Interpretable Net Load Forecasting Using Smooth Multiperiodic Features

Giray Ogut* Bennet Meyers† Stephen Boyd*

April 16, 2024

Abstract

We consider the problem of forecasting net load over a future horizon such as one day, using a trailing window of past net load values as well as date and time. We focus on three variations on this problem: point forecasts, marginal quantile forecasts, and generating conditional samples of the future values. These tasks can be accomplished using methods that range from basic, such as linear regression models, to sophisticated ones involving trees or neural networks. We propose a method that relies on linear regression using some custom engineered time-based features to capture multiple periodicities, such as daily, weekly, and seasonal, and their interactions, such as the variation in daily patterns over the year. Our proposed models are readily interpretable, and rely on efficient and reliable convex optimization to fit. At the same time, the method has strong predictive power, outperforming baseline techniques, and gracefully supports missing data. We illustrate our method on three years of hourly net load data for the state of Rhode Island, comparing predictions made with various subsets of the features. We provide an open source implementation that can be used for any time series that exhibits multiple periodicities.

*Department of Electrical Engineering, Stanford University
†SLAC National Accelerator Laboratory

Contents

1 Introduction

Forecasting net load is vital for utility companies to plan their energy distribution more effectively, as well as plan for contingencies. Forecasting net load has become more challenging in recent years with renewable energy sources coming on line, causing fluctuations in the net load [51]. In parallel, interest in capturing the uncertainty in net load forecasts has been growing. For example, the U.S. Department of Energy's Solar Energy Technologies Office organized a competition in 2023 to encourage the development of new probabilistic forecasting methods for net load [55]. Our participation in this competition motivated us to develop a simple and interpretable method for net load forecasting.

There is a vast literature on time series forecasting in general, and a large literature on net load forecasting, reviewed in §1.3. These range from very simple ones, such as using yesterday's netload as a prediction of today's, to very sophisticated ones that rely on neural networks or collections of decision trees. While the sophisticated methods can perform well, they have three drawbacks worth mentioning. First, these approaches typically require considerable data to train and validate. They rely on large corpora from which to learn their models, and are thus susceptible to all the difficulties of corpus curation and are difficult to integrate with expert knowledge (*e.g.*, physical models or constraints). Second, they are not immediately interpretable, transparent, or auditable. (While the intepretability of large neural network models is an open area of research, the fact remains that model parameters are not directly meaningful to an analyst.) Simply put, we do not know why they make the forecasts they do, and they are hard to fix if they go awry, making it difficult to put trust their predictions, even if the predictions are able to achieve high accuracy scores on standard data sets. Third, these methods require large and specialized computing hardware to train and, increasingly, to use [48, 58–60]. The material requirements underlying computation system are unfortunately not often discussed in literature [50], but we feel are particularly relevant in the context of the analysis of energy systems and net demand.

In this paper we focus on a relatively simple method for net load forecasting that achieves good results, even when trained on not much data, and is entirely interpretable, transparent, and auditable. It uses well known techniques, such as linear quantile regression, with features that are natural and interpretable. The features consist of a set of past realized values, as is used in an autoregressive forecast, and some features that depend only on time, *i.e.*, date and hour of day, informed by the quite old analysis of periodic phenomena. The time-based features are used to capture periodicities in the data, such as diurnal variation, weekly variation, and seasonal or annual variation.

We include cross terms that allow us to model diurnal variation that varies across the year. This method is simple enough that we can fully understand it, with high confidence that it will never produce anomalous forecasts, and the computations may be carried out on a standard laptop. Nonetheless, it is powerful enough to make good predictions. These predictions are useful for, *e.g.*, model predictive control (MPC) of energy distribution [39, 46].

The method we propose gracefully handles missing data. Our method can be almost fully automated using effective methods for choosing the hyperparameters that appear in it. It can be extended in simple ways, adding other features such as weather forecasts, or engineering more complex features.

We illustrate our method on real net load data, hourly reported data over four years for the state of Rhode Island. While our focus is on net load forecasting, our method is generic and can be used to forecast many other time series that exhibit multiple periodicities. We have developed an open-source implementation of our method.

1.1 Setting and tasks

We consider a real-valued time series, possibly with missing data,

$$y = (y_1, \ldots, y_T) \in (\mathbf{R} \cup \{?\})^T,$$

where $y_t = ?$ means that the value y_t is missing. We let $\mathcal{T} = \{t \mid y_t \in \mathbf{R}\}$ denote the set of known values.

Our underlying assumption is that y has statistics that include multiple periodicities, such as daily, weekly, and annual. Moreover these can include interactions, *i.e.*, the shape of the daily variation can change (smoothly) over the year.

Forecasting tasks. We focus on three related forecasting tasks, each of which uses the past M values $y_t, \ldots, y_{t-M+1}$ to make forecasts of the future H values, $y_{t+1}, \ldots, y_{t+H}$. We call M the memory of the forecaster and H the forecast horizon, and introduce the notation

$$p_t = (y_t, \ldots, y_{t-M+1}), \quad t = M, \ldots, T, \tag{1}$$

which is the vector of the M past values at time t, and

$$f_t = (y_{t+1}, \ldots, y_{t+H}), \quad t = 1, \ldots, T - H, \tag{2}$$

the vector of the H future values at time t. Thus our forecasting tasks involve predicting the future f_t from the past p_t. Note that p_t and f_t can contain

missing values. For future use we note that

$$(p_t)_i = y_{t-i+1}, \quad t = M, \ldots, T, \quad i = 1, \ldots, M$$

and

$$(f_t)_i = y_{t+i}, \quad t = 1, \ldots, T-H, \quad i = 1, \ldots, H.$$

The forecasting tasks are:

- *Point forecast.* Estimate f_t with forecasts denoted

  $$\hat{f}_t = (\hat{y}_{t+1}, \ldots, \hat{y}_{t+H}) \in \mathbf{R}^H.$$

  Note that while the data f_t can contain missing values, our forecasts $\hat{f}_t$ do not.

- *Marginal quantile forecast.* Estimate the $\eta_1, \ldots, \eta_Q$-quantiles of the entries of f_t. We denote these as

  $$q_{t,j} \in \mathbf{R}^H, \quad j = 1, \ldots, Q.$$

  Here $0 \leq \eta_1 < \cdots < \eta_Q \leq 1$ are the given quantiles to estimate.

- *Generate samples of the future.* Generate R plausible realizations of the future values, denoted

  $$f_{t,j} \in \mathbf{R}^H, \quad j = 1, \ldots, R.$$

  These are meant to be samples from the conditional distribution of f_t.

These forecasts are based on the past p_t, and can also depend on the time index t. Each of our three forecasting tasks is a function F that maps p_t, and t, into the forecasts. For a point forecast, the function F maps p_t and t to $\hat{f}_t$. For a marginal quantile forecast, F maps p_t and t to the quantiles $q_{t,j}$. If we are generating plausible samples of the future, F maps p_t and t to $f_{t,j}$.

1.2 Forecasting methods

Here we briefly describe some methods for carrying out forecasts, ranging from simple to complex.

Rolling median forecast (RMF). The rolling median forecast focuses on the smallest (fastest) periodicity in the data, with period Π (assumed to be an integer number of periods). For $i = 1, \ldots, H$ we form our estimate $\hat{y}_{t+i}$ as follows. First we find all indices $\tau \in \mathcal{T}$ within our window of memory, *i.e.*, $t - M + 1 \leq \tau \leq t$, that matches $t + i$ modulo Π. We then take $\hat{y}_{t+i}$ to be the

median of the associated values. This method produces a Π-periodic forecast $\hat{f}_t$. RMF is easily generalized to obtain marginal quantile forecasts, by simply replacing the median of the past data as above with a quantile.

RMF is tolerant of missing data, since it works with the median of a set of known past entries. It only fails when this set of past entries is empty. We will assume that the set of indices is nonempty, *i.e.*, there is at least one index in our memory window that equals $t+i$ modulo Π.

As a specific example, suppose the base period is hours, and the smallest period is $\Pi = 24$ (daily), and our memory window is $M = 672$ (four weeks). To forecast y_{t+4} (four hours in the future) we take the median of the known values in the set of past data

$$y_{t+4-24},\ y_{t+4-48},\ \cdots\ y_{t+4-24\times 28}.$$

In words: to predict the value at a particular hour, we take the median value of the known data within our memory window that corresponds to the same hour of the day. This is a very simple point forecast method, and is used as a baseline in practice as well as competitions such as the Net Load Forecasting Prize [55]. We will also use this as a baseline in our numerical experiments.

Regression. A regresssion point forecast has the form

$$\hat{f}_t = \theta^{\text{const}} + \theta^{\text{time}}\psi_t + \theta^{\text{past}}p_t,$$

where $\psi_t \in \mathbf{R}^N$ is a vector of features that are based on time only. The parameters in this model are $\theta^{\text{const}} \in \mathbf{R}^H$, the constant or offset term, $\theta^{\text{time}} \in \mathbf{R}^{H\times N}$, the time-based parameter, and $\theta^{\text{past}} \in \mathbf{R}^{H\times M}$, the autoregressive parameter. To choose these parameters we use a loss function such as the absolute value, which corresponds to estimating the median value. By replacing this loss function with one appropriate for estimating other quantiles, we obtain marginal quantile forecasts.

One advantage of the linear regression method is that it is interpretable. For example, $\theta^{\text{past}}_{ij}$ is the amount by which our forecast of y_{t+i} depends on the past value p_{t-j+1}. This is the forecasting method we will employ.

One critical issue that needs to be addressed in this regression model is how to handle missing data in p_t, both in training (*i.e.*, choosing the model coefficients) and also in forecasting. We will do this using a simple data fill method described later in §3.2.

More complex models. We can also create forecasts using complex models, such as those based on neural networks or multiple decision trees. These methods can work very well when a lot of data is available, but in general result in non-interpretable models.

