End-to-end Learning for Hierarchical Forecasting of Renewable Energy Production with Missing Values

Akylas Stratigakos, Dennis van Der Meer, Simon Camal, Georges Kariniotakis

End-to-end Learning for Hierarchical Forecasting of Renewable Energy Production with Missing Values

Akylas Stratigakos, Dennis van der Meer, Simon Camal, Georges Kariniotakis
MINES Paris, PSL University, Center PERSEE
MINES Paris, PSL University
Sophia Antipolis, France
{akylas.stratigakos, dennis.van_der_meer, simon.camal, georges.kariniotakis}@minesparis.psl.eu

***Abstract*—Power systems feature an inherent hierarchical structure. Ensuring that forecasts across a hierarchy are coherent presents an important challenge in energy forecasting. In this context, proposed reconciliation or end-to-end learning approaches assume coherent historical observations by construction; this assumption, however, is often violated in practice due to equipment failures. This paper proposes an end-to-end learning approach for hierarchical forecasting that directly handles missing values. First, we show that a class of off-the-shelf machine learning models already leads to coherent hierarchical forecasts. Next, we describe a conditional stochastic optimization approach based on prescriptive trees for end-to-end learning with missing values, that fully utilizes the available data. We validate the proposed approach in two case studies comprising 60 wind turbines and 20 photovoltaic parks, respectively. The empirical results show that end-to-end learning outperforms two-step reconciliation approaches and that the proposed solution mitigates the adverse effect of missing values.**

***Index Terms*—hierarchical forecasting, missing data, renewable energy forecasting**

I. Introduction

The ongoing digitization and decentralization of modern power systems brings a plethora of new challenges. To facilitate decision-making across various aggregation levels, forecasts of geographically distributed renewable energy sources should adhere to a set of linear aggregation constraints imposed by a hierarchical structure; this presents an increasingly important challenge in energy forecasting.

Traditional approaches for hierarchical forecasting [1], i.e., forecasting a group of time series that satisfy a set of linear aggregation constraints, include the bottom-up and the top-down. The bottom-up approach involves forecasting the bottom-level series and aggregating them; however, it usually performs poorly as the signal-to-noise ratio tends to be lower at the bottom-level series. Further, the top-down approach may introduce forecast bias. Thus, a significant body of research focuses on two-step methods, where each series is modeled independently, with individual (termed *base*) forecasts being reconciled in a post-processing step. In [2] unbiased base forecasts are reconciled by minimizing the trace of the forecast error covariance matrix. The authors of [3] propose reconciliation by weighted projection of base forecasts. A constrained multivariate regression framework is described in [4], considering both batch and online learning, applied in wind power forecasting. Moving beyond point forecasts, [5] describes a bottom-up approach for coherent probabilistic forecasting, while [6] proposes a block bootstrap method for probabilistic photovoltaic (PV) production forecasts. Recently, end-to-end learning, i.e., training a single model to predict all series in one-shot, has begun to attract attention, as it directly leverages dependencies across series. A deep learning model, involving an internal projection step, is presented in [7] for coherent probabilistic forecasts. Lastly, a general framework for end-to-end forecasting of predictive quantiles is proposed in [8].

A recurring assumption in the literature is that training observations are coherent by construction. In practice, however, missing or erroneous values are commonplace due to communication failures or equipment malfunctions. We identify two broad approaches to deal with missing values. The first is to simply ignore training observations with missing values, an approach typically applied in the works mentioned above. However, in an end-to-end learning setting with a single model predicting all of the series, disregarding observations leads to underutilizing the available data. Alternatively, missing values can be imputed; in turn, this raises the problem of ensuring that imputed values are coherent. To this end, [9] proposes an iterative algorithm to impute missing values while exploiting dependencies across series. In the present work, we examine the case of missing values in the lower levels of the hierarchy, but accurate measurements in the upper levels, which we believe to be of practical interest in power system applications. For example, smart-meters might fail to transmit consumption data at a household level while the respective distribution feeder properly measures aggregated demand.

In short, our contributions are as follows:

- First, we show that a class of non-parametric machine learning models directly derives coherent point and probabilistic hierarchical forecasts, making it possible to employ off-the-shelf tools in an end-to-end learning setting.
- Next, we describe a decision tree algorithm for end-to-end forecasts with missing values that does not require imputation and fully utilizes the available training data.
- Lastly, we validate the proposed solution in two case studies of day-ahead forecasting considering an aggregation

Parts of this research were carried in the frame of the Smart4RES project (No. 864337), supported by the Horizon 2020 Framework Program.

of 60 wind turbines and 20 PV parks, respectively.

