Financial Engineering AY 2023/2024

Electricity Price and Load Forecasting

Readings prelab

Alessandro Brusaferri[†] & Roberto Baviera[‡]

April 8, 2024

(†) Cnr - STIIMA, 12 via Alfonso Corti, Milano
(‡) Politecnico di Milano, Department of Mathematics, 32 p.zza L. da Vinci, Milano

1 EPLF benchmarks

Several methods have been proposed within the wide research literature on Electricity Price and Load Forecasting (EPLF). See, e.g., reviews in Weron (2014) and Hong and Fan (2016). Here we provide a brief introduction to two common benchmarks, namely DNN and LEAR (see, e.g., Lago *et al.* 2018, 2021, for detailed descriptions and experimental comparisons).
Both approaches can be represented as parameterized functions aimed to map the input variables set to the day-ahead predictions over the horizon $h = 1, ..., H$ (e.g., next 24 hours):

$$\hat{y}_{t+h} = f_{\Omega}\left(y_{t-k:t}, z_{t-k:t}, x_{t+h}\right) \tag{1}$$

where Ω is the parameters' set of the model, and k is the maximum lag involved in the observed history of each input series. Often, the conditioning variable set in the right-hand side of (1) comprises the past values of the target price $y_{t-k:t}$, which is available till the current day, as well as further exogenous features.[1] The exogenous set can include both observations from the past days $z_{t-k:t}$ (e.g., the previous electricity demands) and predicted variables x_{t+h}, such as the electricity load forecast, renewable generation predictions, etc.
For example, the DNN benchmark implements $f_{\Omega}(.)$ through a feed-forward neural network architecture. Considering two hidden layers of n_{h_1}, n_{h_2} units, and summarizing the input features available at stage t as a flattened vector x_t of size n_d to lighten notation, it is expressed as:

$$\begin{cases} a_i^{(1)} &= g_i^{(1)}\left(\sum_{d=1}^{n_d} \omega_{d,i}^{(1)} \mathrm{x}_{t,d} + \omega_{0,i}^{(1)}\right) & \text{first hidden layer} \\ a_j^{(2)} &= g_j^{(2)}\left(\sum_{i=1}^{n_{h_1}} \omega_{i,j}^{(2)} a_i^{(1)} + \omega_{0,j}^{(2)}\right) & \text{second hidden layer} \\ \hat{y}_{t+h} &= \sum_{j=1}^{n_{h_2}} \omega_j^{(o)} a_j^{(2)} + \omega_0^{(o)} & \text{output layer} \end{cases} \tag{2}$$

[1] In these notes no masking is assumed, i.e. we suppose that all data are available up to time t.

where $\omega_{d,i}^{(1)} \in \mathbb{R}^{n_d \times n_{h_1}}$, $\omega_{i,j}^{(2)} \in \mathbb{R}^{n_{h_1} \times n_{h_2}}$, $\omega_j^{(o)} \in \mathbb{R}^{n_{h_2}}$, $\omega_{0,i}^{(1)} \in \mathbb{R}^{n_{h_1}}$, $\omega_{0,j}^{(2)} \in \mathbb{R}^{n_{h_2}}$, $\omega_0^{(o)} \in \mathbb{R}$ represent the network weights and biases. $g_i^{(1)}$ and $g_j^{(l)}$ define the activation functions of the hidden layers. Most popular choices include sigmoid, ELU, GELU, RELU and softplus (see Lago *et al.* (2021)).
Beside the simple single step ($h = 1$) formulation reported in (2), the DNN can be straightly extended to a multi-step architecture ($h > 1$), by mapping the whole prediction horizon (see Lago *et al.* (2021) for more details).