Evaluating the forecasts. Forecasters are trained on training data, and evaluated on out-of-sample test data. To judge performance we use the average absolute error (AAE) for point forecasts, and the continuous ranked probability score (CRPS), or an approximation based on the specified quantiles, for marginal quantile forecasts. Evaluating the performance of generated future samples is a bit more complicated. One basic method is to evaluate the log-likelihood, under the model used to generate the samples, on the realized future values. An informal check of the generated samples is a Turing test, where we challenge a domain expert to distinguish generated samples of the future from the actual realized ones. Roughly speaking, we want the generated samples to 'look like' the realized ones.

1.3 Related work

In recent years there has been a lot of interest in forecasting net load. Reliable forecasts of net load are key for effective grid management, including demand side management, scheduling storage systems for grid stability, and facilitating coordination between energy suppliers and network operators in smart grids [30]. Most of the literature on net load forecasting focuses on point forecasting, while only a small fraction of them develop probabilistic forecasts. In our review, we will start by describing the literature on probabilistic forecasting methods, and then narrow our focus to probabilistic forecasting applied to net load.

Probabilistic forecasting. Probabilistic forecasting is a generalization of point forecasting, where the goal is to estimate the entire conditional distribution of the future values, given the past values of the time series and possibly other regressors. It is also known as density forecasting when full conditional distributions are estimated. If there is an assumption about the form of the conditional distribution, then the problem is reduced to estimating the parameters of the distribution, given the past values and becomes a distribution fitting problem.

The origins of probabilistic forecasting can be traced back to the 18th century, with the expression of uncertainty in weather forecasts by J. Dalton [29] in England and J. Lamarck [1] in France. The former used expressed uncertainty in terms of odds, using sentences such as 'the probability of a fair day to that of a wet one is as ten to one'. More quantitative and principled approaches to probabilistic forecasting were not developed until the 20th century. In 1920, C. Hallenbeck [2] reported the results of an experiment in which forecasts of rainfall in a 36 hour period were expressed in terms of numerical probabilities.

A more rigorous approach to probabilistic forecasting was developed in the last half of 20th century, with the development of statistical methods such as kernel density estimation [5], Markov chain Monte Carlo (MCMC) [42, 54], and quantile regression [22]. Although most of the literature on probabilistic forecasting comes from applications in meteorology, over the past several decades, there has been a growing interest in probabilistic forecasting in other fields, such as finance where it is used to estimate the risk of financial assets [20], or calculate tail events and value-at-risk [19].

Quantile regression. Methods for estimating conditional quantiles in data were first motivated by the idea of 'robust regression'. Various formulations of the robust estimation problem have been proposed over the years, with conditional median (*i.e.*, the 0.5 quantile) estimation being a popular approach since the 19th century. The idea of minimizing the sum of absolute errors in an over-determined system of equations actually predates the development of least-squares minimization, having been proposed in the mid-18th century by Boscovich and Simpson [13, 15]. In the 1970s, Koenker and Bassett proposed that it might be interesting to consider regression of quantiles other than the median, and they developed generalized quantile regression [10, 37]. The concept of consistent quantile estimation (in which the different quantile estimated are in the right order) is given in [24].

These methods rely on a 'pinball loss' function, defined later in (9). These simple loss functions are nothing more than a linear combination of the absolute value and a linear function. In quantile regression, we minimize the sum of pinball losses over the training data, which gives a set of quantile estimates. These quantiles can be estimated jointly or separately, and usually there are computational advantages to estimating them separately. If quantiles are estimated separately, there is the awkward possibility that the predicted quantiles are out of order. This is called the 'crossing quantiles' problem [12]. One common method to fix this in practice is to sort the predicted quantiles. More recently Chernezhukov et al. [25, 27] showed that sorted quantiles not only satisfy the natural monotonicity requirement, but also have smaller estimation error than the out of order quantiles. It is also possible to fit the quantiles together in such a way that there is no quantile crossing, but this results in a larger fitting problem.

Analysis of periodic and quasi-periodic phenomena. The analysis of repeating patterns in data is an age old problem. It is believed that ancient Babylonian mathematicians used harmonic analysis to understand astronomical observations as collections of 'periodic phenomena' [7]. In the 19th

century, Fourier developed his theory of periodic functions as sums of infinite series of trigonometric functions, on which we draw considerable inspiration for our methods. Quasi-periodic functions are defined by being "almost periodic" to within some tolerance, and the mathematical theory is "quite cumbersome and abounds with various auxiliary terms, theorems, and lemmas." [57]. Our proposed method models stochastic processes that exhibit both periodic and quasi-periodic statistics. Key to our approach is the ability to capture dynamics at different time scales (*i.e.*, multiple characteristic periods) with minimal data. When the periods in question are incommensurable, the method provides a readily interpretable model of quasi-periodic statistics. Otherwise, our method describes a model with periodic *and* quasi-periodic statistics, even in the case where periods are integer multiples of each other. In the case of two periods, the statistics are modeled to be approximately periodic on the time scale of the shorter period, while being exactly periodic on the longer period.

Continuous ranked probability score (CRPS). The evaluation of probabilistic forecasts has a rich history dating back to the 1950s, and first appeared in the meteorological literature with the introduction of the Brier score [3]. Later, the logarithmic scoring rule and quadratic scoring rule were introduced in statistics by Good [4] and de Finetti [11]. Later many other proper scoring rules such as discrete ranked probability score (DRPS) [6] and continuous ranked probability score (CRPS) [8] were introduced, again in the context of weather forecasting.

CRPS can be used as a metric to evaluate a model's performance when the target variable is continuous and the model predicts the target's distribution. For many distributions there is an analytic expression for the CRPS [45], and for non-parametric predictions, one could use the CRPS with the empirical cumulative distribution function. The CRPS is also very closely related to the well-known mean absolute error (MAE) used in point forecasting, and can be viewed as a generalization of the MAE to distributional predictions [18]. The MAE is a special case of the CRPS when the predicted distribution is degenerate (*i.e.*, a single point).

Net load forecasting. Load forecasting has become increasingly important as the energy sector evolves, particularly with the rise of distributed energy resources like rooftop solar panels and battery storage. These technologies have made electrical loads more volatile and unpredictable. To address this, the focus has shifted from traditional deterministic forecasting to probabilistic forecasting, which better captures uncertainties. This shift has been highlighted in various competitions aimed at pushing the boundaries of forecasting

accuracy such as the Global Energy Forecasting Competition [34].

Probabilistic load forecasting (PLF) studies the range of possible load variations and their probabilities. It mainly uses three forms: interval forecasting, quantile forecasting, and density forecasting [33]. Interval forecasting provides a range within which the load is expected to lie with a certain probability. Quantile forecasting predicts specific percentiles of the load distribution, offering a more detailed view of potential load levels. Density forecasting aims to predict the entire probability distribution of future loads, which is the most comprehensive approach.

Parametric methods and non-parametric methods are two primary ways to perform density forecasting. Parametric methods to predict load involve fitting the forecast errors to a specific parametrized probability distribution, such as the Weibull [14], Gaussian [16], Beta distributions [17, 21] or Gaussian mixture models [28]. This requires a deterministic forecast as a basis. But due to variability in load, assumed densities were not always accurate, and the methods were not widely adopted. Non-parametric methods, such as quantile regression, do not rely on predetermined distribution shapes, offering flexibility in handling load uncertainties. Quantile regression methods have also been extensively used in the literature for load forecasting [31, 38, 41, 44]. One of the common traits of these methods is that they use kernel density estimation with a Gaussian kernel or a Gaussian process to estimate the conditional distribution of the load. Shu et al. [52] developed a density estimation method based on transforming the individual quantile forecasts into the probability density curves and obtaining the weighted combination of different probability density forecasts.

Machine learning has played a significant role in enhancing PLF techniques. Methods like quantile regression neural networks (QRNN) [49], gradient boosted regression trees [35], random forests for quantile forecasting [23], and the use of deep learning methods [32, 40] have shown promise in improving prediction accuracy. These advancements reflect a broader trend towards using sophisticated computational methods to navigate the complexities of modern energy systems. However, as previously discussed, methods based on large neural network architectures suffer from a number of technical drawback including substantial data requirements, a lack of interpretability and auditability, and specialized, energy intensive computational requirements. Therefore, we present our contribution as a more savvy, interpretable baseline forecasting method that, at the very least, all more advanced approaches should be able to significantly outperform, in order to justify these drawbacks.

Data fill methods. Handling missing data in machine learning is a big topic in itself [53]. The most straightforward strategy to handle missing data in training a model is to delete any training data that contains missing values. This can lead to considerable data loss, and especially in models with autoregressive features, since one missing value of y_t leads to M missing values in p_t.

A method that does not suffer from data loss is to fill missing entries with some reasonable value. This fill method must be causal, *i.e.*, not depend on data not available in time period t when the forecast is made at time t. This allows it to be used when making actual forecasts.

One simple fill method is forward fill, where missing entries are filled using the most recent known value. Forward fill can work well when there are not long gaps in the data. Linear interpolation estimates missing values by interpolating between neighboring known points. This method is also not effective when there are gaps of missing data, and in addition is noncausal when the current value is unknown. A causal hybrid method can be used, such as carrying out linear interpolation when both end points are known, and forward fill when the later end point is not known. Another simple method fills missing values with a basic statistic such as the mean, median, or mode, of the time series values.

1.4 Outline

This paper is organized as follows. In §2 we describe a general method for handling time variation in our forecasts, using features which depend only on time. We introduce the concept of multiperiodic basis functions and explain how to construct them. In §3 we describe our method for point forecasting, marginal quantile forecasting and how to generate conditional samples. In §4 we provide a numerical example using real-world net load data. We go over the data, explain how we instantiate our method, and show the results of our experiments for the three tasks.

2 Smooth multiperiodic time features

In this section we describe our general method for handling time variation in our forecasts, using time features, which are features that depend only on time.