The rest of this paper is organized as follows. Section II introduces the mathematical background and the proposed methodology. Section III presents the experimental setting and the results. We conclude and provide directions for further research in Section IV.

II. Background and Methodology

A. Hierarchical forecasting

We examine a group of time series that satisfy a set of linear aggregation constraints, thus forming a hierarchy. Let $y_{it} \in \mathbb{R}$ denote a realization of the i-th series, associated with features $\mathbf{x}_{it} \in \mathbb{R}^p$, for $t = 1, \dots, T$. We define $\mathbf{y}_t \in \mathbb{R}^n$ as the realization of the n series and $\mathbf{x}_t := [\mathbf{x}_{1t}, \dots, \mathbf{x}_{nt}] \in \mathbb{R}^{n \times p}$. Following [5], let $\mathbf{y}_t^b \in \mathbb{R}^{n_b}$ denote the bottom-level series (i.e., series at the leaf nodes), $\mathbf{y}_t^a \in \mathbb{R}^{n_a}$ denote the series formed by aggregation, with $n = n_b + n_a$, and $\mathbf{S} \in \{0,1\}^{n \times n_b}$ be an aggregation matrix. For a specific hierarchy, observations for all the levels are derived by

$$\mathbf{y}_t = \mathbf{S}\mathbf{y}_t^b \iff \begin{bmatrix} \mathbf{y}_t^a \\ \mathbf{y}_t^b \end{bmatrix} = \begin{bmatrix} \mathbf{S}_a \\ \mathbf{I}_{n_b} \end{bmatrix} \mathbf{y}_t^b, \quad \forall t \in [T] \tag{1}$$

where $\mathbf{S}_a \in \{0,1\}^{n_a \times n_b}$ aggregates the bottom-level series $\mathbf{y}_t^b$, $\mathbf{I}_{n_b}$ is an n_b-size identity matrix, and $[T] := \{1, \dots, T\}$. The notation is illustrated in Fig. 1. Following [7], a convenient way to represent (1) is given by

$$\mathbf{A}\mathbf{y}_t = \mathbf{0}, \quad \forall t \in [T], \tag{2}$$

where $\mathbf{A} = [\mathbf{I}_{n_a}, -\mathbf{S}_a]^\intercal$ and $\mathbf{I}_{n_a}$ is an n_a-size identity matrix. Let $\widehat{\mathbf{y}}_{t+k}$ denote a set of *base* point forecasts issued at time t with horizon k, i.e., independent forecasts for each series. The notion of *additive coherency* (*coherency* hereafter) is defined as follows.

Definition 1. *Forecasts $\widehat{\mathbf{y}}_{t+k}$ are said to be coherent if they satisfy $\mathbf{A}\widehat{\mathbf{y}}_{t+k} = \mathbf{0}$.*

Typically, base forecasts $\widehat{\mathbf{y}}_{t+k}$ will not satisfy the coherency constraints, thus a post-processing step is required. Let us further define $\mathcal{S} := \{\mathbf{y} \mid \mathbf{A}\mathbf{y} = \mathbf{0}\}$ to be the feasible set that satisfies the linear aggregation constraints. From (1), the following assumption is in place.

Assumption 1. *Historical observations $\mathbf{y}_t$ are coherent by construction, i.e., $\mathbf{y}_t \in \mathcal{S}$ $\forall t \in [T]$.*

This assumption is standard in the forecasting literature. In the following, we show that a class of non-parametric machine learning models directly provides coherent point forecasts. First, we state a standard result from convex analysis.

Proposition 1. *Any convex combination of historical observations $\mathbf{y}_t$ satisfies the coherency constraints.*

Proof. This is follows from the convexity of $\mathcal{S}$ and Assumption 1. □

The above holds for additional convex constraints, e.g., non-negativity of forecasts. A corollary of Proposition 1 is that a

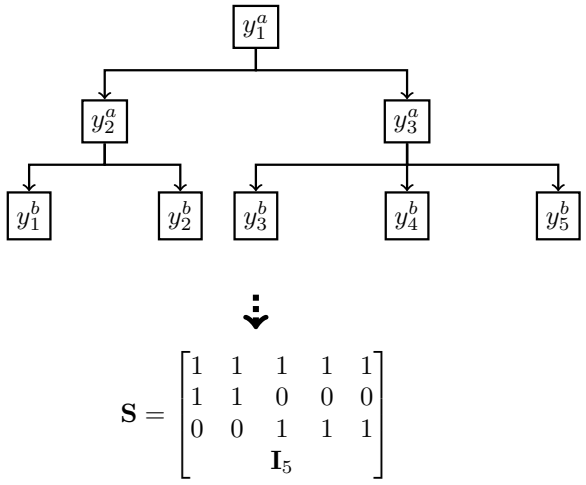


Fig. 1: Example of a 3-level hierarchy (top) and the respective aggregation matrix (bottom) with $n = 8$, $n_a = 3$, and $n_b = 5$.

