



HAL
open science

Prescriptive Trees for Value-oriented Forecasting and Optimization: Applications on Storage Scheduling and Market Clearing

Akylas Stratigakos, Simon Camal, Andrea Michiorri, Georges Kariniotakis

► **To cite this version:**

Akylas Stratigakos, Simon Camal, Andrea Michiorri, Georges Kariniotakis. Prescriptive Trees for Value-oriented Forecasting and Optimization: Applications on Storage Scheduling and Market Clearing. 2021. hal-03363876

HAL Id: hal-03363876

<https://hal.science/hal-03363876>

Preprint submitted on 4 Oct 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Prescriptive Trees for Value-oriented Forecasting and Optimization: Applications on Storage Scheduling and Market Clearing

Akylas Stratigakos, Simon Camal, Andrea Michiorri, Georges Kariniotakis
Center PERSEE
MINES ParisTech, PSL University
Sophia Antipolis, France

Abstract—Decision-making in the presence of contextual information is a ubiquitous problem in modern power systems. The typical data-decisions pipeline comprises several forecasting and optimization components deployed in sequence. However, the loss function employed during learning is only a proxy for task-specific costs (e.g. scheduling, trading). This work describes a data-driven alternative to improve prescriptive performance in conditional stochastic optimization problems based on nonparametric machine learning. Specifically, we propose prescriptive trees that minimize task-specific costs during learning, embedded with a scenario reduction procedure to reduce computations, and then derive a weighted Sample Average Approximation of the original problem. We present experimental results for two problems: storage scheduling for price arbitrage and network-constrained stochastic market clearing, respectively associated with electricity price and load forecasting. The empirical results show significant improvements on deterministic and stochastic lookahead policies, with the relative difference being more pronounced when training data are limited.

Index Terms—Data-driven optimization, decision trees, electricity market, prescriptive analytics, value-oriented forecasting

I. INTRODUCTION

The ongoing transition towards the smart grid era is associated with an influx of data from various sources. Making informed decisions under uncertainty, entails leveraging such contextual information (also known as explanatory data or features) within the data-decisions pipeline. The conventional modeling approach comprises a sequential process, where forecasting models first estimate uncertain parameters (e.g. load, market prices), which are subsequently input into an optimization problem. However, learning is performed by minimizing a proxy loss function, without considering the impact of forecasts on downstream costs. Assessing this impact and designing algorithms that maximize *forecast value*, rather than accuracy, constitute key challenges in energy forecasting [1] for the coming years.

Conditional stochastic optimization departs from the classical approach to study decision-making under uncertainty in

the presence of contextual information. The data-driven framework put forward in [2] and extended in [3] integrates machine learning with optimization by employing local learning (non-parametric) algorithms to infer a weighted Sample Average Approximation (SAA) of stochastic problems conditioned on contextual information. A related stream of research proposes learning under alternative loss functions to derive, possibly biased, forecasts that explicitly minimize downstream costs (or equivalently maximize forecast value) [4], termed cost-/value-oriented forecasts. Relevant applications on wind [5] and load [6] forecasting rely on a two-step approach that involves first inferring a convex loss, then training the model, which, however, does not directly leverage the optimization component. In [7] authors prescribe value-oriented demand forecasts to clear a day-ahead market, by considering forecast impact on balancing costs during learning. A generic end-to-end learning approach to train probabilistic forecasting models to minimize task-specific costs for problems with a smooth objective is described in [8], with applications in storage and grid scheduling. The trading cost for a risk-aware renewable producer is used as an alternative loss to forecast market quantities in [9]. Lastly, empirical risk minimization (ERM) can be applied to directly forecast decisions, see e.g. data-driven solutions to the newsvendor problem [10], with relevant applications on renewable trading found in [11], [12], respectively involving linear decision rules and neural networks. A main drawback, however, is that the ERM approach fails to guarantee the out-of-sample feasibility of decisions. Considering a discrete set of actions circumvents this issue; see e.g. mode-based control of storage posed as a supervised classification problem [13].

The work presented here integrates the framework established in [3] with learning under alternative loss functions, describing a decision tree algorithm that directly minimizes task-specific costs during learning and outputs decisions rather than predictions (*prescriptive trees*), effectively learning a policy from data. We extend our previous work [14] by considering a multi-temporal setting, describing an embedded scenario reduction procedure, and explicitly modeling recourse actions when generating prescriptions. To illustrate the proposed solution, we provide a comprehensive evaluation against deterministic and stochastic lookahead solutions derived under

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

the standard modeling approach in two applications: price arbitrage with storage and network-constrained market clearing. Further, we provide empirical evidence on the effect of training sample size and regularization hyperparameters on the out-of-sample optimization performance. Overall, the main contribution of this paper is to propose and validate an effective data-driven alternative to the standard "forecast, then optimize" modeling approach, offering a generalized perspective that jointly examines value-oriented forecasts and decisions, directly leverages the underlying optimization problem, and ensures out-of-sample feasibility. The proposed approach is generic and readily applicable to convex conditional stochastic optimization problems.

The rest of the paper is organized as follows. Section II details the prescriptive trees methodology. Section III describes the applications and presents the results. Finally, we conclude and provide directions for future research in Section IV.