2.1 Multiperiodic basis functions

We start by creating an appropriate set of time-based basis functions $\phi_i : \mathbf{R} \to \mathbf{R}$ for $i = 1, \ldots, N$. (We will evaluate these for integer values of t, *i.e.*, at our time periods, but they are defined for other values as well.) We interpret $\phi_i(t)$ as the value of the ith time-based basis function at time t.

Period harmonics. Consider a period $\Pi > 0$, which need not be an integer. Natural basis functions for a smooth Π-periodic function are the $2K$ sinusoidal Fourier terms or harmonics up to K,

$$\cos(2\pi kt/\Pi), \quad \sin(2\pi kt/\Pi), \quad k = 1, \ldots, K.$$

We refer to k as the harmonic number. A basis function with harmonic number k and period Π has period Π/k.

Multiple periods. To model different periodicities, we have K_i harmonics for each of the periods Π_i, $i = 1, \ldots, M$, with $\Pi_P < \cdots < \Pi_1$. This gives $\sum_{i=1}^{P} K_i$ basis functions. Each of these basis functions has a harmonic number.

Cross terms. To model the interactions between the different periods, we generate additional basis functions that are products of the basis functions associated with different periods. If we take all such products we obtain

$$\sum_{1 \leq i < j \leq P} (2K_i)(2K_j)$$

cross term basis functions. (To simplify the model, we might not use all cross terms.)

With each of these cross term or product basis functions we associate the smaller period in the product, and we assign a harmonic number which is the one associated with the smaller period. For example, consider the cross term

$$\cos(2\pi kt/\Pi)\sin(2\pi\tilde{k}/\tilde{\Pi}),$$

with $\Pi < \tilde{\Pi}$. We associate it with period Π, and assign harmonic number k to it. Unless $\Pi/\tilde{\Pi}$ is rational, these cross term basis functions are not periodic, but they are almost periodic [57].

Creating new features as products of existing features, as we do to create the cross terms, is a well known technique in machine learning. For example, it is used in the construction of wavelet bases [26, Ch. 5].

Full basis. All together we have

$$N = 2\sum_{i=1}^{P} K_i + 4 \sum_{1 \le i < j \le P} K_i K_j$$

time-based basis functions. We denote the basis functions as $\phi_1, \ldots, \phi_N$, with associated harmonic numbers $k_1, \ldots, k_N$. These range from $\min\{K_1, \ldots, K_P\}$ to $\max\{K_1, \ldots, K_P\}$. Note that all the basis functions are perpendicular to each other.

We emphasize that the idea of using sinusoidal basis functions to encode seasonal patterns is not a new idea, having already been described in Appendix A of [46] for the purpose of establishing a seasonal baseline and in [43] in the context of time series forecasting for PV data. More recently in [61] the authors enforce smoothness by requiring estimated signal to be representable as a linear combination of smooth basis functions such as splines and polynomials. However, to the best of our knowledge, using cross terms to model interactions between different periods is a novel idea and not previously described in the literature. We consider this as an important contribution of our work because it allows us to capture the interactions between different periods, such as diurnal variation that varies across the year.

Time based features. Given the basis functions $\phi_1, \ldots, \phi_N$, we create time-based feature vectors as

$$\psi_t = (\phi_1(t), \ldots, \phi_N(t)) \in \mathbf{R}^N, \quad t = 1, \ldots, T. \tag{3}$$

The basis functions ϕ_i are defined for all real arguments, but in creating the features we evaluate them only at integer times.

2.2 Example

Suppose t denotes hours, and we have $P = 3$ periodicities, annual, weekly, and daily, corresponding to periods $\Pi_1 = 8765.8$, $\Pi_2 = 168$, and $\Pi_3 = 24$. We take $K_1 = 2$, $K_2 = 3$, $K_3 = 4$, *i.e.*, we use 2 harmonics for annual, 3 for weekly, and 4 for daily periods. Examples of basis functions are

$$\phi_i(t) = \cos(2\pi 3t/24),$$

which is a third harmonic of the daily period with harmonic number 3,

$$\phi_i(t) = \sin(2\pi 2t/8765.8),$$

which is a second harmonic of the annual period with harmonic number 2, and their product

$$\phi_i(t) = \cos(2\pi 3t/24)\sin(2\pi 2t/8765.8),$$

which is a daily-annual cross term, with harmonic number 3.

In this example there are 88 basis functions associated with the daily period, 30 basis functions associated with the weekly period, and 4 basis functions associated with the annual period. The total number of basis functions is 122, with harmonic numbers ranging from 1 to 4.

2.3 A roughness measure

Here we describe a simple quadratic roughness measure, that we use for regularization when fitting a forecaster.

First consider the case when there is only one period Π_1, with K_1 harmonics, so $N = 2K_1$. A linear combination of these basis functions, $f(t) = \sum_{i=1}^{N} c_i\phi_i(t)$, is a truncated Fourier series. We have

$$\frac{1}{\Pi_1}\int_0^{\Pi_1} \big(f'(t)\big)^2 \; dt = \frac{1}{2}\sum_{i=1}^{N} k_i^2 c_i^2.$$

This justifies $\sum_{i=1}^{N} k_i^2 c_i^2$ as a measure of roughness for the special case with one period.

Motivated by this we define the roughness measure for the P different periods as

$$R_i = \frac{1}{2}\sum_{j\in\mathcal{P}_i} c_j^2 k_j^2, \quad i = 1, \ldots, P, \tag{4}$$

where $\mathcal{P}_i$ is the set of indices of basis functions associated with period i. Thus we measure roughness separately for each period. Our overall roughness measure will be

$$\mathcal{R} = \lambda_1 R_1 + \cdots + \lambda_P R_P, \tag{5}$$

where $\lambda_1, \ldots, \lambda_P$ are positive hyperparameters associated with the different periods. The roughness measure $\mathcal{R}$ is a function of the coefficients $c_1, \ldots, c_N$, and the hyperparameters $\lambda_1, \ldots, \lambda_P$ so we write it as $\mathcal{R}(c; \lambda)$.

3 Forecasts

3.1 Data split

We split the data into train, validation and test sets, each described by a subset of the known time values $\mathcal{T} \subseteq \{1, \ldots, T\}$, denoted as $\mathcal{T}^{\text{train}}$, $\mathcal{T}^{\text{val}}$, and $\mathcal{T}^{\text{test}}$,

respectively. We refer to $\mathcal{T}^{\text{train}} \cup \mathcal{T}^{\text{val}}$ as the in-sample data, and $\mathcal{T}^{\text{test}}$ as the out-of-sample data.

The in-sample and out-of-sample split is done sequentially, meaning that for each in-sample t and out-of-sample s, we have $t < s$. Within the in-sample set, train and validation sets are created by randomly splitting the in-sample set with some predetermined target ratio ratio such 70% train and 30% validation.

3.2 Point forecast

We use a simple and interpretable linear regression forecast of future values,

$$\hat{f}_t = \theta^{\text{const}} + \theta^{\text{time}}\psi_t + \theta^{\text{past}}p_t, \tag{6}$$

where $\psi_t \in \mathbf{R}^N$ is the time-based feature vector defined in (3) and $p_t \in (\mathbf{R} \cup \{?\})^M$ is the vector of autoregressive features, *i.e.*, past values, defined in (1). The parameters of the model are the vector $\theta^{\text{const}} \in \mathbf{R}^H$, and the matrices $\theta^{\text{time}} \in \mathbf{R}^{H\times N}$ and $\theta^{\text{past}} \in \mathbf{R}^{H\times M}$.

Missing data in the past vectors. While the time-based feature vector does ψ_t not contain missing data, the autoregressive feature vector p_t can. Indeed, each missing value in the original time series gives rise to a missing entry in M different values of p_t. To handle this we fill in missing entries in each p_t using forward fill method. First we fill the original time series by replacing each unknown entry y_t After forward filling of missing entries in y_t, we standardize it, *i.e.*, subtract a median value and divide by the absolute error. (The time-based features are already approximately standardized.) We use the same symbol for these filled and standardized values, to keep the notation simple. This fill in procedure is used for creating the vectors p_t, but *not* the future vectors f_t to be used in training, which still contain missing values.

Fitting the point forecast model. To fit the parameters in the point forecast model (6), we minimize the convex function

$$\sum_{t,i|t+i\in\mathcal{T}^{\text{train}}} \frac{1}{2}|(f_t)_i - (\hat{f}_t)_i| + \mathcal{R}(\theta^{\text{time}}; \lambda^{\text{time}}) + \lambda^{\text{past}}\|\theta^{\text{past}}\|_F^2, \tag{7}$$

where $\|\cdot\|_F$ is the Frobenius norm. Note that the loss function, the first term in the objective above, only uses known entries in the training set. The regressor p_t, however, is filled using the forward fill method, and also standardized.

The optimization problem (7) is separable across the entries of f_t, which means we can solve for each row of the parameters separately, in parallel.

We use an absolute value loss function, which corresponds to forecasting the median value. The fitting problem (7) has $P+1$ hyperparameters, $\lambda^{\text{time}} \in \mathbf{R}^P$ and $\lambda^{\text{past}} \in \mathbf{R}$.

3.3 Hyperparameter selection

We use cross validation on the in-sample data to choose appropriate values of the hyperparameters. For a given set of hyperparameters, we train the model and evaluate the performance on the validation set using AAE on the known entries, as in (7). We do this for a number of values of the hyperparameters, and then choose one that achieves the smallest validation AAE, biasing our choice toward larger values, *i.e.*, more regularization. Once we choose the hyperparameters, we re-train the model on the entire in-sample set. Finally, we evaluate our model using the test data error, to be sure it is not too far from the validation error.