The LEAR benchmark implements $f_\Omega(.)$ as a simple affine mapping, as in the output layer of the DNN. This model falls within the class of AutoRegressive eXogenous (ARX) linear models: they allow to use standard statistical tools, including partial autocorrelation function (PACF) evaluation and collinearity.
The linear and non-linear benchmarks are only the two extreme possibilities. Often, when a significant seasonality is observed –as in the case of household load– hybrid approaches can lead to good performances, especially in probabilistic *ex-post* forecasts (see, e.g., Baviera and Messuti 2023, Baviera and Azzone 2021, Baviera and Manzoni 2022). In the hybrid case, an Ordinary Least Square is followed by non linear techniques.

For both DNN and LEAR benchmarks, a first task is the selection of input features in the whole potential set (e.g. specific lags may be chosen): this can be done either by means of manual or automatic techniques. In the former case, a detailed data analysis must be performed, considering also in the non-linear case the standard tools for linear models (PACS and collinearity) or expert judgement (see, e.g. Baviera and Messuti 2023, for an implementation in a non linear load forecasting model). In the latter case, features are treated as hyper-parameters, as detailed in section 2.3.

2 Model parameters vs hyper-parameters

The scope of the training procedure is to estimate the model parameters (weights and biases in the DNN) minimizing the target loss function (e.g., prediction error). For neural networks, this is often performed by means of stochastic gradient descent and related extensions (see e.g., Goodfellow *et al.* (2016)-Ch8 for details). Still, additional configurations need to be specified to complete the learning framework, i.e., hyper-parameters. For DNNs these include, e.g., the number of hidden units and the activation function in the network, the optimizer class and the related learning-rate. In electricity price and load forecasting practice, hyper-parameters are often tuned by means of cross-validation (CV). Specifically, the available dataset is split in three parts:

- training set: used to estimate the model parameters through the learning algorithm;
- validation set: use to set the best value of the hyper-parameters by assessing the performance of the model on samples not accessible during training;
- test set: to evaluate the final model performance in out-of-samples conditions.

See Goodfellow *et al.* (2016) for a detailed description.

2.1 Parameters: Overfitting and early-stopping

Model calibration on the training set selects the set of parameters for a given configuration set (i.e. fixing the hyper-parameters).
Flexible models such as the feedforward network introduced in Section 1 are prone to overfitting. Overfitting occurs as training proceeds, when the model begins to extract part of the residual noise as representing the structure of the underlying system, i.e., to memorize the characteristics of the finite training samples, thus leading to poor generalization. Early-stopping is a simple technique to mitigate such issue. The model performances are assessed on a validation subset after each epoch (a training run in a neural network). The training procedure is stopped when the validation score does not decrease, which indicates that the model is losing generalization capabilities. See James *et al.* (2014)-Ch10 for more details on early stopping, as well as further regularization techniques to prevent overfitting.

2.2 Hyper-parameters

The model is repeatedly trained under different hyper-parameters configurations. The setup achieving the best performance on the validation set is adopted for the final test. Multiple techniques can be employed to explore the space of the possible hyper-parameter choices, from simple grid search procedures to advanced Bayesian optimization (see, e.g., Watanabe 2023). See James *et al.* (2014)-Ch5.1 for further details on cross validation and related extensions (e.g. K-fold CV).
Time series CV can be rather different. Two are the main criticalities: the limited dataset and the relevance of the latest pieces of information.

2.3 Features selection as hyper-parameters

In the DNN benchmark (see Lago *et al.* (2021)), the authors include the feature selection process within the hyper-parameter search. To this end, boolean indicator variables are associated to each lag, leaving to the search procedure the choice to include/exclude each feature from the final model, depending on the impact on the validation performance.

3 Loss functions

In this section, we introduce the most used loss functions in EPLF, dividing them into two natural categories: the ones utilised in point forecasting and the ones designed for probabilistic forecasting. The choice of the loss function represents a crucial moment in training a forecasting model, as it determines what the model should consider as a *good* forecast. The selection of the most appropriate loss function depends on the specific problem and objective, and it plays a crucial role in designing a well-performing forecaster. In this section, we introduce the most commonly used loss functions in EPLF, dividing them into two natural categories: those utilized in point forecasting and those designed for probabilistic forecasting

3.1 Point forecasting

In point forecasting, the primary objective is to generate forecasts that closely match observed values. Measures of absolute and relative distance between the forecasted and observed values are typically employed as loss functions. Additionally, supplementary terms are often included to address the overfitting problem.