II. METHODOLOGY

A. Prescriptive Analytics Problem

We consider optimization problems that involve uncertain parameters of interest $\xi \in \Xi \subseteq \mathbb{R}^{d_\xi}$ (e.g. load), associated with features $\gamma \in \Gamma \subseteq \mathbb{R}^{d_\gamma}$ (e.g. weather conditions). Our focus is on solving the prescriptive analytics problem

$$v = \min_{z \in \mathcal{Z}} \mathbb{E}_{\mathbb{P}}[c(z; \xi) | \gamma = \bar{\gamma}] = \min_{z \in \mathcal{Z}} \mathbb{E}_{\xi \sim \mathbb{P}_{\bar{\gamma}}} [c(z; \xi)], \quad (1)$$

where v is the objective value, $z \in \mathbb{R}^{d_z}$ is the decision vector, \mathcal{Z} is a convex set of feasible solutions, $c(\cdot)$ is a cost function, $\gamma = \bar{\gamma}$ is a feature observation, \mathbb{P} is the joint distribution of γ and ξ , and $\mathbb{P}_{\bar{\gamma}}$ is the marginal distribution of ξ conditioned on $\bar{\gamma}$. Typically, $\mathbb{P}, \mathbb{P}_{\bar{\gamma}}$ are unknown, thus we want to infer a solution from a training data set $\{(\xi_i, \gamma_i)\}_{i=1}^n$ of n observations, by learning a *policy* \hat{z} that varies as a function of γ . In this work, ξ_i typically represents a sample path realization of a stochastic process, e.g. one day of load observations. Further, we assume access to an oracle that efficiently generates an SAA of (1) with off-the-shelf solvers.

The standard modeling approach, termed "forecast, then optimize" (FO), suggests first deploying forecasting models to infer an approximation $\hat{\mathbb{P}}_{\bar{\gamma}}$ of the marginal conditional distribution of ξ (or its expectation), and then solving

$$\hat{z}^{FO} = \arg \min_{z \in \mathcal{Z}} \mathbb{E}_{\xi \sim \hat{\mathbb{P}}_{\bar{\gamma}}} [c(z; \xi)], \quad (2)$$

which now defines a conventional stochastic optimization problem, that can be tackled with standard methods. However, as the forecasting and optimization components are separate, the decision-maker cannot effectively hedge against miscalibrated forecasting models. Integrating machine learning and optimization offers a viable alternative in this case. Following [3], we consider a weighted SAA of (1) as

$$\hat{z}^{PP}(\bar{\gamma}) = \arg \min_{z \in \mathcal{Z}} \sum_{i=1}^n \omega_{n,i}(\bar{\gamma}) c(z; \xi_i), \quad (3)$$

termed *predictive prescription*, where $\omega_{n,i}(\bar{\gamma}) \geq 0$ denotes weights learned from a local learning algorithm, e.g. nearest

neighbors and decision trees, satisfying $\sum_{i=1}^n \omega_{n,i}(\bar{\gamma}) = 1$. Recall that for the standard SAA $\omega_{n,i}(\bar{\gamma}) = \frac{1}{n}$; thus we optimize over the empirical distribution of ξ without leveraging any contextual information. We extend this framework by proposing a methodology to derive $\omega_{n,i}(\bar{\gamma})$ by directly minimizing expected costs (1). Following [15], we formally define the problem of searching over functions $f : \Gamma \rightarrow \Xi$ that improve prescriptive performance as

$$\min_{f \in \mathcal{F}, z^f(\gamma_i) \in \mathcal{Z}} \sum_{i \in [n]} c(z^f(\gamma_i); \xi_i) \quad (4a)$$

s.t.

$$z^f(\gamma_i) = \arg \min_{z \in \mathcal{Z}} \sum_{i \in [n]} \omega_{n,i}^f(\gamma_i) c(z; \xi_i) \quad \forall i \in [n], \quad (4b)$$

where $[n] := \{1, \dots, n\}$. In the following, we focus exclusively on decision tree ensembles, which generally outperform other local learning algorithms.

B. Prescriptive Trees

The proposed tree algorithm follows the popular classification and regression trees (CART) [16] method, which recursively applies locally optimal splits, leading to a set of L leaves, $R_{1:L}$. A node split defines a disjoint partition of feature space at feature $j \in d_\gamma$ and point s : $R_1(j, s) = \{i \in [n] | \gamma_{ij} < s\}$ and $R_2(j, s) = \{i \in [n] | \gamma_{ij} \geq s\}$. Following [16], the locally optimal split that minimizes task-specific loss is derived from

$$\min_{j,s} \left[\min_{z_1 \in \mathcal{Z}} \sum_{\gamma_i \in R_1(j,s)} c(z_1; \xi_i) + \min_{z_2 \in \mathcal{Z}} \sum_{\gamma_i \in R_2(j,s)} c(z_2; \xi_i) \right]. \quad (5)$$

The inner minimization problems reflect the SAA solution of each partition, where \hat{z}_1, \hat{z}_2 are locally constant decisions. Decision trees are generally prone to overfitting, which hinders their capacity. An effective way of mitigating the adverse effects of overfitting, i.e., high variance, is via randomization-based ensemble methods such as Random Forests [17] and ExtraTrees [18], which can be readily adapted to our framework to grow ensembles of prescriptive trees (*prescriptive forests*).