Refined grid search. There are many methods for choosing candidate hyperparameter values. One traditional method, useful when P (the number of hyperparameters) is modest, is a grid search, where we evaluate all combinations of M^{grid} values of each parameter, spaced logarithmically over some given range. Grid search is often carried out with a first crude parameter gridding, with the candidate values spaced by a factor of ten or so; then, when good values of these parameters are found, a more refined grid search is used to focus in on parameters near the good ones found in the first crude search. Gridding is practical only when $(M^{\text{grid}})^{P+1}$, the number of hyperparameter values for which we form a model, is modest.

Cyclical greedy search. Another simple method, which we have found effective, uses a cyclical greedy search. We start with some initial choice of hyperparameters. We choose one of the hyperparameters and increase it by some factor such as $\eta = 2$, and evaluate it (provided it does not go above some given upper limit). If it achieves smaller validation error, we take this as our new value for that hyperparameter. If not, we decrease the hyperparameter by the factor η, and try this value (provided it does not go below a given lower limit). If this decreases the validation error, we take it as the new value; otherwise we move on the next hyperparameter. We stop when one cycle through all hyperparameters does not improve the validation error. Like a grid search, this greedy method can also be run first with a crude search with $\eta = 3$ or $\eta = 10$, and then with a finer search, using, *e.g.*, $\eta = 1.4$.

3.4 Marginal quantile forecasts

To estimate marginal quantiles we simply modify the loss function used in (7) to estimate quantiles other than $\eta = 0.5$. For the η_j quantile, we choose parameters by minimizing

$$\sum_{t,i|t+i\in\mathcal{T}^{\text{train}}} \ell_{\eta_j}\left((f_t)_i - (\hat{f}_t)_i\right) + \mathcal{R}(\theta^{\text{time}};\lambda^{\text{time}}) + \lambda^{\text{past}}\|\theta^{\text{past}}\|_F^2, \tag{8}$$

where ℓ_{η_j} is the pinball loss associated with jth quantile η_j defined as

$$\ell_{\eta_j}((f_t)_i, q_{t,j}) = \begin{cases} \eta_j\left((f_t)_i - q_{t,j}\right) & (f_t)_i \geq q_{t,j} \\ (1-\eta_j)\left((f_t)_i - q_{t,j}\right) & (f_t)_i < q_{t,j}. \end{cases} \tag{9}$$

Data split, standardization, regularization, and hyperparameter selection are done exactly as in point forecast model described above, except that we use the pinball loss (9) instead of the absolute value loss and use CRPS instead of AAE to evaluate the performance of the model while tuning hyperparameters. In fact, we use an approximation of the CRPS, which is the sum of the pinball loss values. This is same as using pinball loss to judge the error for each quantile separately.

As with the point forecast, the marginal quantile estimation problem can be solved separately for each horizon $h = 1, \ldots, H$. Furthermore, it can also be solved separately for each quantile $j = 1, \ldots, Q$. But this creates the possibility that the estimated quantiles are out of order. One way to ensure that the estimated quantiles are in order is to sort them after fitting the model [25, 27]. Alternatively, we can couple the quantile estimates by adding the constraint that the quantiles are in order, *i.e.*,

$$q_{t,j} \leq q_{t,j+1}, \quad j = 1, \ldots, Q-1, \quad t = 1, \ldots, T.$$

This is a convex constraint that ensures the quantile estimates are in order, at least on the training data. Finding separate quantile models, in parallel, followed by sorting the quantiles when needed, is faster and works well in practice.

3.5 Conditional sample forecasts

The simplest model of the conditional distribution of f_t, given p_t and ψ_t, that takes into account dependence among the entries of f_t, is Gaussian. We model f_t, conditioned on p_t and ψ_t, as

$$f_t \sim \mathcal{N}(\hat{f}_t + \mu, \Sigma), \tag{10}$$

where $\hat{f}_t$ is our point estimate, μ is the point forecast error mean, and $\Sigma \in \mathbf{R}^{H\times H}$ the point forecast error covariance.

We define the point forecast errors or residuals as $r_t = \hat{y}_t - y_t$; our task is to estimate its mean and covariance. One challenge here is missing data, since the residual vector inherits missing entries from y_t. The standard method for fitting μ and Σ from samples $r_1, \ldots, r_T$ that contain missing entries is the expectation-minimization (EM) method [9, 47]. A much more naïve method, described below, also works well when regularization is used.

Empirical error mean and covariance. A simple and naïve estimate of Σ is the empirical covariance, computed separately for each entry, using only the known data values. We start with the empirical means of the entries of r, using only known entries,

$$\mu_i^{\text{emp}} = \frac{1}{|\mathcal{S}_i|}\sum_{t\in\mathcal{S}_i}(r_t)_i, \quad i = 1, \ldots, H,$$

where $\mathcal{S}$ is the set of time periods for which $(r_t)_i$ is known. We then form the empirical covariance, over known entries, as

$$\Sigma_{ij}^{\text{emp}} = \frac{1}{|\mathcal{S}_{ij}|}\sum_{t\in\mathcal{S}_{ij}}\left((r_t)_i - \mu_i)\right)\left((r_t)_j - \mu_j\right),$$

where $\mathcal{S}_{ij} = \mathcal{S}_i \cap \mathcal{S}_j$ is the set of time periods where both $(r_t)_i$ and $(r_t)_j$ are known. A well known potential drawback is that the empirical covariance matrix Σ^{emp} need not be positive semidefinite, since the entries are found separately. This is addressed with regularization decribed below, which also improves the performance of the estimator.

Factor regularization. We describe a simple regularization method that can improve the covariance estimate, either EM or empirical, which we denote as $\tilde{\Sigma}$. Regularization also addresses the issue of Σ^{emp} not being positive semidefinite.

We use a factor form, $\Sigma = FF^T + D$, where $F \in \mathbf{R}^{H\times q}$ and $D \in \mathbf{R}^{H\times H}$ is diagonal with nonnegative entries, and $q \leq H$ is number of factors. We take FF^T as the rank q appromixation of $\tilde{\Sigma}$ obtained from an eigendecomposition. (With the empirical covariance matrix we must choose q so that the first q eigenvalues are nonnegative.) We choose D so that $\Sigma_{ii} = \tilde{\Sigma}_{ii}$, $i = 1, \ldots, H$ [36]. The number of factors q can be chosen as the number of significant eigenvalues of $\tilde{\Sigma}$.

Generating forecasts. To obtain samples $f_{t,1}, \ldots, f_{t,R}$ of the future, we generate samples from the conditional distribution (10).

4 Numerical example

4.1 Data

We used ISO New England (ISO-NE) electric energy load data for Rhode Island between January 1 2020 and December 31 2022. The data is sampled hourly, so in our full data set we have 26304 data values. There is no missing data. This data can be accessed at [56]. We used data from January 1 2020 to December 31 2021 for in-sample training, and data from January 1 2022 to December 31 2022 for test. There are 17544 data points for training and 8760 data points for test.

To visualize seasonal variation, we first look at daily average net load, shown in figure 1. We observe that in-sample and test data are similar, with a small increase in net load in the test period compared to the in-sample period. The median of training data is 850.7 kWh and its mean absolute deviation is 150.9. The median of test data is 858.0 kWh and its mean absolute deviation is 151.3. We also observe that there are 2 major peaks in energy demand each year, a smaller one in winter and a larger one in summer.

Next, we look at the data at a finer scale. Figure 2 shows the demand for the first week of January 2022 and the first week of August 2022, both in the test data. We observe several expected phenomena, *e.g.*, demand is higher during the day than at night, and a bit higher in summer than in winter. We can see some small variation over a week. One interesting observation is that the shape of the daily demand in winter differs considerably from the shape of the daily demand in summer. In winter we see a double bump, with peaks in the morning and afternoon, while in summer we see a smoother daily variation with one peak in the early afternoon. Also, even if the daily demand curves are similar for the same season, they show some variation. For instance, the daily demand on Monday 08-01-2022 is smaller and less smooth than the other days of the week.

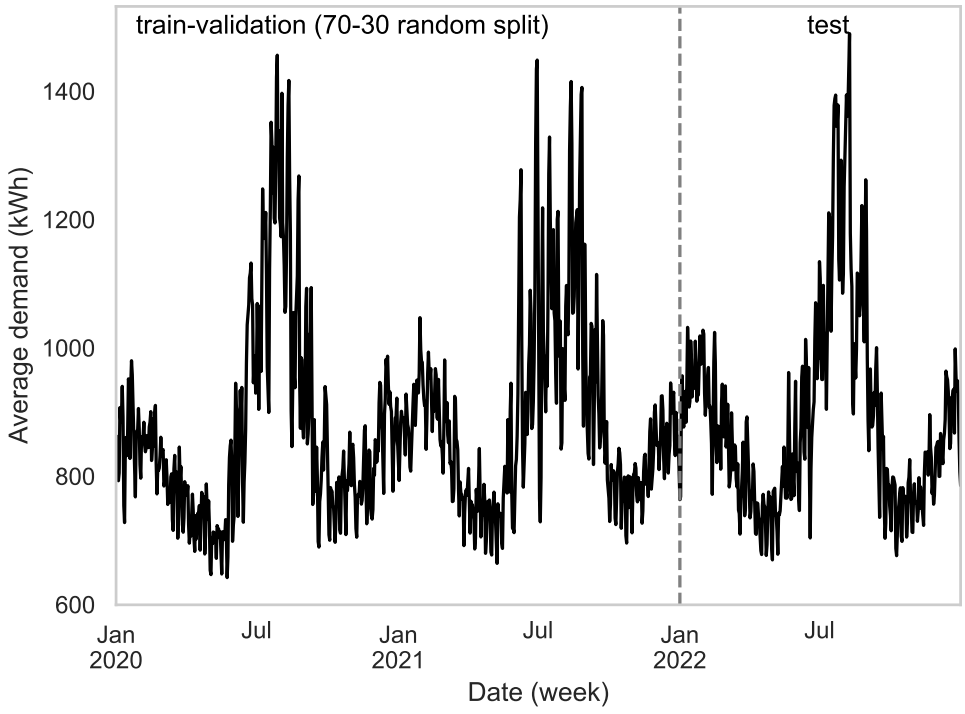

Figure 1: Daily average net load. The dashed line shows the in-sample / test split.