class of machine learning models, including, among others, k-nearest neighbors, kernel regression, and decision trees, directly leads to coherent forecasts. These models are based on the idea of local averaging or smoothing of historical observations. For an out-of-sample observation $\mathbf{x}_{t+k}$, we derive a set of non-negative weights $\omega_t(\cdot)$, with $\sum_{t \in [T]} \omega_t(\mathbf{x}_{t+k}) = 1$, and the respective point forecast $\widehat{\mathbf{y}}_{t+k}$ is given by

$$\widehat{\mathbf{y}}_{t+k} = \sum_{t \in [T]} \omega_t(\mathbf{x}_{t+k}) \mathbf{y}_t, \tag{3}$$

i.e., a convex combination of historical observations. From Proposition 1, we see that $\widehat{\mathbf{y}}_{t+k}$ are coherent. The practical implication is that off-the-shelf machine learning tools are readily applicable for end-to-end hierarchical forecasting. This result is somewhat trivial; nonetheless, it seems to have escaped the respective forecasting literature.

The above-mentioned models can also be employed for probabilistic hierarchical forecasting. For example, the selected neighbors $\mathbf{y}_t$ in a k-nearest neighbor model can be treated as (coherent) sample path realizations of the joint predictive density of all the series in the hierarchy. Similarly, one could treat the output of individual trees within an ensemble as realizations of the multivariate predictive density. Therefore, off-the-shelf machine learning tools are also applicable to probabilistic hierarchical forecasting, presenting a computationally cheaper alternative to the internal sampling and projection approach proposed in [7] and the bottom-up method described in [5].

B. Dealing with missing values

Next, we examine the case when bottom-level series have missing values due to equipment failures, but aggregated series maintain correct measurements. We assume that missing values are set at 0, therefore $\mathbf{A}\mathbf{y}_t \neq \mathbf{0}$ and $\mathbf{y}_t \notin \mathcal{S}$. Without loss of generality, the term "missing" refers both to missing and erroneous measurements, as long as these are identified and do not propagate through the hierarchy. We

examine this problem under a conditional stochastic optimization lens, integrating predictive and prescriptive analytics [10], and formulate prescriptive trees for end-to-end hierarchical forecasting. Prescriptive trees [11] refer to decision trees that output prescriptions rather than predictions. In our case, and with a slight abuse of terminology, the prescriptions correspond to hierarchical point forecasts, which must satisfy the coherency constraints (and possibly additional ones). Out-of-sample conditional prescriptions are derived via a weighted Sample Average Approximation (SAA) of the original stochastic optimization problem [10].

We follow the popular CART method [12] by recursively partitioning the feature space with locally optimal splits. Mathematically, a node split separates a feature space $R \subseteq \mathbb{R}^{n\times p}$ at feature j and point s into two disjoint partitions $R = R_l \cup R_r$, such that $R_l = \{t \in [T] \mid x_{tj} < s\}$ and $R_r = \{t \in [T] \mid x_{tj} \geq s\}$, with scalar x_{tj} denoting the t-th observation of the j-th feature[1]. Here, we minimize a generic cost function subject to a set of linear aggregation constraints. The problem of finding the locally optimal split is given by

$$\min_{j,s} \left[\min_{\mathbf{z}_l \in \mathcal{S}} \sum_{t\in R_l(j,s)} c(\mathbf{z}_l; \mathbf{x}_t) + \min_{\mathbf{z}_r \in \mathcal{S}} \sum_{t\in R_r(j,s)} c(\mathbf{z}_r; \mathbf{x}_t) \right], \quad (4)$$

with subscripts l, r referring to the left and right child node, $\mathbf{z}_{\{l,r\}} \in \mathbb{R}^n$ being the locally constant decisions (i.e., hierarchical forecasts), which satisfy the coherency constraints $\mathcal{S}$, $R_{\{l,r\}}$ being index sets, and $c(\cdot)$ being the cost function to be minimized. Thus, the main difference from the CART algorithm is the requirement for predictions to satisfy a set of constraints and the use of a generic, task-based loss function.