3.1.1 Mean Squared Error

The Mean Squared Error (MSE) is the standard loss function considered for training point forecasters. Formally, it is defined as:

$$\text{MSE} := \frac{1}{N}\sum_{n=1}^{N}(\hat{y}_n - \mathrm{y}_n)^2 \tag{3}$$

where $\hat{y}_n$ and y_n represent the model output and true value for the n-th sample in the dataset, denoted as $\mathcal{D} = \{\mathrm{x}_n, \mathrm{y}_n\}_{n=1}^{N}$. Here, $\mathcal{D}$ comprises $N = N_d \times H$ realizations of the input features and dependent variables pairs, with N_d the number of days involved in the assessed dataset.
In electricity price and load forecasting applications, the samples are typically built from the time series by means of sliding window techniques (see Lago *et al.* 2021).

3.1.2 LASSO, Ridge and Elastic Net

In section 1, we briefly introduced the LEAR model as a simple affine map from the conditioning set, thus following the conventional AutoRegressive eXogenous (ARX) form (see Lago *et al.* (2021)). Still, the LEAR model include a further backbone components, namely a LASSO regularizer. Specifically, LASSO adds to the usual regression objective functions (e.g., sum of squared errors) the $\ell 1$-norm of the model parameters vector, weighted by a tunable scalar λ:

$$\underset{\Omega}{min} \sum_{n=1}^{N}\left(f_\Omega(\mathrm{x}_n) - \mathrm{y}_n\right)^2 + \lambda\sum_{j=1}^{\#\Omega}|\omega_j| \tag{4}$$

In practice, the introduction of the $\ell 1$-norm in the regressors tends to privilege sparse solutions, i.e. characterized by numerous zero-valued parameters. Therefore, the shrinkage factor, when applied to the coefficients related to the input variables, de-facto introduces a feature selection mechanism across the lags of the multi-input series. Besides, it provides a natural form of regularization, reducing model complexity and potential over-fitting issues. In practice, the feature extraction and regularization mechanism must be configured by tuning the penalty term; to such an aim, cross-validation techniques provide a valuable option.
The replacement of the $\ell 1$-norm with a $\ell 2$-norm in the loss (4) lead to a regularization method known as Ridge regression. Moreover, both techniques can be combined, leading to the Elastic Net method:

$$\underset{\Omega}{min} \sum_{n=1}^{N}\left(f_\Omega(\mathrm{x}_n) - \mathrm{y}_n\right)^2 + \lambda\sum_{j=1}^{\#\Omega}|\omega_j| + \gamma\sum_{j=1}^{\#\Omega}\omega_j^2 \tag{5}$$

3.1.3 Mean Absolute Percentage Error

Another loss function commonly encountered in the point forecasting literature is the Mean Absolute Percentage Error (MAPE; see, e.g., Zhang *et al.* 2022). This loss function considers the absolute relative prediction errors and is defined as:

$$\text{MAPE} := \frac{1}{N}\sum_{n=1}^{N}\left|\frac{\hat{y}_n - \mathrm{y}_n}{\mathrm{y}_n}\right| \tag{6}$$

and – as suggested by its name – it is often reported as a percentage value.
For practical applications, one advantage of MAPE is that it represents a weighted average of the prediction error, where each weight corresponds to the inverse of the absolute observed value y_n. This property makes MAPE robust in the presence of sudden spikes, which are typically encountered in electricity markets. However, this beneficial property comes with a trade-off: MAPE may not be the best choice when the dataset contains prices and loads close to zero.