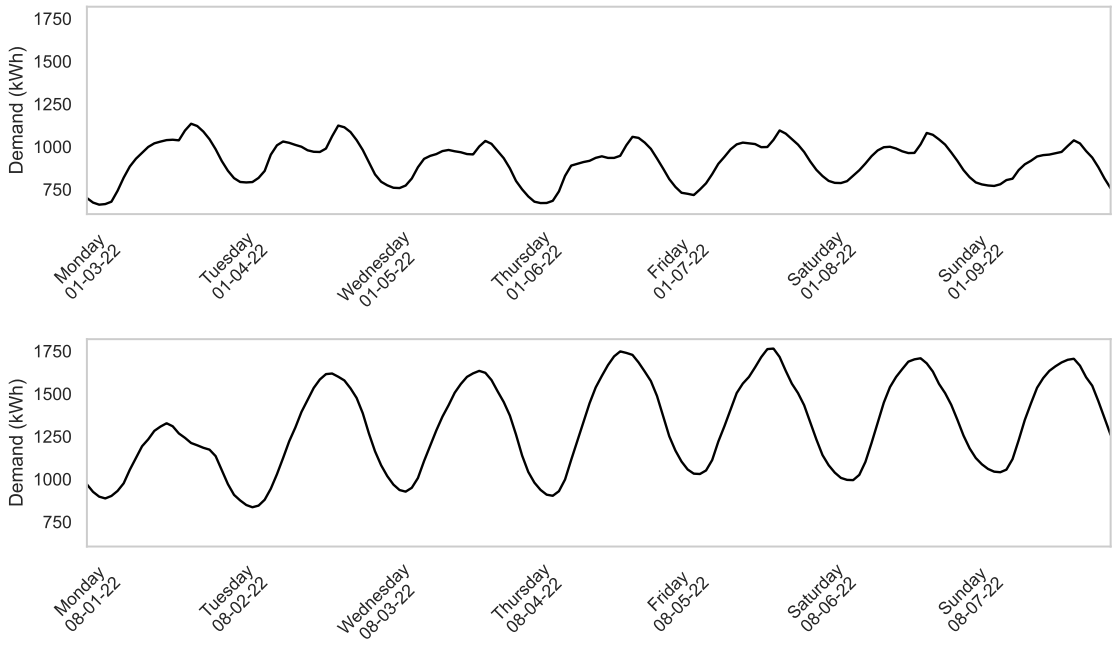

Figure 2: Two different weeks of demand data on test set. *Top.* Winter. *Bottom.* Summer.

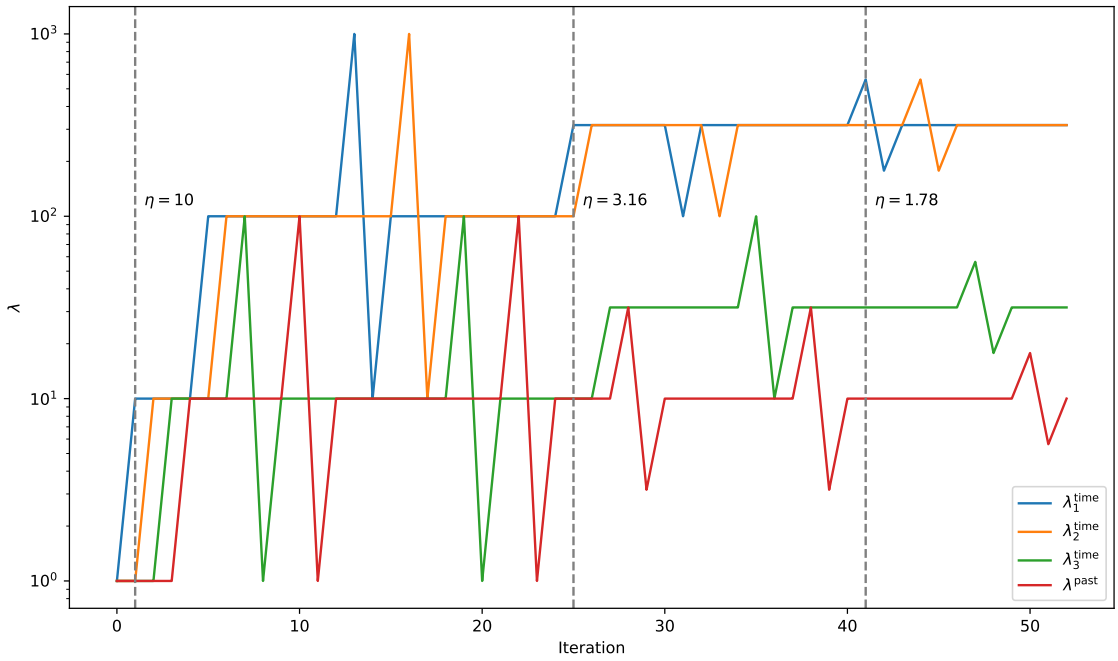

Figure 3: Cyclical hyperparameter tuning with $\eta = 10$, 3.16, and 1.87.

4.2 Prediction horizon, memory, and basis

We choose forecast horizon $H = 24$ (one day), and memory $M = 72$ (three days). For our multiperiodic basis we choose $P = 3$ periodicities: daily, weekly, and annual, corresponding to periods $\Pi_1 = 8765.8$, $\Pi_2 = 168$, and $\Pi_3 = 24$. We take $K_1 = 2, K_2 = 3, K_3 = 4$.

4.3 Point forecast

We use cyclical greedy search as explained in §3.3 to choose hyperparameters

$$\lambda^{\text{past}} = 10.00, \quad \lambda_1^{\text{time}} = 316.23, \quad \lambda_2^{\text{time}} = 316.23, \quad \lambda_3^{\text{time}} = 31.62.$$

The hyperparameter search is shown in figure 3. Our search ends with coefficents associated with yearly and weekly terms more heavily regularized than those associated with daily terms and past values. It is worth noting that yearly variation still manifests itself through cross terms with other periodicities.

We also show different versions of our forecasting model that includes subsets of features as well as using a baseline method commonly used in practice

Model	AAE (kWh)
Baseline model	70.4
Time features alone	68.6
Past features alone	43.3
All features except cross terms	41.2
Full features	**37.1**

Table 1: AAE on test set for different models.

and compare their performances. For the models that need hyperparameter tuning, we did cyclical greedy search for each of them and report results on best values.

- *Baseline model.* This model is same as RMF presented in §1.2, using a trailing window of the past 30 days' median.
- *Time features alone.* This model uses the same multiperiodic features but does not use past values of the time series.
- *Past features alone.* This model uses the past values of the time series, but does not use time features.
- *All features except cross terms.* This model uses the past values of the time series as well as time features but does not use cross terms.

Table 1 shows the average absolute error (AAE) on the test set for our model and the other models. We see that just using past features gives a more significant improvement on baseline than just using time features. Still, the lowest AAE is achieved by combining these two types of features. We also observe that the cross terms give a substantial reduction in AAE.

Figure 4 shows the AAE for different models across different horizons. We observe that our proposed method gives the lowest AAE for all horizons.

Figure 5 shows the point forecasts using our proposed model. Both for summer and winter we were able to come up with predictions that are close to actual realizations.

Interpretability. We can interpret our model by simply looking at the coefficients of θ^{past} and θ^{time}. Since our features are reasonably well standardized, the magnitude of the coefficients can be used to explain which features were important for the forecasted values.

Figure 6 shows entries of θ_h^{past} for horizons $h = t+1, t+4, t+16$. First, we observe that the coefficients are approximately sparse, with most of the entries near zero. This shows that only a few past values of the time series are important for the forecasted values. Next, we see that especially for near

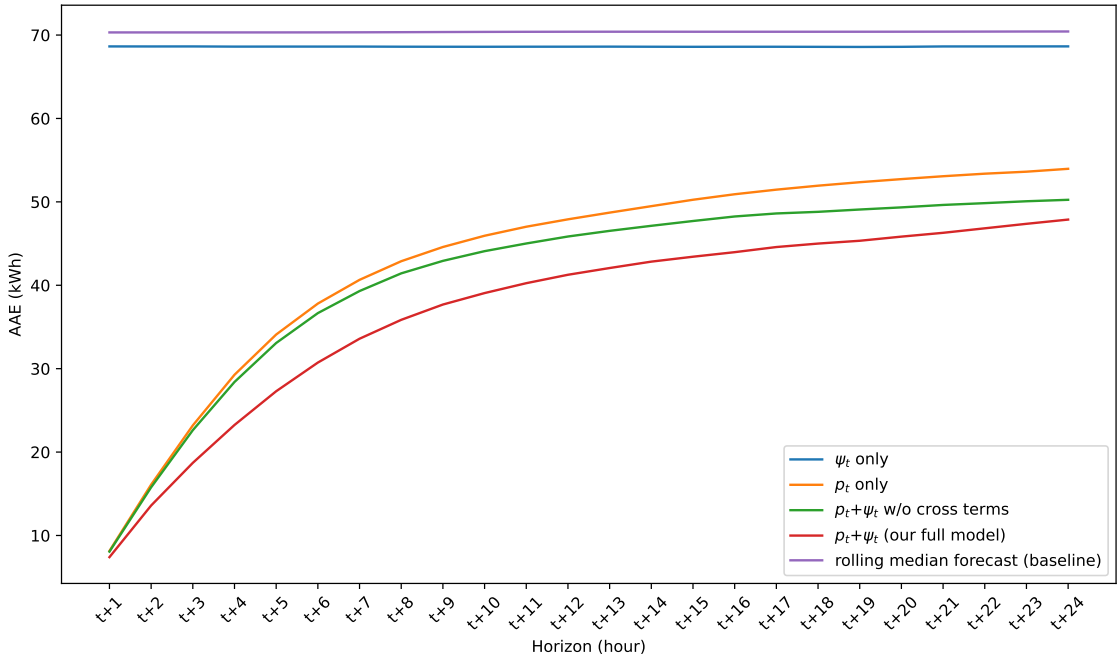

Figure 4: AAE for different models, for different horizons.