In a typical regression setting, $c(\cdot)$ would correspond to the squared ℓ_2 norm. Here, we want to ignore missing values, without disregarding any quality data points. To this end, we employ an indicator matrix $\mathbf{\Gamma} \in \{0,1\}^{n\times T}$ that checks whether historical observations are missing. A single entry γ of $\mathbf{\Gamma}$ is given by

$$\gamma_{it} = \begin{cases} 1, & \text{if } y_{it} \text{ is missing} \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in [n], t \in [T], \quad (5)$$

and the cost of sample t is given by

$$c(\cdot) = \|\mathbf{y}_t - (\mathbf{1} - \boldsymbol{\gamma}_t) \odot \mathbf{z}\|_2^2, \quad (6)$$

with $\boldsymbol{\gamma}_t$ denoting the t-th column vector of $\mathbf{\Gamma}$ and $\odot$ denoting the elementwise multiplication. Note that series without missing values are weighted more heavily in the objective, therefore, reliable nodes within the hierarchy assume greater importance during training, which we consider to be a desirable property for this application.

As discussed, forecasts are required to satisfy the coherency constraints imposed by $\mathcal{S}$. Note the sub-problems in (4) are equality constrained quadratic problems. An analytical solution can be derived by solving a system of linear equations obtained from the Karush–Kuhn–Tucker (KKT) optimality conditions (see Appendix A). Other possible constraints, e.g., non-negativity or monotonicity, can be readily included. In this case, a general-purpose convex solver can be called on to evaluate (4). For an out-of-sample observation $\mathbf{x}_{t+k}$ point forecasts are derived via a weighted SAA given by

$$\widehat{\mathbf{y}}_{t+k} = \underset{\mathbf{z}\in\mathcal{S}}{\arg\min} \sum_{t\in[T]} \omega_t(\mathbf{x}_{t+k}) \|\mathbf{y}_t - (\mathbf{1} - \boldsymbol{\gamma}_t) \odot \mathbf{z}\|_2^2. \quad (7)$$

In general, decision trees are highly prone to overfitting. Randomization-based ensembles provide a remedy and lead to impressive predictive performance; these are readily applicable within the proposed framework, leading to a *prescriptive forest*. A single tree is fully compiled, with its leaves outputting coherent forecasts. The corresponding weights are given by

$$\omega_t(\mathbf{x}_{t+k}) = \frac{\mathbb{I}[R(\mathbf{x}_t) = R(\mathbf{x}_{t+k})]}{|R(\mathbf{x}_{t+k})|}, \quad (8)$$

where $R(\mathbf{x}_{t+k})$ is the leaf that out-of-sample observation $\mathbf{x}_{t+k}$ falls into, $|\cdot|$ the leaf cardinality, and $\mathbb{I}[\cdot]$ an indicator function that checks whether training observation $\mathbf{x}_t$ falls into $R(\mathbf{x}_{t+k})$. Lastly, for an ensemble of B trees the weights are obtained

$$\omega_t(\mathbf{x}_{t+k}) = \frac{1}{B} \sum_{b=1}^{B} \frac{\mathbb{I}[R^b(\mathbf{x}_t) = R^b(\mathbf{x}_{t+k})]}{|R^b(\mathbf{x}_{t+k})|}. \quad (9)$$

III. Experimental Setting and Results

This section details the experiment and respective results. We examine performance for day-ahead hourly forecasting of wind and PV production, considering only point forecasts. As this case study involves renewable energy forecasting, we also require forecasts to be non-negative, thus $\mathcal{S} := \{\mathbf{y} | \mathbf{A}\mathbf{y} = \mathbf{0}, \mathbf{y} \geq \mathbf{0}\}$; Proposition 1 holds as $\mathcal{S}$ remains convex.

1) Data: We use power measurements from 60 wind turbines and 20 PV parks located in mid-west France, with a nominal aggregated capacity of 120 MW and 4 MW, respectively. The available data sets span the period from December 2018 to September 2020 with an hourly resolution. Approximately 15 months of data are used for training and tuning, with the remaining 5 months used for testing the performance. For the PV data, only measurements recorded at a zenith angle under 85^o are considered, thus the effective sample size is smaller. Table I provides a summary of the two data sets. We examine performance in day-ahead point forecasting, with a horizon of 12-36 hours ahead. This setting is typical in market-related applications. In both cases, a 3-level hierarchy is considered. Wind production data are naturally aggregated at park level (13 wind parks in total); for the PV production data we construct a fictitious hierarchy based on spatial k-means clustering. Figure 2 provides an overview of the geographical distribution of the power plants.

For feature data we consider a grid of numerical weather predictions (NWPs) obtained from ECMWF, issued daily at 00:00 UTC with a spatial resolution of $0.1^o \times 0.1^o$. The 5 extracted features are: surface solar radiation downwards, 100m U- and V-wind speeds, 2m temperature and total cloud cover.