3.2 Probabilistic forecasting

The main challenge of probabilistic forecasting is that, while forecasts are provided in terms of intervals or probability densities, we never observe the true distribution of the underlying process, but only point realizations. Therefore, loss functions should be designed to assess distances between realizations and predicted quantiles, or realizations and predicted distributions.

3.2.1 Pinball loss

The Pinball Loss is a measure of fit for quantiles, which is typically employed for quantile regression, i.e. for fitting a model that predicts a specific quantile (e.g. the median, or the 95%-quantile).
Given a confidence level $\alpha \in (0, 1)$, a forecast $\hat{q}_{\alpha,t}$ for the α-quantile of the target variable for the n-th sample and the corresponding realisation y_n, the associated Pinball Loss is defined as

$$\mathcal{P}(\alpha, \hat{q}_{\alpha,n}, \mathrm{y}_n) := \alpha[\mathrm{y}_n - \hat{q}_{\alpha,n}]\mathbb{1}(\mathrm{y}_n \geq \hat{q}_{\alpha,n}) + (1-\alpha)[\hat{q}_{\alpha,n} - \mathrm{y}_n]\mathbb{1}(\mathrm{y}_n < \hat{q}_{\alpha,n}) \tag{7}$$

where $\mathbb{1}(\cdot)$ denotes the indicator function.
Training an ensemble of models, each one responsible for predicting a specific quantile, is a fairly common practice in EPLF, which has in some cases proven to be a convincing forecasting choice (see, e.g., Ziel 2019). In this case, each model is generally trained using the mean of the Pinball Loss over the samples, i.e.,

$$\text{Pinball}(\alpha) := \frac{1}{N}\sum_{n=1}^{N}\mathcal{P}(\alpha, \hat{q}_{\alpha,t}, \mathrm{y}_t) \tag{8}$$

3.2.2 Negative Log-Likelihood

A different forecasting paradigm in probabilistic forecasting is the one of density forecasting. In this case, a forecasters predicts a vector of parameters that characterise the

density of the inferred distribution. An example of forecaster of this kind is a model that predicts the mean $\hat{\mu}_n$ and the standard deviation $\hat{\sigma}_n$ of a Gaussian distribution. In this case, the output of the model is a distribution $\hat{f}_n$, whose distance from the realisation y_n should be conveniently defined.
In this framework, the well-known Maximum Likelihood Estimation approach can be translated in terms of loss function by introducing the Negative Log-Likelihood. Under the assumption of independent realisations (which should be properly tested and justified in a time-series framework), this loss function is defined as

$$\mathcal{L} := -\sum_{n=1}^{N} \log \hat{f}_n(\mathrm{y}_n)\,. \tag{9}$$

In the Gaussian case, the Negative Log-Likelihood assumes the well-known form

$$\mathcal{L} := \frac{1}{2}\sum_{n=1}^{N}\left[\log(2\pi\hat{\sigma}_n^2) + \left(\frac{\mathrm{y}-\hat{\mu}_n}{\hat{\sigma}_n}\right)^2\right]. \tag{10}$$

It is worth remarking that the MSE loss function in (3) can be obtained as a sub-case of (10), in which the standard deviation σ_n is assumed to be known and constant in time.

3.2.3 Average Pinball Loss

The Pinball Loss presented in (7) can be adapted to become a loss function suitable for density forecasting. In this case, one aims to minimize (on average) the Pinball Loss $\mathcal{P}(\alpha, \hat{q}_{\alpha,n}, \mathrm{y}_n)$ for the n-th sample over a set of quantiles (e.g., all 99 percentiles).
The obtained loss function is called Average Pinball Loss (APL), and is defined as follows:

$$\mathrm{APL} := \frac{1}{99}\sum_{\alpha=1\%}^{99\%} \mathrm{Pinball}\alpha\,. \tag{11}$$