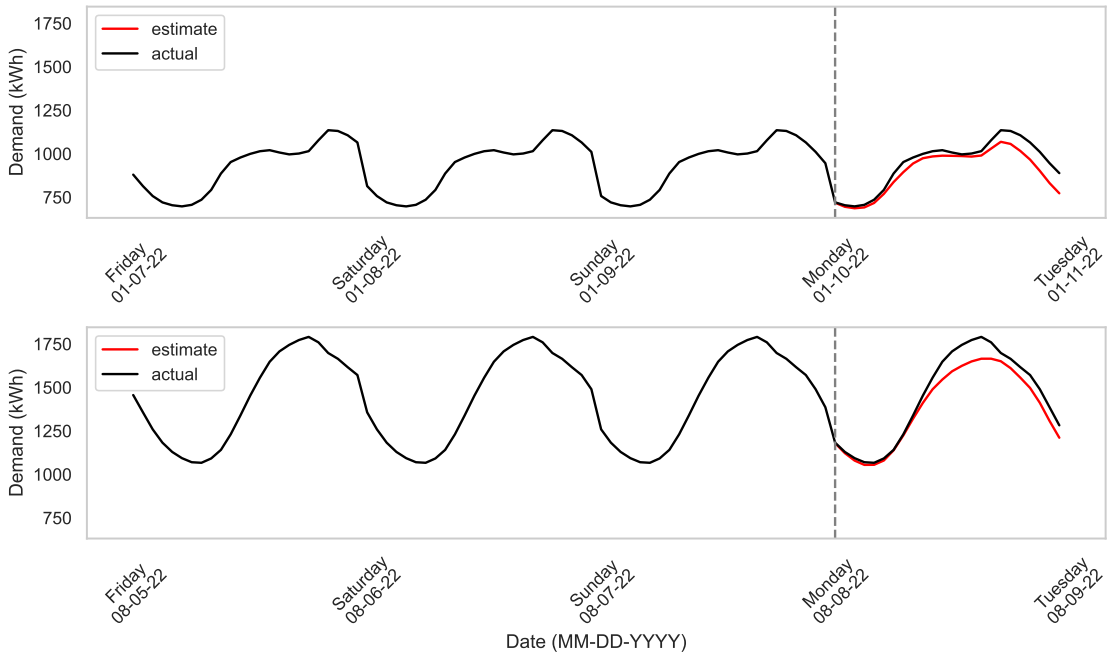

Figure 5: Point forecasts using full features. *Top.* Winter. *Bottom.* Summer.

horizons, the largest coefficients are located at or near 24 hour intervals. This can be interpreted as the model paying special attention to the previous values 24, 48, and 72 hours ago. Note that with increasing horizon, we observe more volatility in the coefficients and less structure. This confirms our expectation that last known values becomes less useful as we forecast further into the future. We can similarly interpret the coefficients of θ^{time} by looking at the magnitude and sign of the entries and associating them with the basis functions.

Another way to interpret our model is to look at the norms of the rows of θ^{past} and θ^{time}. Their relative norms can be used to investigate how important time features are compared to past features for various horizons. Figure 7 shows the ℓ_2 norm of rows of θ^{past} and θ^{time}. We observe that the norm of rows of θ^{past} generally decreases with increasing horizon whereas the norm of rows of θ^{time} increases with increasing horizon. This shows that past features lose their relative importance as we forecast further into the future whereas time features gain more importance.

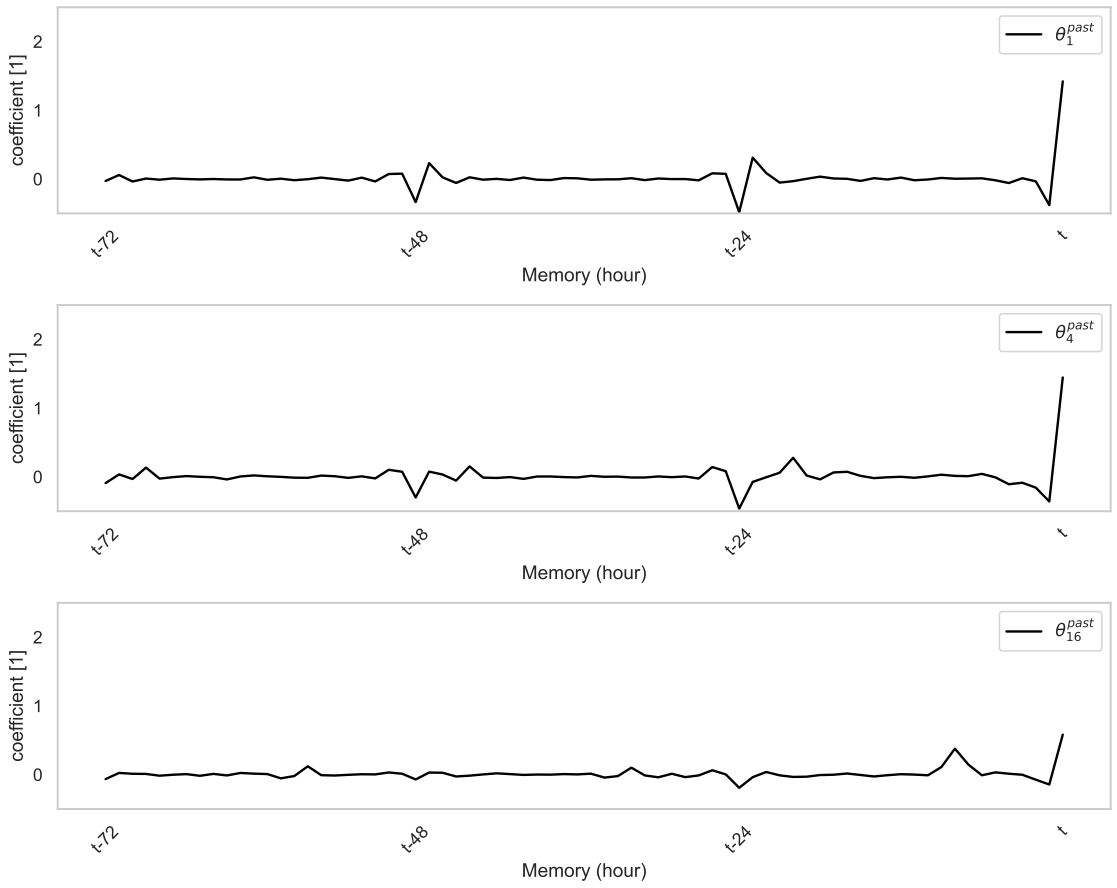

Figure 6: Coefficients of θ. *Top.* Horizon $t + 1$. *Middle.* Horizon $t + 4$. *Bottom.* Horizon $t + 16$.

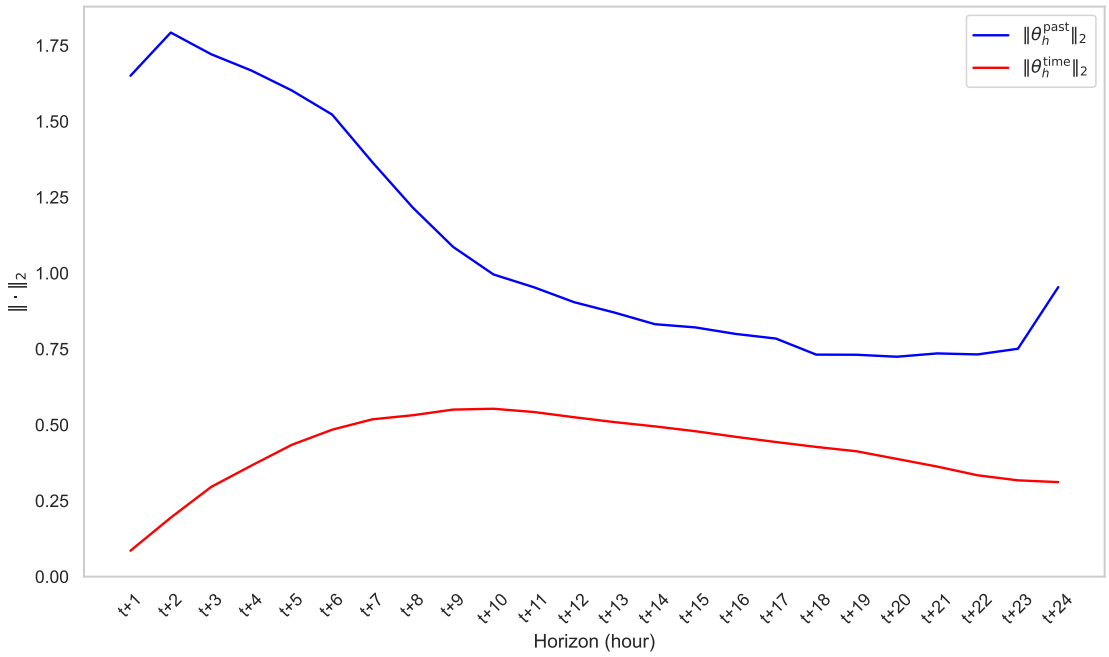

Figure 7: ℓ_2-norm of rows of θ^{time} and θ^{past}.

Model	CRPS (kWh)
Baseline model	24.8
Time features alone	26.1
Past features alone	20.8
All features except cross terms	14.5
Full features	**13.1**

Table 2: Average CRPS on test set for different models.