Starting at the root node, we recursively partition the feature space until either a stopping criterion is met or performance cannot be further improved. Typical stopping criteria include the maximum tree depth Δ^{max} , the minimum number of observations n^{min} per leaf, and a predefined threshold for cost reduction. At each tree node, we apply perturbation by selecting a subset of features and candidate split points (and possibly bootstrapping samples), which is an integral part of ensemble methods. The ExtraTree algorithm suggests randomly selecting K features and a random candidate split point per feature (without bootstrapping), while in the Random Forest algorithm all candidate splits per selected feature are evaluated (with bootstrapping). In this work, we opt for the ExtraTree algorithm due to its reduced computations (fewer splits are evaluated per node). After selecting K candidate

splits, the aggregated cost (5) for each one is estimated and compared with the cost at its parent node, updating the tree structure accordingly. Algorithm 1 details the subroutine of splitting a single tree node with a random split criterion.

Algorithm 1 NodeSplit

Input: Data $\{\gamma_i, \xi_i\}_{i=1}^{n_0}$, current partition R_0 , current depth Δ , hyperparameters $\{n^{min}, K, \Delta^{max}\}$

Output: Local split (j, s)

- 1: Estimate cost $v_0 = \min_{z \in \mathcal{Z}} \sum_{\gamma_i \in R_0} c(z; \xi_i)$ and prescription $\hat{z}_0 = \arg \min_{z \in \mathcal{Z}} \sum_{\gamma_i \in R_0} c(z; \xi_i)$ at current node R_0
 - 2: **if** $\Delta < \Delta^{max}$ and $n_0 \geq 2n^{min}$ **then**
 - 3: Select K random features $\{\gamma_1, \dots, \gamma_K\}$
 - 4: Select random split point $s_\kappa \forall \kappa \in K$ from $U[\gamma_\kappa^{min}, \gamma_\kappa^{max}]$ such that $|R_1|, |R_2| \geq n^{min}$
 - 5: Solve (5) $\forall (\kappa, s_\kappa)$ {Optional: apply ReducedSAA}
 - 6: Find tuple (j^*, s^*) such that $v(s^*) = \min_{\kappa=1, \dots, K} v(s_\kappa)$
 - 7: **if** $v(s^*) < v_0$ **then**
 - 8: $\Delta \leftarrow \Delta + 1$
 - 9: **return** tuple (j^*, s^*) {Create new split}
 - 10: **else**
 - 11: **return** nothing {Node becomes leaf with output \hat{z}_0 }
 - 12: **end if**
 - 13: **end if**
-

Similarly to CART trees, for query $\bar{\gamma}$ the corresponding weights $\omega_{n,i}(\bar{\gamma})$ are derived as:

$$\omega_{n,i}(\bar{\gamma}) = \frac{\mathbb{I}[\gamma_i \in R(\bar{\gamma})]}{|R(\bar{\gamma})|}, \quad (6)$$

where $R(\bar{\gamma})$ denotes the target leaf that $\bar{\gamma}$ belongs to, $|\cdot|$ denotes the set cardinality, and $\mathbb{I}[\gamma_i \in R(\bar{\gamma})]$ is an indicator function to assess whether γ_i falls into $R(\bar{\gamma})$. For an ensemble of B trees the derived weights are

$$\omega_{n,i}(\bar{\gamma}) = \frac{1}{B} \sum_{b=1}^B \frac{\mathbb{I}[\gamma_i \in R^b(\bar{\gamma})]}{|R^b(\bar{\gamma})|}. \quad (7)$$

Single trees are fully compiled, i.e., leaves provide prescriptions without requiring further computations during implementation. For prescriptive forests, taking the average decision might lead to infeasibility, thus an additional step of solving (3) for a query $\bar{\gamma}$ is required.

Finally, it is straightforward to derive value-oriented forecasts as $\hat{\xi} = \sum_{i=1}^n \omega_{n,i}(\bar{\gamma}) \xi_i$. The key difference is that the internal tree structure is now dictated by local splits that minimize the expected cost of a constrained problem, rather than forecast error, as implemented in the CART method.

C. Embedded Scenario Reduction

Problem (5) is of a discrete nature and generally intractable for all but the simplest problems, which motivates the application of the ExtraTree algorithm, as it requires fewer evaluations per node. We propose to further reduce computations by accelerating individual split evaluations by means of applying

scenario reduction (the terms scenarios and sample paths are used interchangeably). First, we select an upper threshold for the number of scenarios n^{max} , such that $n^{max} \geq n^{min}$, which guarantees that scenario reduction will be applied in both partitions. Next, we apply a scenario reduction method, solve the SAA (5), and finally scale the expected cost back to aggregated cost. The process is detailed in Algorithm 2. Naturally, as we move down the tree structure, the average number of observations per node decreases. Therefore, scenario reduction mostly affects splits near the top of the tree, and is applied progressively less frequently moving downwards. We refer the reader to [19], and references therein, for an overview of the literature on scenario reduction and a comparison of different methods to a stochastic unit commitment problem. For this work, we employ k-medoids clustering to select representative samples.