[1]For brevity of exposition we focus on quantitative features, although it is straightforward to also include categorical features

TABLE I: Data set description

	Train / test sample size	n_b	n_a	Capacity (MW)
Wind	13875 / 3671	60	14	120
PV	6355 / 2322	20	4	4

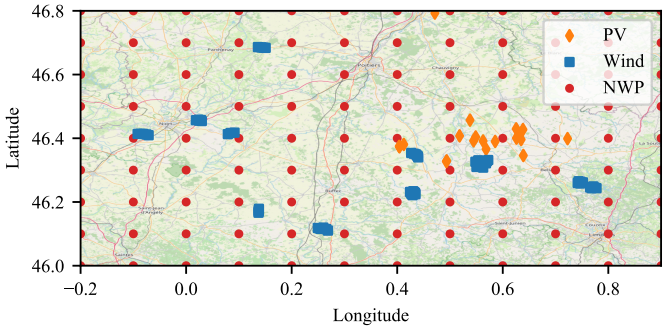


Fig. 2: Overview of the NWP grid points and the wind and PV plants.

We do not consider historical production lags as features, as these typically do not improve forecasts in the horizon of interest. Further, including historical lags would introduce missing values in $\mathbf{x}$, which is outside of our scope. For the i-th series, respective feature vector $\mathbf{x}_{it}$ comprises the NWPs from the closest grid point in terms of Euclidean distance; when forecasting a group of series in one-shot, respective features are concatenated in a single vector.

2) Verification: Individual production data are normalized by the nominal capacity and performance is assessed in terms of normalized root mean squared error (NRSME). Further, following [4], the Scaled RMSE (SRMSE) is used by dividing by the total number of child notes. For a set of T forecasts, the SRMSE for the i-th series is defined as:

$$\text{SRMSE}_i = \left(\frac{1}{T} \sum_{t \in [T]} \left(\frac{y_{it} - \widehat{y}_{it}}{s_i} \right)^2 \right)^{\frac{1}{2}}, \qquad (10)$$

where s_i the number of child nodes.

3) Benchmark models: We compare the efficacy of a variety of post-processing and end-to-end learning approaches. Note that for post-processing methods, any learning algorithm can be used to generate base forecasts. To allow for a fair comparison, similar base learners are considered in all cases, i.e., randomized tree ensembles based either on the Random Forest [13] or the ExtraTrees [14] algorithm. The following approaches are examined:

- `BASE`: Base forecasts with a Random Forest model for each series, without reconciliation. Typically, these will not be coherent.
- `BASE-BU`: Bottom-up reconciliation applied to the base forecasts of the bottom-level series.
- `BASE-PRJ`: Base forecasts post-processed with a Euclidean projection step. The reconciled forecasts are given by

$$\arg\min_{y \in \mathcal{S}} \|\mathbf{y} - \widehat{\mathbf{y}}_{t+k}\|_2, \qquad (11)$$

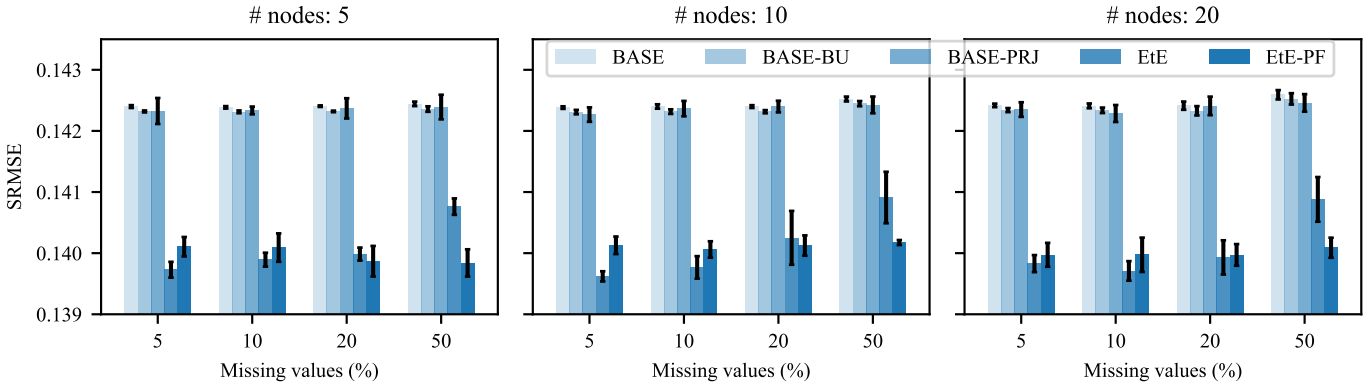
where $\widehat{\mathbf{y}}_{t+k}$ are the base forecasts at time t with horizon k. Alternative reconciliation methods, such as MinT [2] and constrained multivariate least squares [4], were also examined. However, as these methods require additional training, results with missing values were not robust and thus are omitted.
- `EtE`: A single Random Forest model predicting the whole hierarchy, to examine the efficacy of end-to-end learning.
- `EtE-PF`: End-to-end learning with prescriptive forests to deal with missing values.