4.4 Marginal quantile forecasts

We carried out cyclical greedy search to tune hyperparameters but this time evaluated for distributional forecast performance as measured by CRPS, to find the values

$$\lambda^{\text{past}} = 10.00, \quad \lambda_1^{\text{time}} = 316.23, \quad \lambda_2^{\text{time}} = 316.23, \quad \lambda_3^{\text{time}} = 31.62,$$

the same as the ones found for a point forecast.

We compare the performance of our proposed method with the models described in section §4.3. Table 2 shows the average CRPS on the test set for our model and the other models. We see similar results to the point forecast case.

Figure 8 shows the CRPS for different models across different horizons. It is very similar to figure 4 for point forecast. One major difference is that this time importance of using time features together with past features becomes more apparent. This is because the performance of the model with just past features quickly deteriorates with increasing horizon and reaches the baseline level.

Figure 9 shows the marginal quantile forecasts using our proposed model. For winter we see tight bands of quantiles centered around the actual realization. For summer, even though the actual realization was close to the estimated median, the interquantile distance is much higher. This can be explained by the fact that net load values in summer showed more volatility.

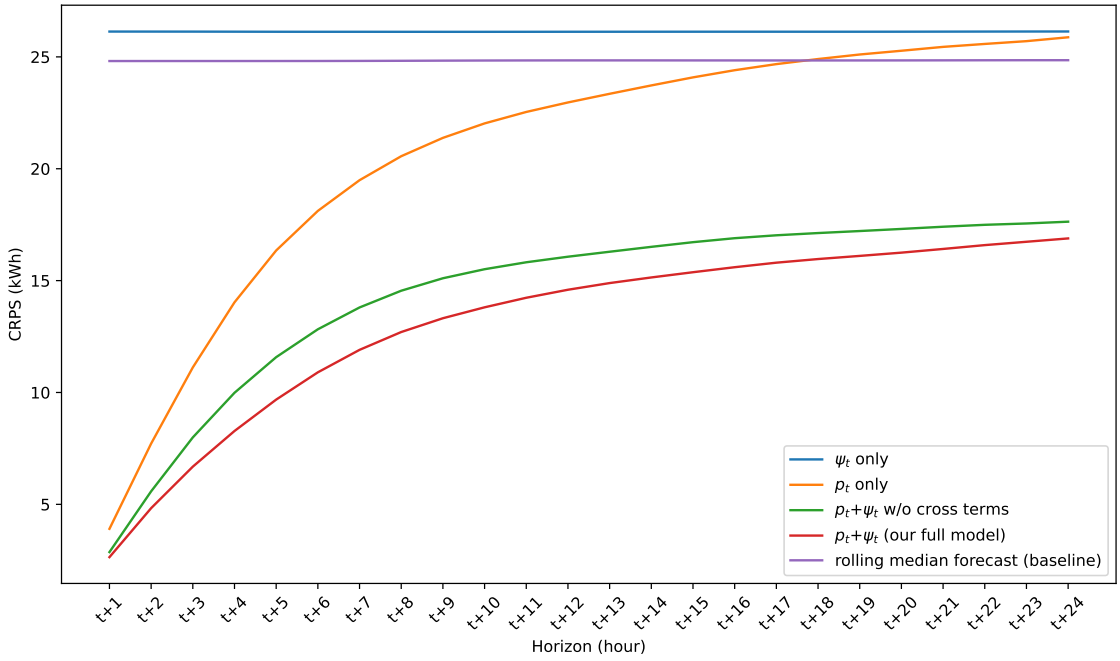

Figure 8: CRPS for different models.

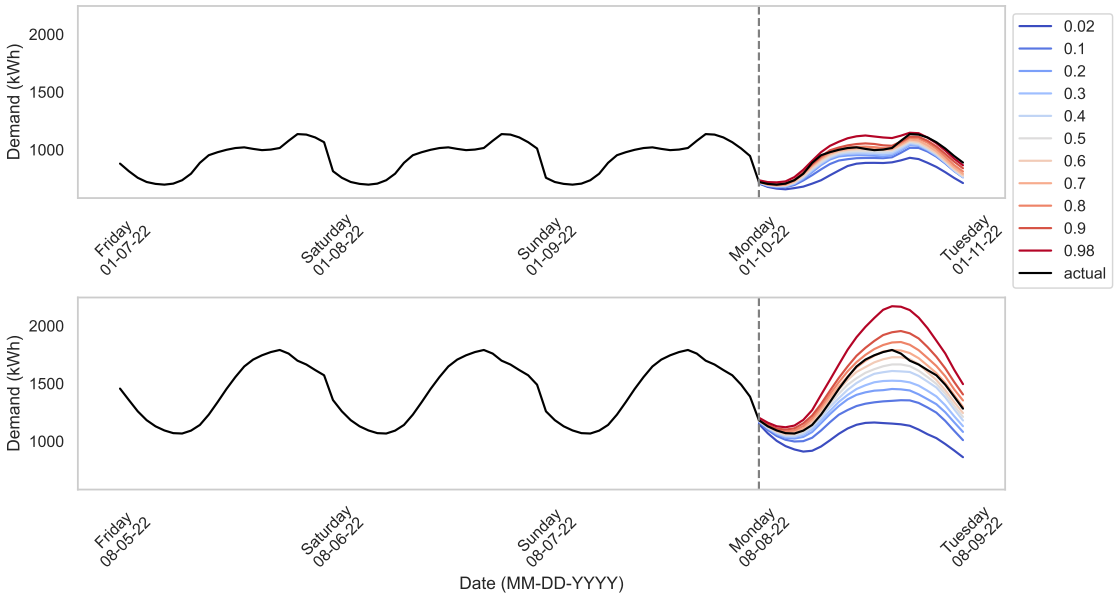

Figure 9: Marginal quantile forecasts using full features. *Top.* Winter. *Bottom.* Summer.

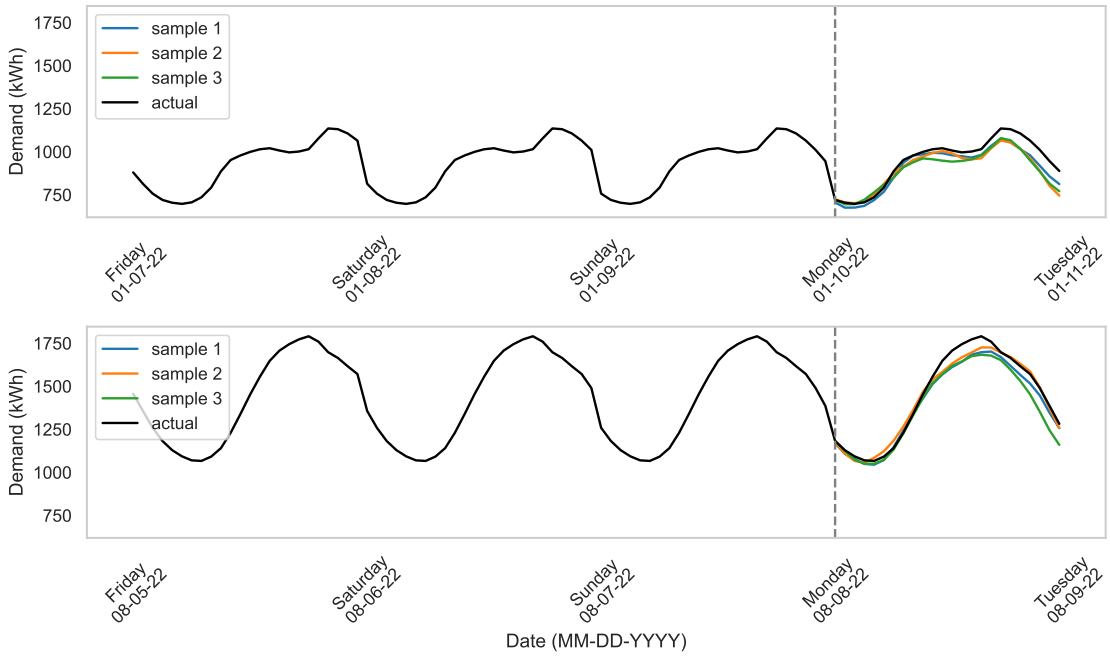

Figure 10: Conditional generated samples. *Top.* Winter. *Bottom.* Summer.

4.5 Generating conditional samples

We used the same hyperparameters for generating conditional samples as the ones used for point forecast. Figure 10 shows 3 conditional generated samples for winter and summer. In all cases, we observe that the generated samples are similar to the actual values of the time series.

Acknowledgment

This material is based on work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) under the Solar Energy Technologies Office Award Number 38529. Stephen Boyd's work was funded in part by the AI Chip Center for Emerging Smart Systems (ACCESS).

References

[1] J. Lamarck, *Annuaire météorologique pour l'an xi, n 4*, 1803.

[2] C. Hallenbeck, "Forecasting precipitation in percentages of probability," *Monthly Weather Review*, vol. 48, 11 1920, ISSN: 0027-0644.

[3] G. Brier, "Verification of forecasts expressed in terms of probability," *Monthly Weather Review*, vol. 78, 1 1950, ISSN: 0027-0644.

[4] I. Good, "Rational decisions," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 14, pp. 107–114, 1952.

[5] E. Parzen, "On estimation of a probability density function and model," *The Annals of Mathematical Statistics*, vol. 33, 3 1962, ISSN: 0003-4851.

[6] E. Epstein, "A scoring system for probability forecasts of ranked categories," *Journal of Applied Meteorology*, vol. 8, 6 1969, ISSN: 0021-8952.

[7] O. Neugebauer, *The Exact Sciences in Antiquity* (Acta historica scientiarum naturalium et medicinalium). Dover Publications, 1969, ISBN: 9780486223322.

[8] J. Matheson and R. Winkler, "Scoring rules for continuous probability distributions," *Management Science*, vol. 22, 10 1976, ISSN: 00251909.