Algorithm 2 ReducedSAA

Input: Data $\{\xi_i\}_{i=1}^{n_0}$, hyperparameter n^{max}

Output: SAA solution of subproblem with scenario reduction

- 1: Apply scenario reduction, generate new set of samples $\xi' \in \Xi'$, where $|\Xi'| = n^{max}$
 - 2: Estimate cost $v = \min_{z \in \mathcal{Z}} \frac{1}{|\Xi'|} \sum c(z; \xi'_i)$ and prescription $\hat{z} = \arg \min_{z \in \mathcal{Z}} \frac{1}{|\Xi'|} \sum c(z; \xi'_i)$
 - 3: Scale cost to aggregate level $v^{agg} = n_0 v$
 - 4: **return** (v^{agg}, \hat{z})
-

III. APPLICATIONS

To illustrate the efficacy of the proposed method, we compare it to the traditional modeling approach in two applications. The first considers scheduling a storage device to perform price arbitrage in a day-ahead (DA) market. Next, we consider stochastic market clearing with network constraints and uncertain demand. Thus, the two applications are associated with electricity price and load forecasting, respectively. To facilitate reproducibility of the results, we provide the code for the experiments in [20].

A. Forecasting Models

As discussed, the standard modeling approach requires first generating forecasts of uncertain parameters. Depending on the optimization module, these forecasts range from point forecasts, to probabilistic forecasts and trajectories (scenarios). As the purpose of this work is not to provide a comprehensive evaluation of forecasting models, but rather to showcase the efficacy of an alternative approach, we select established benchmarks. In the following, electricity prices and load forecasts are generated using a Quantile Regression Forests (QRF) model, which is a generalization of Random Forests. The QRF is robust to overfitting, requires little tuning, and can generate both point and probabilistic forecasts. Sample-based approximations of multi-period stochastic optimization

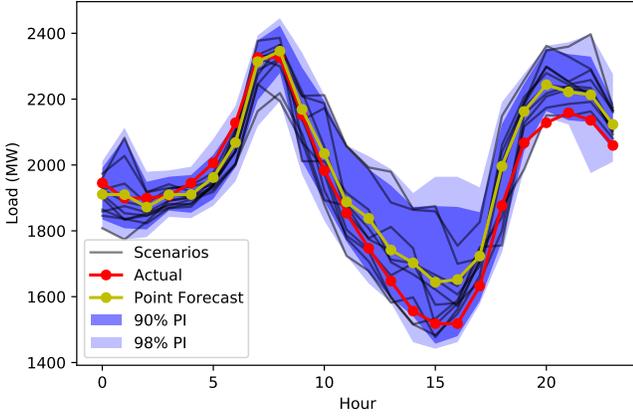


Fig. 1: Example of DA load forecasts: point forecasts, probabilistic forecasts as prediction intervals (PI), and scenarios.

problems require modeling temporal correlations between successive time periods; to this end, a Gaussian copula function is employed, given the conditional marginal predictive densities obtained from the QRF, following the procedure detailed in [21]. For the sake of completion, comparisons with naive benchmarks are also provided. For reference, Fig. 1 illustrates different types of load forecasts for the DA horizon.

B. Storage Scheduling and Price Forecasting

1) *Problem description*: The first application concerns scheduling a generic storage device, similar to [8]. The operator of a grid-scale storage device decides the charging z^{in} and discharging z^{out} actions, which consequently induce a specific state of charge z^{soc} , for each period t of the forecast horizon T . The operator wants to maximize profits via price arbitrage, while also considering battery degradation costs, and penalizing excessive deviations from the starting state. Here, we assume that degradation costs and excessive deviations (henceforth both referred to as penalties) are modeled as quadratic regularization terms. The problem, which depends on estimating the uncertain DA prices π^{da} , is formulated as

$$\min_{z_t^{in}, z_t^{out}, z_t^{soc}} \mathbb{E}_{\pi^{da}} \left[\sum_{t \in [T]} \pi_t^{da} (z_t^{in} - z_t^{out}) + \gamma \|z^{soc} - z_0\|_2^2 + \epsilon \|z^{out}\|_2^2 + \epsilon \|z^{in}\|_2^2 \right] \quad (8a)$$

s.t.

$$z_t^{soc} \leq z_t^{soc} \leq \bar{z}^{soc} \quad \forall t \in [T], \quad (8b)$$

$$z_t^{in} \leq z_t^{in} \leq \bar{z}^{in} \quad \forall t \in [T], \quad (8c)$$

$$z_t^{out} \leq z_t^{out} \leq \bar{z}^{out} \quad \forall t \in [T], \quad (8d)$$

$$z_{t+1}^{soc} = z_t^{soc} - \frac{1}{\eta^{out}} z_t^{out} + \eta^{in} z_t^{in} \quad \forall t \in [T-1], \quad (8e)$$

$$z_1^{soc} = z_0, \quad z_T^{soc} - \frac{1}{\eta^{out}} z_T^{out} + \eta^{in} z_T^{in} = z_0, \quad (8f)$$

TABLE I: Problem Parameters

Parameter	Value
\bar{z}^{soc}	1
\bar{z}^{in}	0.5
\bar{z}^{out}	0.2
z^{soc}, z^{in}, z^{out}	0
η^{in}	0.8
η^{out}	0.9
γ, ϵ	{0.01, 0.05, 0.1, 0.5, 1}

where η^{in}, η^{out} denote the charging and discharging efficiency, z_0 the initial state of charge, z (resp. \bar{z}) the lower (upper) limit of decision variables, and $[T] := \{1, \dots, T\}$. For simplicity, the state of charge must be at the starting level at the end of the day. Recourse actions are not considered. The following approaches (policies) are compared:

- FO-Deterministic (FO-Det): Generate point price forecasts, then solve a deterministic optimization problem, which is the standard modeling approach.
- Prescriptive Forest (PF): Generate predictive prescriptions with the proposed prescriptive forest algorithm. This is equivalent to solving a deterministic optimization problem with value-oriented forecasts.
- SAA: Determine an SAA of (8) given sample paths of $\{\pi^{DA}\}_{i=1}^n$, which minimizes in-sample cost. The SAA serves as a naive benchmark, since it ignores contextual information.