3.2.4 Continuous Ranked Probability Score

The last loss function we consider is called Continuous Ranked Probability Score (CRPS) and is a special case of the general energy score (cf. Gneiting and Raftery 2007). This loss function enjoys various appealing features, such as robustness and sensitivity to distances, while rewarding densities around the realizations. See Gneiting and Raftery (2007) for a more detailed review and analysis of the mathematical properties.
Formally, given a forecast $\hat{F}_n$ for the n-th cumulative distribution function (CDF), the associated CRPS is defined as follows:

$$\mathrm{CRPS}(\hat{F}_n, \mathbf{y}_n) = \int_{-\infty}^{+\infty}\left[\hat{F}_n(z) - \mathbb{1}\left(z \geq \mathrm{y}_n\right)\right]^2 \mathrm{d}z \tag{12}$$

where $\mathbb{1}(\cdot)$ represents the indicator function. Under finite first moment of $P(\mathbf{y})$, the CRPS can be expressed in the form:

$$\mathrm{CRPS}(\hat{F}_n, \mathrm{y}_n) = \mathbb{E}_{\hat{F}_n}|y - \mathrm{y}_n| - \frac{1}{2}\mathbb{E}_{\hat{F}_n,\hat{F}_n}|y - y'| \tag{13}$$

given independent samples y, y' from the distribution $\hat{F}_n$.
Then, by exploiting the empirical approximation to the predictive distribution, the CRPS can be numerically computed over each target sample through:

$$\mathrm{CRPS}_N = \frac{1}{N}\sum_{n=1}^{N}\left[\frac{1}{m}\sum_{i=1}^{m}|\hat{y}_n^{(i)} - \mathrm{y}_n| - \frac{1}{2m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}|\hat{y}_n^{(i)} - \hat{y}_n^{(j)}|\right] \tag{14}$$

where m, N represents the number of the samples from the predictive distribution and the target dataset size respectively.
It is worth noting that the APL in (11), computed over the distribution percentiles, is commonly employed in electricity price and load forecasting as a cheap discrete approximation to the CRPS (see, e.g., Marcjasz *et al.* 2023).

4 Test set evaluation

Loss functions can be used as scoring functions in the evaluation process on the test set and vice-versa. This section reminds this important concept and introduces some scoring functions that are often found in the literature only as scoring functions. Two main classes of scoring functions exist: the ones on point prediction performance evaluation and the ones on probabilistic prediction performance. The former focuses on the quality of my prediction compared to the (actual) realized quantity (either load or price in this lab). The latter tries to evaluate the distributional properties of my forecast via two main evaluation metrics: sharpness and reliability (see, e.g., Nowotarski and Weron (2018)). Sharpness verifies that the forecast is as tight as possible around the expected value; reliability attests distribution's statistical significance, it is the equivalent of unconditional backtesting in finance (Kupiec 1995) and it is the most important evaluation method for probabilistic forecasting.

4.1 Point prediction performance

To evaluate the point prediction accuracy, several indicators are often considered in the literature, including: Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE) and the Symmetric Mean Absolute Percentage Error (sMAPE).

- root mean square error (RMSE):

$$\mathrm{RMSE} = \left[\frac{1}{N}\sum_{n=1}^{N}(\mathrm{y}_n - \hat{y}_n)^2\right]^{\frac{1}{2}} \tag{15}$$

- mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{N}\sum_{n=1}^{N}|\mathrm{y}_n - \hat{y}_n| \tag{16}$$

- symmetric mean absolute percentage error (sMAPE):

$$\mathrm{sMAPE} = \frac{1}{N}\sum_{n=1}^{N}\frac{|\mathrm{y}_n - \hat{y}_n|}{0.5\,(|\mathrm{y}_n| + |\hat{y}_n|)} \tag{17}$$

- relative mean absolute error (rMAE):

$$\text{rMAE} = \frac{\frac{1}{N}\sum_{n=1}^{N}|\text{y}_n - \hat{y}_n|}{\frac{1}{N}\sum_{n=1}^{N}|\text{y}_n - \hat{y}_n^{naive}|} \tag{18}$$