In all cases, except for `EtE-PF`, missing values are disregarded prior to training. For the `EtE` approach, this means that observation $\mathbf{y}_t$ is disregarded if at least two of the n series have a missing value, thus we expect this approach to be affected the most by missing values. A grid search is performed to tune the hyperparameters of the Random Forest models. For the `EtE-PF`, we employ random node splits to speed-up computations, following the ExtraTrees algorithm [14], and similarly perform a grid search for tuning.

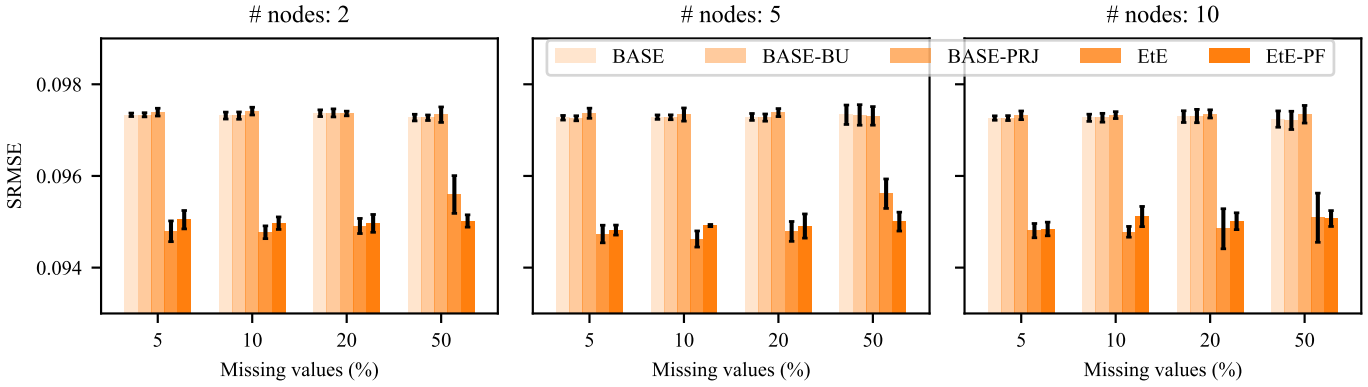
4) Results: We simulate the effect of missing values due to equipment malfunctions by sampling a subset of bottom-level nodes and setting a percentage of training observations at zero. We vary both the number of nodes and the percentage of missing values per node and repeat the experiment 5 times to derive aggregate statistics. For simplicity, we assume that missing values occur at the same timestamp for all nodes.

Fig. 3 shows the aggregated SRMSE as a function of the number of sampled nodes and the percentage of missing values per node. Overall, the following are observed: i) end-to-end learning (`EtE`) outperforms post-processing methods but is also more heavily affected by missing values, ii) post-processing methods are robust against missing values, and iii) the proposed `EtE-PF` combines the best of both worlds. Regarding the wind data set (Fig. 3a), both `EtE` and `EtE-PF` show improved accuracy for lower percentage of missing values, with the performance of `EtE` gradually degrading as the percentage of missing values increases. Conversely, the `EtE-PF` proves to be robust, consistently outperforming the reconciliation methods. This result persists both for the case of an increased number of malfunctioning nodes and an increased percentage of missing values per node. Further, `BASE-PRJ` performs, albeit slightly, better than the `BASE` and `BASE-BU`, corroborating previous findings on the benefits of post-processing. Similar results are observed for the PV data set (Fig. 3b), with a smaller sample size. Overall, the relative increase in average SRMSE for `EtE` from the smallest (5%) to the largest (50%) percentage of missing observations is 0.8% for the wind data set and 0.6% for the PV data set.

By examining the accuracy of end-to-end learning as a function of the number of selected nodes and the percentage of missing observations, on one hand, we observe that the former has a negligible effect in overall performance; this is partly attributed to the design of the experiment, as missing



(a) Wind production.



(b) PV production.

Fig. 3: Aggregated SRMSE for the hierarchy as a function of the number of sampled nodes and the percentage of missing values per node over 5 iterations. Bars correspond to one standard deviation.

TABLE II: Average SRMSE ($\pm$ one standard deviation) per hierarchy level. The best-performing model is underlined in bold font. Bold font indicates that a result does not differ from the best-performing model at the 1% level (Welch's t-test).