[9] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, 1 1977, ISSN: 0035-9246.

[10] R. Koenker and G. Bassett, "Regression quantiles," *Econometrica*, vol. 46, p. 33, 1 1978.

[11] M. Rosenblatt-Roth and B. de Finetti, "Probability, induction and statistics. the art of guessing," *International Statistical Review / Revue Internationale de Statistique*, vol. 47, 1 1979, ISSN: 03067734.

[12] G. Bassett and R. Koenker, "An empirical quantile function for linear models with iid errors," *Journal of the American Statistical Association*, vol. 77, 378 1982, ISSN: 1537274X.

[13] S. Stigler, "Studies in the history of probability and statistics XL Boscovich, Simpson and a 1760 manuscript note on fitting a linear relation," *Biometrika*, vol. 71, no. 3, pp. 615–620, 1984.

[14] G. Irwin, W. Monteith, and W. Beattie, "Statistical electricity demand modelling from consumer billing data," *IEE Proceedings C Generation, Transmission and Distribution*, vol. 133, 6 1986, ISSN: 01437046.

[15] D. MacKenzie and S. Stigler, "The history of statistics: The measurement of uncertainty before 1900," *Technology and Culture*, vol. 29, 2 1988, ISSN: 0040165X.

[16] R. Herman and J. Kritzinger, "The statistical description of grouped domestic electrical load currents," *Electric Power Systems Research*, vol. 27, 1 1993, ISSN: 03787796.

[17] A. Ghosh, D. Lubkeman, M. Downey, and R. Jones, "Distribution circuit state estimation using a probabilistic approach," *IEEE Transactions on Power Systems*, vol. 12, 1 1997, ISSN: 08858950.

[18] H. Hersbach, "Decomposition of the continuous ranked probability score for ensemble prediction systems," *Weather and Forecasting*, vol. 15, 5 2000, ISSN: 08828156.

[19] A. McNeil and R. Frey, "Estimation of tail-related risk measures for heteroscedastic financial time series: An extreme value approach," *Journal of Empirical Finance*, vol. 7, 3-4 2000, ISSN: 09275398.

[20] A. Timmermann, "Density forecasting in economics and finance," *Journal of Forecasting*, vol. 19, 4 2000, ISSN: 0277-6693.

[21] S. Heunis and R. Herman, "A probabilistic model for residential consumer loads," *IEEE Transactions on Power Systems*, vol. 17, 3 2002, ISSN: 08858950.

[22] R. Koenker, *Quantile regression*. 2005.

[23] N. Meinshausen, "Quantile regression forests," *Journal of Machine Learning Research*, vol. 7, 2006, ISSN: 15337928.

[24] I. Takeuchi, Q. Le, T. Sears, and A. Smola, "Nonparametric quantile estimation," *Journal of Machine Learning Research*, vol. 7, 2006, ISSN: 15337928.

[25] V. Chernozhukov, I. Fernández-Val, and A. Galichon, "Improving point and interval estimators of monotone functions by rearrangement," *Biometrika*, vol. 96, 3 2009, ISSN: 00063444.

[26] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning, second edition," *Springer New York, NY*, vol. 27, 2 2009, ISSN: 0172-7397.

[27] V. Chernozhukov, I. Fernández-Val, and A. Galichon, "Quantile and probability curves without crossing," *Econometrica*, vol. 78, 3 2010, ISSN: 0012-9682.

[28] R. Singh, B. Pal, and R. Jabr, "Statistical representation of distribution system loads using gaussian mixture model," *IEEE Transactions on Power Systems*, vol. 25, 1 2010, ISSN: 08858950.

[29] J. Dalton, *Meteorological observations and essays*. 2011.

[30] M. El-Hawary, "The smart grid - state-of-the-art and future trends," *Electric Power Components and Systems*, vol. 42, 3-4 2014, ISSN: 15325008.

[31] S. Haben and G. Giasemidis, "A hybrid model of kernel density estimation and quantile regression for gefcom2014 probabilistic load forecasting," *International Journal of Forecasting*, vol. 32, 3 2016, ISSN: 01692070.

[32] Y. He, Q. Xu, J. Wan, and S. Yang, "Short-term power load probability density forecasting based on quantile regression neural network and triangle kernel function," *Energy*, vol. 114, 2016, ISSN: 03605442.

[33] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *International Journal of Forecasting*, vol. 32, 3 2016, ISSN: 01692070.

[34] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *International Journal of Forecasting*, vol. 32, 3 2016, ISSN: 01692070.

[35] S. Taieb, R. Huser, R. Hyndman, and M. Genton, "Forecasting uncertainty in electricity smart meter data by boosting additive quantile regression," *IEEE Transactions on Smart Grid*, vol. 7, 5 2016, ISSN: 19493053.

[36] M. Udell, C. Horn, R. Zadeh, and S. Boyd, "Generalized low rank models," *Foundations and Trends in Machine Learning*, vol. 9, 1 2016, ISSN: 19358245.

[37] R. Koenker, "Quantile regression: 40 years on," *Annual Review of Economics*, vol. 9, pp. 155–176, 1 Aug. 2017.

[38] B. Liu, J. Nowotarski, T. Hong, and R. Weron, "Probabilistic load forecasting via quantile regression averaging on sister forecasts," *IEEE Transactions on Smart Grid*, vol. 8, 2 2017, ISSN: 19493053.

[39] M. Wytock, N. Moehle, and S. Boyd, "Dynamic energy management with scenario-based robust mpc," 2017.

[40] D. Gan, Y. Wang, S. Yang, and C. Kang, "Embedding based quantile regression neural network for probabilistic load forecasting," *Journal of Modern Power Systems and Clean Energy*, vol. 6, 2 2018, ISSN: 21965420.

[41] Y. He and Y. Zheng, "Short-term power load probability density forecasting based on yeo-johnson transformation quantile regression and gaussian kernel function," *Energy*, vol. 154, 2018, ISSN: 03605442.

[42] D. van Ravenzwaaij, P. Cassey, and S. Brown, "A simple introduction to markov chain monte–carlo sampling," *Psychonomic Bulletin and Review*, vol. 25, 1 2018, ISSN: 15315320.

[43] J. Thorey, C. Chaussin, and V. Mallet, "Ensemble forecast of photovoltaic power with online crps learning," *International Journal of Forecasting*, vol. 34, 4 2018, ISSN: 01692070.

[44] Y. Yang, S. Li, W. Li, and M. Qu, "Power load probability density forecasting using gaussian process quantile regression," *Applied Energy*, vol. 213, 2018, ISSN: 03062619.

[45] M. Zamo and P. Naveau, "Estimation of the continuous ranked probability score with limited information and applications to ensemble weather forecasts," *Mathematical Geosciences*, vol. 50, 2 2018, ISSN: 18748953.

[46] N. Moehle, E. Busseti, S. Boyd, and M. Wytock, "Dynamic energy management," *Large Scale Optimization in Supply Chains and Smart Manufacturing: Theory and Applications*, pp. 69–126, 2019.

[47] N. Sammaknejad, Y. Zhao, and B. Huang, *A review of the expectation maximization algorithm in data-driven process identification*, 2019.

[48] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," Association for Computational Linguistics, 2019, pp. 3645–3650.

[49] W. Zhang, H. Quan, and D. Srinivasan, "An improved quantile regression neural network for probabilistic load forecasting," *IEEE Transactions on Smart Grid*, vol. 10, 4 2019, ISSN: 19493053.

[50] A. Borning, B. Friedman, and N. Logler, "The 'invisible' materiality of information technology," *Communications of the ACM*, vol. 63, pp. 57–64, 6 May 2020.

[51] S. Impram, S. Nese, S. Varbak, and B. Oral, "Challenges of renewable energy penetration on power system flexibility: A survey," *Energy Strategy Reviews*, vol. 31, p. 100 539, 2020.

[52] S. Zhang, Y. Wang, Y. Zhang, D. Wang, and N. Zhang, "Load probability density forecasting by transforming and combining quantile forecasts," *Applied Energy*, vol. 277, 2020, ISSN: 03062619.

[53] T. Emmanuel, T. Maupong, D. Mpoeleng, T. Semong, B. Mphago, and O. Tabona, "A survey on missing data in machine learning," *Journal of Big Data*, vol. 8, 1 2021, ISSN: 21961115.

[54] G. Jones and Q. Qin, "Markov chain monte carlo in practice," *Annual Review of Statistics and Its Application*, vol. 9, 2022, ISSN: 2326831X.

[55] DOE, *American-made net load forecasting prize*, Accessed: 2024-01-25, 2023. [Online]. Available: https://www.energy.gov/eere/solar/american-made-net-load-forecasting-prize.

[56] ISONE. "Energy, load, and demand reports." (2023), [Online]. Available: https://www.iso-ne.com/isoexpress/web/reports/load-and-demand/-/tree/whlsecost-hourly-rhodeisland (visited on 10/25/2023).

[57] V. Ivchenko, "A gentle introduction to quasi-periodic phenomena," *Resonance*, vol. 28, 7 2023, ISSN: 0973712X.

[58] A. Luccioni and A. Hernandez-Garcia, "Counting carbon: A survey of factors influencing the emissions of machine learning," *ArXiv Preprint*, Feb. 2023.

[59] A. de Vries, "The growing energy footprint of artificial intelligence," *Joule*, vol. 7, pp. 2191–2194, 10 Oct. 2023.

[60] D. Widder, S. West, and M. Whittaker, "Open (for business): Big tech, concentrated power, and the political economy of open AI," *SSRN Electronic Journal*, 2023.

[61] E. Tuzhilina, T. Hastie, D. McDonald, J. Tay, and R. Tibshirani, "Smooth multi-period forecasting with application to prediction of covid-19 cases," *Journal of Computational and Graphical Statistics*, vol. 0, no. 0, pp. 1–13, 2024.