RMSE expresses the squared root of the second sample moment of the residuals, but tend to be sensitive to large errors and outliers. Mean Absolute Error provides a different view, influenced in proportion to the absolute value of prediction errors. The Mean Absolute Percentage Error (MAPE) is considered in some works in the literature, to provide a relative view on the residuals, since absolute error indicators might be difficult to be compared over different studies (e.g., due to different rescaling). However, MAPE can be distorted by small values, becoming very large regardless the actual value of the residuals. Moreover, MAPE values turns out to be small when processing higher values, irrespective of the absolute errors. sMAPE extends MAPE by introducing lower and upper bounds. Conventional sMAPE formula expresses results in a range 0–200% which could be misleading to be interpreted. The alternative formulation, reported in (17), applies a 0.5 scaling factor within the denominator, providing more intuitive indications. Moreover, authors in Lago *et al.* (2021) advocate for the integration of rMAE, as it provides a more reliable relative metric to support fair cross-dataset comparison between studies. This indicator scales the absolute error with reference to a naive forecaster (e.g., the true value observed in the same hour of the previous day).

4.2 Probabilistic forecasting evaluation

4.2.1 Winkler's score

The Winkler's score (or interval score) is a proper scoring rule (see, e.g., Gneiting and Raftery 2007) to assess probabilistic forecasts formed as Prediction Intervals (PI) at discrete coverage levels $1-\alpha$ (see Nowotarski and Weron (2018)). Formally, it is defined as:

$$Winkler_n = \begin{cases} \delta_n, & \text{if: } \text{y}_n \in [\hat{L}_n, \hat{U}_n] \\ \delta_n + \dfrac{2}{1-\alpha}(\hat{L}_n - \text{y}_n), & \text{if: } \text{y}_n < \hat{L}_n \\ \delta_n + \dfrac{2}{1-\alpha}(\text{y}_n - \hat{U}_n), & \text{if: } \text{y}_n > \hat{U}_n \end{cases} \tag{19}$$

where $\delta_n = \hat{U}_n - \hat{L}_n$ is the width the α-PI, with $\hat{L}_n, \hat{U}_n$ lower and upper bounds respectively. The first case in (19) rewards narrow PIs (i.e., sharpness), while the others penalizes the occurrence of test observations outside the predicted interval.

4.2.2 Continuous Ranked Probability Score

Reliable probabilistic forecasting systems have to maximize sharpness subject to reliability. Various summary measurements - unifying both aspects - have been proposed to correctly rank probabilistic forecasters (Nowotarski and Weron 2018). In particular, the Continuous Ranked Probability Score (CRPS) is broadly adopted as a de-facto standard to assess distributional forecasting systems (see, e.g., Hong and Fan 2016, and references therein).

4.2.3 Delta Empirical Coverage

The empirical coverage has been introduced by Kupiec (1995) in the backtesting of VaR. *Reliability* is the most important evaluation method for probabilistic forecasting. It refers to the statistical consistency between the probabilistic forecasts and the realized observations OS in the test set. In practice, it determines the fraction of observations that fall outside the confidence interval (CI) with a given nominal level α; e.g., if the fraction of the realized daily power consumptions, that falls within the 90% CI, is close to 90% then this CI is said to be reliable.
More in detail. Let $\hat{L}_t$ and $\hat{U}_t$ be, respectively, the lower and upper bounds for a given (central) α CI, where α is the CI nominal level, and y_t the actual consumption at time t, the indicator I_t takes two values: 1 if the actual consumption falls within the forecasted CI and zero otherwise, i.e.

$$I_t = \begin{cases} 1 & \text{if } y_t \in [\hat{L}_t, \hat{U}_t] \quad \text{“hit”} \\ 0 & \text{if } y_t \notin [\hat{L}_t, \hat{U}_t] \quad \text{“violation”} \end{cases} .$$

