenario generation using flow-based generative network. *IEEE Access* 8:77587–77597
22. Gneiting T, Raftery AE (2007) Strictly proper scoring rules, prediction, and estimation. *J Am Stat Assoc* 102(477):359–378
23. Gneiting T, Balabdaoui F, Raftery AE (2007) Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69(2):243–268
24. Gneiting T, Wolfram D, Resin J, et al (2022) Model diagnostics and forecast evaluation for quantiles. *Annual Review of Statistics and Its Application* 10
25. González Ordiano JA, Gröll L, Mikut R et al (2020) Probabilistic energy forecasting using the nearest neighbors quantile filter and quantile regression. *Int J Forecast* 36(2):310–323
26. Heidrich B, Bartschat A, Turowski M, et al (2021) pyWATTS: Python workflow automation tool for time series, [arXiv:2106.10157](https://arxiv.org/abs/2106.10157)
27. Heidrich B, Turowski M, Phipps K, et al (2022) Controlling non-stationarity and periodicities in time series generation using conditional invertible neural networks. *Appl Intell* pp 1–18
28. Hong T, Pinson P, Fan S et al (2016) Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond. *Int J Forecast* 32(3):896–913

29. Hyndman RJ, Athanasopoulos G (2018) *Forecasting: principles and practice*, 2nd edn. OTexts, Melbourne, Australia
30. Izmailov P, Vikram S, Hoffman MD, et al (2021) What are Bayesian neural network posteriors really like? In: International conference on machine learning, PMLR, pp 4629–4640
31. Jamgochian A, Wu D, Menda K, et al (2022) Conditional approximate normalizing flows for joint multi-step probabilistic electricity demand forecasting, 2201.02753
32. Kaplan D, Huang M (2021) Bayesian probabilistic forecasting with large-scale educational trend data: A case study using NAEP. *Large-scale Assessments in Education* 9(1):1–31
33. Kingma DP, Ba JL (2015) Adam: A Method for Stochastic Optimization. In: Bengio Y, LeCun Y (eds.) 3rd International Conference on Learning Representations (ICLR 2015)
34. Kingma DP, Dhariwal P (2018) Glow: Generative flow with invertible 1x1 convolutions. In: *Advances in Neural Information Processing Systems*, pp 10215–10224
35. Koenker R, Chernozhukov V, He X et al (2017) *Handbook of quantile regression*. CRC Press
36. Koochali A, Schichtel P, Dengel A et al (2019) Probabilistic forecasting of sensory data with generative adversarial networks-ForGAN. *IEEE Access* 7:63868–63880
37. Krzysztofowicz R (1999) Bayesian theory of probabilistic forecasting via deterministic hydrologic model. *Water Resour Res* 35(9):2739–2750
38. Lim B, Arık SÖ, Loeff N et al (2021) Temporal fusion transformers for interpretable multi-horizon time series forecasting. *Int J Forecast* 37(4):1748–1764
39. Liu J, Wu N, Qiao Y et al (2021) A scientometric review of research on traffic forecasting in transportation. *IET Intel Transport Syst* 15(1):1–16
40. Maciejowska K, Nowotarski J (2016) A hybrid model for GEF-Com2014 probabilistic electricity price forecasting. *Int J Forecast* 32(3):1051–1056
41. Matheson JE, Winkler RL (1976) Scoring rules for continuous probability distributions. *Manage Sci* 22(10):1087–1096
42. Murphy KP (2023) *Probabilistic Machine Learning: Advanced Topics*. MIT Press, <http://probml.github.io/book2>
43. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12:2825–2830
44. Petropoulos F, Apiletti D, Assimakopoulos V et al (2022) Forecasting: theory and practice. *Int J Forecast* 38(3):705–871
45. Raftery AE (2016) Use and communication of probabilistic forecasts. *Statistical Analysis and Data Mining: The ASA Data Sci J* 9(6):397–410
46. Rasul K, Sheikh AS, Schuster I, et al (2020) Multivariate probabilistic time series forecasting via conditioned normalizing flows, 2002.06103
47. Ribeiro MHD, da Silva RG, Ribeiro GT et al (2023) Cooperative ensemble learning model improves electric short-term load forecasting. *Chaos, Solitons & Fractals* 166:112982
48. Salinas D, Flunkert V, Gasthaus J et al (2020) DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int J Forecast* 36(3):1181–1191
49. Saravanan A, Parida S, Murugan M et al (2023) Thermal performance prediction of a solar air heater with a C-shape finned absorber plate using RF, LR and KNN models of machine learning. *Therm Sci Eng Prog* 38:101630
50. Sauer J, Mariani VC, dos Santos Coelho L, et al (2021) Extreme gradient boosting model based on improved Jaya optimizer applied to forecasting energy consumption in residential buildings. *Evolving Syst* pp 1–12
51. Scott C, Ahsan M, Albarbar A (2023) Machine learning for forecasting a photovoltaic (pv) generation system. *Energy* 278:127807
52. Smith RC (2013) *Uncertainty quantification: theory, implementation, and applications*, vol 12. Siam
53. Stankeviciute K, M Alaa A, van der Schaar M (2021) Conformal time-series forecasting. *Adv Neural Inf Process Syst* 34:6216–6228
54. Wang Y, Hug G, Liu Z et al (2020) Modeling load forecast uncertainty using generative adversarial networks. *Electric Power Syst Res* 189:106732
55. Wen R, Torkkola K (2019) Deep generative quantile-copula models for probabilistic forecasting. In: 36th International conference on machine learning (ICML2019)
56. Williams WH, Goodman M (1971) A simple method for the construction of empirical confidence limits for economic forecasts. *J Am Stat Assoc* 66:752–754
57. Winkler RL (1972) A decision-theoretic approach to interval estimation. *J Am Stat Assoc* 67:187–191
58. Xu Z, Lv Z, Li J et al (2022) A novel perspective on travel demand prediction considering natural environmental and socioeconomic factors. *IEEE Intell Transp Syst Mag* 15(1):136–159
59. Zaffran M, Féron O, Goude Y, et al (2022) Adaptive conformal predictions for time series. In: *International conference on machine learning*, PMLR, pp 25834–25866
60. Zhang L, Zhang B (2019) Scenario forecasting of residential load profiles. *IEEE J Sel Areas Commun* 38(1):84–95

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

Authors and Affiliations

Kaleb Phipps¹  · Benedikt Heidrich¹  · Marian Turowski¹  · Moritz Wittig^{1,2}  · Ralf Mikut¹  · Veit Hagenmeyer¹ 

Benedikt Heidrich
benedikt.heidrich@kit.edu

Marian Turowski
marian.turowski@kit.edu

Moritz Wittig
moritz.wittig@mobilityhouse.com

Ralf Mikut
ralf.mikut@kit.edu

Veit Hagenmeyer
veit.hagenmeyer@kit.edu

¹ Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, Eggenstein-Leopoldshafen 76344, Germany

² The Mobility House GmbH, St.-Cajetan-Str. 43, Munich 81669, Germany