Data set	Level (# nodes)	BASE	BASE-BU	BASE-PRJ	EtE	EtE-PF
Wind	1	0.0969 $\pm$ 0.0002	0.0977 $\pm$ 0.0001	**0.0968 $\pm$ 0.0002**	0.0972 $\pm$ 0.0005	0.0971 $\pm$ 0.0003
	2 (13)	0.1327 $\pm$ 0.0132	0.1322 $\pm$ 0.0132	0.1326 $\pm$ 0.0123	**0.1306 $\pm$ 0.0128**	**0.1305 $\pm$ 0.0127**
	3 (60)	0.1453 $\pm$ 0.0144	0.1453 $\pm$ 0.0144	0.1452 $\pm$ 0.0138	**0.1429 $\pm$ 0.0141**	**0.1428 $\pm$ 0.0141**
PV	1	0.0761 $\pm$ 0.0001	0.0762 $\pm$ 0.0001	**0.0759 $\pm$ 0.0001**	0.0765 $\pm$ 0.0005	0.0764 $\pm$ 0.0002
	2 (3)	**0.0812 $\pm$ 0.0040**	**0.0811 $\pm$ 0.0041**	**0.0811 $\pm$ 0.0037**	**0.0808 $\pm$ 0.0035**	**0.0807 $\pm$ 0.0035**
	3 (20)	0.1008 $\pm$ 0.0176	0.1008 $\pm$ 0.0176	0.1009 $\pm$ 0.0170	**0.0980 $\pm$ 0.0153**	**0.0980 $\pm$ 0.0153**

values occur at the same timestamp across all nodes. On the other hand, the percentage of missing values has a more pronounced effect. In order to present a comprehensive study, we repeat the above experiment only for `EtE-PF` with missing values occurring at different timestamps. The results presented in Fig. 4 are similar to the ones achieved before, with the forecast accuracy decreasing only slightly as the number of nodes increases. Thus, we conclude that the proposed `EtE-PF` successfully mitigates the adverse effects of missing values in the lower levels of the hierarchy.

Lastly, we examine performance for each level of the hierarchy. From Table II we observe that for all of the examined models the SRMSE is lower for higher levels of aggregation due to the spatial smoothing effect. This effect is more pronounced for the wind production data, which we partly attribute to the fact that a larger number of wind production series are examined. For both data sets, the `EtE-PF` leads to the best performance in the 2nd and 3rd (bottom) level, while `BASE-PRJ` leads to the best performance for the 1st (top) level, with the results being, generally, statistically significant. Overall, both `EtE` and `EtE-PF` consistently improve performance for the bottom-level nodes, highlighting the benefits of exploiting dependencies across time series in an end-to-end learning setting. Note that the results shown in Table II are obtained over all the iterations (for uniform timestamps); the difference between `EtE-PF` and `EtE` becomes statistically

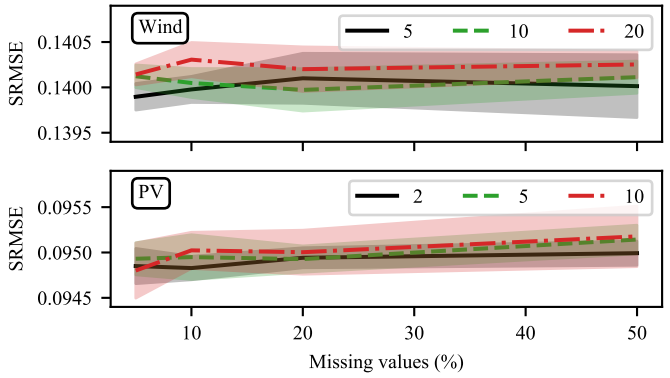


Fig. 4: Performance of EtE−PF for missing values at different timestamps. The lines indicate the number of malfunctioning nodes, the shaded areas show one standard deviation.

significant if we only examine adverse scenarios (higher percentage of missing values), as the performance of EtE declines.

IV. Conclusions

In this work, we examined the problem of forecasting the production of geographically distributed renewable energy sources with missing values under a conditional stochastic optimization lens. First, we showed that a large class of non-parametric machine learning models generates coherent hierarchical forecasts. The main practical implication is that off-the-shelf machine learning tools can be utilized for hierarchical forecasting, presenting an easy way to create benchmarks. Then, we proposed a prescriptive trees algorithm for end-to-end learning with missing values. Performance was evaluated in two case studies of wind and PV production point forecasting on a day-ahead horizon. On one hand, end-to-end learning showed improved aggregate performance against two-step reconciliation approaches; for the bottom-level series this improvement was 1.7% and 2.8% for the wind and PV data, respectively. On the other hand, reconciliation approaches proved to be more robust against the number of missing values. The proposed solution managed to combine the best of both worlds as it led to improved performance and also mitigated the adverse effect of missing data for end-to-end learning.