The empirical coverage EC_α is the Out-Sample mean of the indicator for a given nominal level α. Qualitatively, the closer is the empirical coverage to the nominal level, the better it is. In particular, it is relevant the difference between the nominal level and the empirical coverage for several values of α relative to the tails of distribution.
The Delta Coverage is defined as the average over the percentiles between α_{min} and α_{Max} of the absolute value of this difference

$$\Delta C := \frac{1}{100 * (\alpha_{Max} - \alpha_{min})} \sum_{a=100*\alpha_{min}}^{100*\alpha_{Max}} |EC_a - a| \tag{20}$$

where often the interval is chosen as the set of percentiles between $\alpha_{min} = 90\%$ and $\alpha_{Max} = 99\%$.

Abbreviations

APL	Average Pinball Loss
ARX	AutoRegressive eXogenous linear models
CRPS	Continuous Ranked Probability Score (example of Probabilistic Evaluation)
CV	cross-validation
DNN	Deep Neural Network
LEAR	LASSO-Estimated AutoregrìRessive model
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
PACF	Partial AutoCorrelation Function
PI	Prediction Intervals
RMSE	Root Mean Squared Error (example of Point Evaluation)

References

Baviera, R. and Azzone, M., 2021. Neural network middle-term probabilistic forecasting of daily power consumption, *The Journal of Energy Markets*, 14 (1), 1–26.

Baviera, R. and Manzoni, P., 2022. Tree-based learning in rnns for power consumption forecasting, *arXiv preprint arXiv:2209.01378*.

Baviera, R. and Messuti, G., 2023. Daily middle-term probabilistic forecasting of power consumption in north-east england, *Energy Systems*, 1–23.

Gneiting, T. and Raftery, A.E., 2007. Strictly proper scoring rules, prediction, and estimation, *Journal of the American Statistical Association*, 102 (477), 359–378.

Goodfellow, I., Bengio, Y., and Courville, A., 2016. *Deep Learning*, MIT press.

Hong, T. and Fan, S., 2016. Probabilistic electric load forecasting: A tutorial review, *International Journal of Forecasting*, 32 (3), 914–938.

James, G., Witten, D., Hastie, T., and Tibshirani, R., 2014. *An Introduction to Statistical Learning: with Applications in R*, Springer.

Kupiec, P., 1995. Techniques for verifying the accuracy of risk measurement models, *The Journal of Derivatives*, 3 (2), 73–84.

Lago, J., De Ridder, F., and De Schutter, B., 2018. Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms, *Applied Energy*, 221, 386–405.

Lago, J., Marcjasz, G., De Schutter, B., and Weron, R., 2021. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark, *Applied Energy*, 293, 116983.

Marcjasz, G., Narajewski, M., Weron, R., and Ziel, F., 2023. Distributional neural networks for electricity price forecasting, *Energy Economics*, 125, 106843.

Nowotarski, J. and Weron, R., 2018. Recent advances in electricity price forecasting: A review of probabilistic forecasting, *Renewable and Sustainable Energy Reviews*, 81, 1548–1568.

Watanabe, S., 2023. Tree-structured parzen estimator: Understanding its algorithm components and their roles for better empirical performance, *arXiv preprint arXiv:2304.11127*.

Weron, R., 2014. Electricity price forecasting: A review of the state-of-the-art with a look into the future, *International Journal of Forecasting*, 30 (4), 1030–1081.

Zhang, J., Wang, Y., and Hug, G., 2022. Cost-oriented load forecasting, *Electric Power Systems Research*, 205, 107723.

Ziel, F., 2019. Quantile regression for the qualifying match of GEFCom2017 probabilistic load forecasting, *International Journal of Forecasting*, 35 (4), 1400–1408.

Acknowledgements

We thank P. Manzoni for fruitful discussions on this topic and a critical reading of the manuscript.