2) *Experiment setup*: This step employs a total of 3 years of market data from France, obtained from the ENTSO-E Transparency platform. Features γ include lagged prices, predicted net load at the system level, and categorical variables for month, weekday, and hour of the day. We consider a batch-mode learning setting and vary the size of the training sample n from 6 months to 2 years to assess the sensitivity of the out-of-sample performance. We also examine the impact of regularization parameters γ, ϵ (for simplicity assumed equal). Table I presents the complete list of indicative parameter values used. For all cases, evaluation is performed on the last year of data. To train the PF, the hyperparameters are set as $\{B = 50, K = d_x, n^{min} = 10\}$, with individual trees being maximally grown, and without scenario reduction.

3) *Results*: Fig. 2 summarizes the out-of-sample prescriptive performance of all of the considered cases. Overall, the PF consistently outperforms the other benchmarks, resulting in the highest profit, while also setting the efficient frontier in all cases (values towards the top and left are better). With respect to the standard FO-Det, for a given value of γ, ϵ , the PF generally incurs similar penalties combined with significantly higher profits. Regarding sensitivity to sample size, in general, larger samples translate into improved optimization performance, which is expected as the learning component improves. Regularization penalties remain fairly stable throughout; however, attained profit steadily increases with the sample size, with results converging for 2 years of training data. The PF and the SAA exhibit less sensitivity to sample size compared

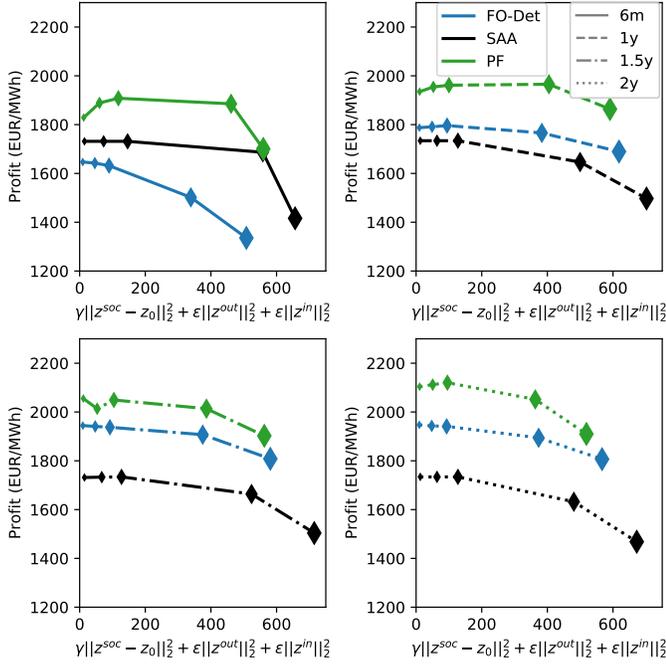


Fig. 2: Plots of aggregated profit versus penalties for the various sample sizes. Marker size is analogous to the value of γ, ϵ . Points towards the top and left are better.

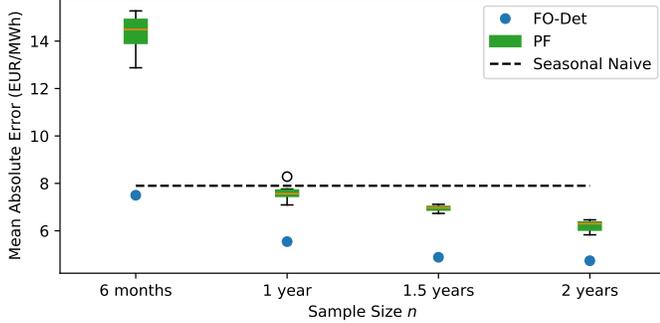


Fig. 3: Forecast accuracy for standard (FO-Det), value-oriented (PT), and seasonal naive forecasts.

to the FO-Det approach, which fails to outperform the naive SAA solution for the smallest sample examined (6 months). Thus the SAA poses an attractive alternative for the case of limited training data. For larger sample size, both the PF and the FO-Det outperform the SAA, with results being more pronounced for larger values of γ, ϵ . Finally, Fig. 3 examines forecast accuracy under a standard loss function (from FO-Det) and an alternative loss function (from PF), highlighting that, in this case study, improved accuracy does not translate into improved optimization performance.