In future work, we plan to study probabilistic hierarchical forecasting with missing values and extend it to other relevant power system case studies, such as demand forecasting. Another interesting direction is to examine missing or corrupt values in the feature vector.

Acknowledgment

The authors wish to thank HESPUL, SERGIES, and VESTAS for the provision of renewable production data and ECMWF for Numerical Weather Predictions.

References

[1] F. Petropoulos, D. Apiletti, V. Assimakopoulos, M. Z. Babai, D. K. Barrow, S. Ben Taieb *et al.*, "Forecasting: theory and practice," *International Journal of Forecasting*, 2022.

[2] S. L. Wickramasuriya, G. Athanasopoulos, and R. J. Hyndman, "Optimal forecast reconciliation for hierarchical and grouped time series through trace minimization," *Journal of the American Statistical Association*, vol. 114, no. 526, pp. 804–819, 2019.

[3] T. Van Erven and J. Cugliari, "Game-theoretically optimal reconciliation of contemporaneous hierarchical time series forecasts," in *Modeling and stochastic learning for forecasting in high dimensions*. Springer, 2015, pp. 297–317.

[4] C. Di Modica, P. Pinson, and S. B. Taieb, "Online forecast reconciliation in wind power prediction," *Electric Power Systems Research*, vol. 190, p. 106637, 2021.

[5] S. B. Taieb, J. W. Taylor, and R. J. Hyndman, "Coherent probabilistic forecasts for hierarchical time series," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3348–3357.

[6] D. Yang, "Reconciling solar forecasts: Probabilistic forecast reconciliation in a nonparametric framework," *Solar Energy*, vol. 210, pp. 49–58, 2020.

[7] S. S. Rangapuram, L. D. Werner, K. Benidis, P. Mercado, J. Gasthaus, and T. Januschowski, "End-to-end learning of coherent probabilistic forecasts for hierarchical time series," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8832–8843.

[8] X. Han, S. Dasgupta, and J. Ghosh, "Simultaneously reconciled quantile forecasting of hierarchically related time series," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 190–198.

[9] Z. Liu, Y. Yan, J. Yang, and M. Hauskrecht, "Missing value estimation for hierarchical time series: A study of hierarchical web traffic," in *2015 IEEE International Conference on Data Mining*. IEEE, 2015, pp. 895–900.

[10] D. Bertsimas and N. Kallus, "From predictive to prescriptive analytics," *Management Science*, vol. 66, no. 3, pp. 1025–1044, 2020.

[11] D. Bertsimas, J. Dunn, and N. Mundru, "Optimal prescriptive trees," *INFORMS Journal on Optimization*, vol. 1, no. 2, pp. 164–183, 2019.

[12] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[13] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[14] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

Appendix

In this section we show how to derive analytical solutions for the SAA subproblems that arise in (4). Consider evaluating the SAA for $t \in [T]$ given by

$$\min_{\mathbf{z}} \left\{ \frac{1}{2} \sum_{t \in [T]} \|\mathbf{y}_t - (\mathbf{1} - \boldsymbol{\gamma}_t) \odot \mathbf{z}\|_2^2 \mid \mathbf{A}\mathbf{z} = \mathbf{0} \right\}. \tag{12}$$

For simplicity, the objective is scaled. The KKT optimality conditions for this problem can be written as

$$\sum_{t \in [T]} (\mathbf{y}_t - (\mathbf{1} - \boldsymbol{\gamma}_t) \odot \mathbf{z}) + \mathbf{A}^\intercal \mathbf{v} = \mathbf{0}, \mathbf{A}\mathbf{z} = \mathbf{0}, \tag{13}$$

where $\mathbf{v} \in \mathbb{R}^{n_a}$ denotes the dual variables. We write (13) as

$$\begin{bmatrix} \mathbf{P} & -\mathbf{A}^\intercal \\ \mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \sum_{t \in [T]} \mathbf{y}_t \\ \mathbf{0} \end{bmatrix}, \tag{14}$$

where $\mathbf{P} = \operatorname{diag}\left(\sum_{t \in [T]}(1 - \gamma_{1t}), \ldots, \sum_{t \in [T]}(1 - \gamma_{nt})\right)$ is an n-size diagonal matrix whose entries equal the number of non-missing values for each series. Therefore, we need to solve this set of $n + n_a$ linear equations in the $n + n_a$ variables. Lastly, note that it is possible for $\mathbf{P}$ to become singular; in this case, the least-squares solution of (14) can be used.