C. Stochastic Market Clearing and Load Forecasting

1) *Problem description:* Next, we examine market clearing mechanisms in conjunction with the different modeling approaches. We consider clearing a forward (day-ahead or DA) market and a real-time (RT) market with network constraints

and uncertain demand. In the DA horizon, the system operator sets the dispatch levels of controllable generators in order to satisfy the uncertain demand, while respecting a set of technical and network constraints. When uncertainty is realized during the RT operation, corrective dispatch actions are required to maintain the demand-supply equilibrium. A DC linear approximation of network constraints is applied, as is standard practice. For simplicity, the unit commitment problem (start-up/shut-down actions) is not modeled. Note that in this subsection we are primarily interested in improving the out-of-sample prescriptive performance for a given market clearing mechanism, rather than examining in detail its economic properties. The following approaches are compared:

- Deterministic market clearing under FO (FO-Det): Generate point estimates of uncertain load, then solve a deterministic problem to clear the network-constrained DA market. This defines a multi-period DC-OPF problem, and serves as the base solution.
- Stochastic market clearing under FO (FO-Stoch): Generate probabilistic load forecasts and temporally correlated trajectories, then clear the market with a two-stage linear program with recourse actions, as proposed in [22]. Here, the DA schedule considers the forecast uncertainty and anticipates possible network congestion.
- Stochastic market clearing with PF (PF-Stoch): Derive predictive prescriptions by embedding the two-stage linear program [22] within the proposed tree algorithm. The key difference with the FO-Stoch is that scenarios are now weighted based on prescriptive importance.

On a high level, the two-stage linear program with recourse actions is formulated as

$$\begin{aligned} \min_{z, y(\xi)} \quad & \mathbb{E}_{\xi} \left[\sum_{t \in [T]} c(z_t, y_t(\xi); \xi | \gamma = \bar{\gamma}) \right] \\ \text{s.t.} \quad & \end{aligned} \quad (9)$$

$$\text{DA and RT Nodal Balance} \quad \forall \xi \in \Xi, t \in [T],$$

$$\text{DA and RT Flow Limits} \quad \forall \xi \in \Xi, t \in [T],$$

$$\text{Generator Technical Limits} \quad \forall \xi \in \Xi, t \in [T],$$

where z_t are the first-stage (DA) decisions, and $y_t(\xi)$ the recourse (RT) decisions that depend on the realization of uncertainty ξ , for each period t . The detailed mathematical formulation is provided in Appendix A.

For simplicity, we further assume that reserve capacity procurement does not incur additional costs, therefore, all flexible operators with spare capacity are eligible for providing upward and downward regulation during RT operation (w.r.t to technical constraints). As the stochastic market clearing formulation co-optimizes energy and reserves, this mainly affects the deterministic formulation, in order to avoid clearing an additional reserve capacity market. Finally, we highlight that our solution explicitly accounts for recourse actions, and network and temporal constraints when generating predictive prescriptions, which are not considered in relevant works [6], [7].

2) *Experiment setup*: The modified IEEE-24 bus system described in [23] is used in this experiment. Regarding load, we employ data from the Global Energy Forecasting Competition 2014 (GEFCom2014) [24]. A total of 2.5 years of load data is selected, with the last year reserved for testing. Features γ comprise day-ahead temperature forecasts (averaged over all meteorological stations), historical lags and relevant calendar variables. We assume a peak load of 2700MW and scale the GEFCom data between the range of $[0.20 \times 2700, 2700]$. The load is distributed to the respective nodes according to the percentages provided in [23]. The cost of load shedding is set at $500\$/MWh$. To avoid infeasible solutions during RT operation, an additional slack is included for downward regulation. The cost is set significantly lower than load shedding, at $200\$/MWh$. Regarding model training, the PF-Stoch hyperparameters are set at $\{B = 20, K = 5, n^{min} = 10\}$, with individual trees trained in parallel. Scenario reduction is discussed when applicable.

3) *Results*: Table II presents the aggregated out-of-sample costs as a function of sample size n , with PF-Stoch resulting in the lowest overall costs. Further examination shows that the PF-Stoch leads to the highest expected DA costs and the lowest expected redispatch costs, as illustrated in Fig. 4. Similarly to the previous experiment, the difference between the PF-Stoch and the FO solutions is more pronounced for smaller n , with all models converging for 1.5 year of data. For a sample size of 6 months, the learning component of the FO methods fails to generate adequate load forecasts. The stochastic market clearing design (both FO-Stoch and PF-Stoch) leads to lower aggregated costs, as a result of reduced redispatch costs during RT operation which is to be expected, as the stochastic formulation endogenously models the forecast uncertainty. The deterministic market clearing (FO-Det) consistently results in lower DA costs, also to be expected.

To gain further insight, Fig. 5 examines a single day of operation for the full training data. Fig. 5a presents the aggregated DA production, which matches the respective load forecast, and the actual aggregated demand. Note that both approaches incur similar forecast error for the selected day. In Fig. 5b, the DA dispatch schedule of the flexible generators is presented for 7:00, 15:00, and 23:00. In all cases, the PF-Stoch approach results in a larger number of generators being scheduled, with fewer of them reaching their maximum capacity. In turn, this leads to higher availability of regulation services, in both directions, during RT operation. Interestingly, this holds true even for 7:00 and 23:00, where the aggregated scheduled production for the PF-Stoch is significantly smaller than for the FO-Det and FO-Stoch approaches. This is attributed to the PF assigning scenario weights according to their prescriptive importance. In contrast, in the FO-Stoch solution all scenarios are equiprobable; therefore, a scenario that could possibly lead to large redispatch costs is considered solely based on probability of occurrence, which mitigates its impact on downstream costs. Overall, we conclude that the proposed modeling approach improves the utility of the

TABLE II: Aggregated Operational Cost ($10^6\%$)

Sample Size n	FO-Det	FO-Stoch	PF-Stoch
6 months	119.39	115.69	<u>80.04</u>
1 year	83.49	81.90	<u>73.84</u>
1.5 years	82.20	81.70	<u>73.77</u>

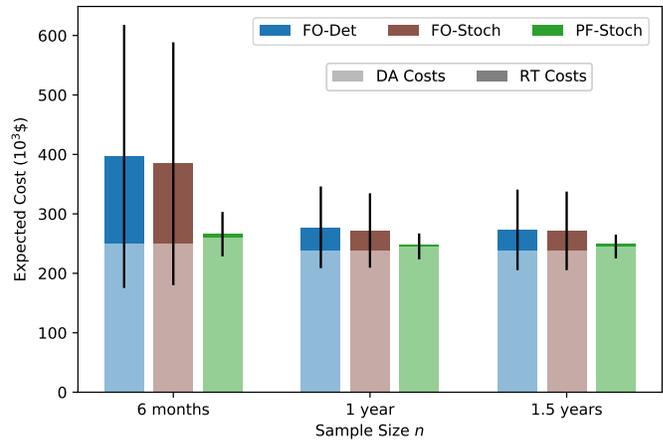


Fig. 4: Expected daily operating cost.

stochastic market clearing design.

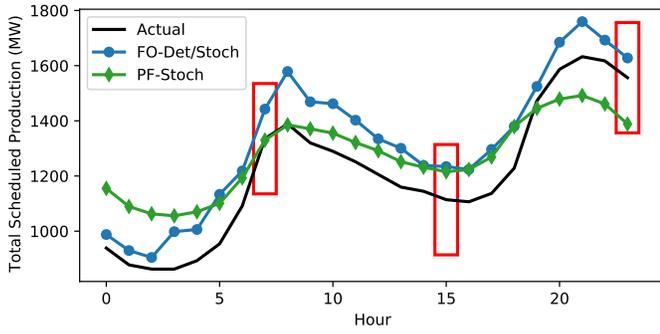
Finally, we examine the impact of scenario reduction during learning. The results presented in Table III are obtained using a standard PC featuring Intel Core i7 CPU with a clock rate of 2.7 GHz and 16GB of RAM (no parallelization), for the full dataset, and hyperparameters $\{B = 20, K = 5, n^{min} = 10\}$. We observe that a higher reduction in CPU time correlates with a larger performance decrease, with a maximum reduction of around 81% in CPU time for $n^{max} = 10$. For $n^{max} = 50$, a 43% reduction in CPU time is associated with a modest 2.87% decrease in performance. Note that increasing K , i.e., the number of splits evaluated per node, is expected to result in even larger relative reductions in CPU time. Overall, the embedded scenario reduction provides an attractive method to improve scalability with a modest decrease in performance.

IV. CONCLUSIONS

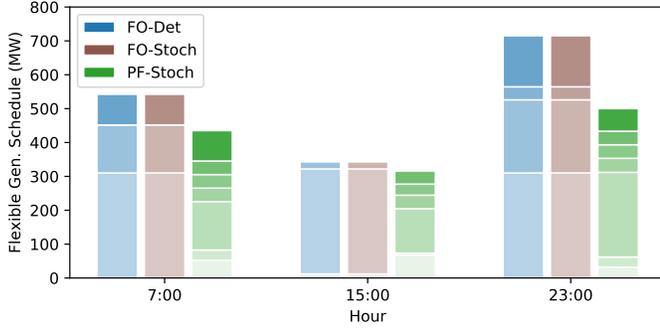
This work presented a data-driven method for value-oriented forecasting and decision-making in the presence of contextual information. The proposed method is based on prescriptive trees, trained under alternative loss function, that approximate the solution to conditional stochastic optimization problems.

TABLE III: Impact of Scenario Reduction on Average CPU Time and Performance

	PF	PF with Scenario Reduction, n^{max}		
		10	20	50
CPU time (min)	18.05 \pm 1.42	3.35 \pm 0.38	5.74 \pm 0.62	10.26 \pm 1.2
Cost increase (%)	-	17.71%	7.46%	2.87%



(a) Aggregated DA dispatch schedule. Both approaches incur a Mean Absolute Error of 116.1MW for the selected day.



(b) DA dispatch schedule of flexible generators for specific hours. Color density ranks generators according to cost (higher density denotes higher cost).

Fig. 5: DA schedule for a single day.

An internal scenario reduction methodology to facilitate training is also described. Experiments in two case studies examine the performance compared to deterministic and stochastic lookahead policies derived under the standard “forecast, then optimize” approach. The results show that the proposed method improves the out-of-sample prescriptive performance, and is less sensitive to the training sample size. Overall, the results validate the proposed approach as an effective alternative to the standard modeling approach. Future work could focus on adapting the proposed approach in an online setting and tackling sequential decision-making problems.

REFERENCES

- [1] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, “Energy forecasting: A review and outlook,” *IEEE Open Access Journal of Power and Energy*, 2020.
- [2] L. Hannah, W. Powell, and D. Blei, “Nonparametric density estimation for stochastic optimization with an observable state variable,” *Advances in Neural Information Processing Systems*, vol. 23, pp. 820–828, 2010.
- [3] D. Bertsimas and N. Kallus, “From predictive to prescriptive analytics,” *Management Science*, vol. 66, no. 3, pp. 1025–1044, 2020.
- [4] A. N. Elmachtoub and P. Grigas, “Smart “predict, then optimize”,” *Management Science*, 2021.
- [5] G. Li and H.-D. Chiang, “Toward cost-oriented forecasting of wind power generation,” *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2508–2517, 2016.
- [6] J. Zhang, Y. Wang, and G. Hug, “Cost-oriented load forecasting,” *arXiv preprint arXiv:2107.01861*, 2021.
- [7] J. M. Morales, M. Muñoz, and S. Pineda, “Prescribing net demand for electricity market clearing,” 2021.

- [8] P. L. Donti, B. Amos, and J. Z. Kolter, “Task-based end-to-end model learning in stochastic optimization,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5490–5500.
- [9] A. Stratigakos, A. Michiorri, and G. Kariniotakis, “A value-oriented price forecasting approach to optimize trading of renewable generation,” in *2021 IEEE Madrid PowerTech*, 2021, pp. 1–6.
- [10] G.-Y. Ban and C. Rudin, “The big data newsvendor: Practical insights from machine learning,” *Operations Research*, vol. 67, no. 1, pp. 90–108, 2019.
- [11] M. Munoz, J. M. Morales, and S. Pineda, “Feature-driven improvement of renewable energy forecasting and trading,” *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3753–3763, 2020.
- [12] T. Carriere and G. Kariniotakis, “An integrated approach for value-oriented energy forecasting and data-driven decision-making application to renewable energy trading,” *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6933–6944, 2019.
- [13] G. Henri and N. Lu, “A supervised machine learning approach to control energy storage devices,” *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 5910–5919, 2019.
- [14] A. Stratigakos, S. Camal, A. Michiorri, and G. Kariniotakis, “Prescriptive Trees for Integrated Forecasting and Optimization Applied in Trading of Renewable Energy,” Aug. 2021, working paper or preprint. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-03330017>
- [15] N. Mundru, “Predictive and prescriptive methods in operations research and machine learning: an optimization approach,” Ph.D. dissertation, Massachusetts Institute of Technology, 2019.
- [16] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.
- [17] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [18] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.
- [19] Y. Dvorkin, Y. Wang, H. Pandzic, and D. Kirschen, “Comparison of scenario reduction techniques for the stochastic unit commitment,” in *2014 IEEE PES General Meeting—Conference & Exposition*. IEEE, 2014, pp. 1–5.
- [20] https://github.com/akylasstrat/prescriptive_trees_power_apps.
- [21] P. Pinson, H. Madsen, H. A. Nielsen, G. Papaefthymiou, and B. Klöckl, “From probabilistic forecasts to statistical scenarios of short-term wind power production,” *Wind Energy: An International Journal for Progress and Applications in Wind Power Conversion Technology*, vol. 12, no. 1, pp. 51–62, 2009.
- [22] J. M. Morales, A. J. Conejo, K. Liu, and J. Zhong, “Pricing electricity in pools with wind producers,” *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1366–1376, 2012.
- [23] C. Oudoudis, P. Pinson, J. M. Morales, and M. Zugno, “An updated version of the ieeer ts 24-bus system for electricity market and power system operation studies,” *Technical University of Denmark*, vol. 13, 2016.
- [24] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, “Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond,” *International Journal of Forecasting*, vol. 32, no. 3, pp. 896–913, 2016.

APPENDIX

A. Mathematical formulation for the market clearing problem

Let \mathcal{G} denote the set of generators, \mathcal{L} the set of inelastic loads, Λ the set of transmission lines, and \mathcal{V} the set of buses. We further define $B \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ to be the bus susceptance matrix, $A \in \mathbb{R}^{|\Lambda| \times |\mathcal{V}|}$ to be the bus-branch incidence matrix, $b^{diag} \in \mathbb{R}^{|\Lambda| \times |\Lambda|}$ to be a diagonal matrix with line susceptances, and $A^{\mathcal{G}} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{G}|}$ and $A^{\mathcal{L}} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{L}|}$ to be auxiliary matrices that map generators and loads to buses. The generation $p \in \mathbb{R}^{|\mathcal{G}|}$ and the voltage angles $\theta^{da} \in \mathbb{R}^{|\mathcal{V}|}$ define the first-stage (DA) decision variables, while the uncertain demand is denoted as $d \in \mathbb{R}^{|\mathcal{L}|}$. All symbols are augmented by index t to refer to a specific period. The forecast horizon is $T = 24$, and $[T]$ is shorthand for $\{1, \dots, T\}$. Second-stage

(RT) decisions are further augmented by index ξ to refer to a specific scenario.

1) *Deterministic formulation*: The deterministic formulation of the market clearing problem with network constraints (multi-period DC-OPF) is as follows

$$\underset{Z}{\text{minimize}} \quad \sum_{t \in [T]} c^\top p_t \quad (10a)$$

subject to

$$A^G p_t - A^L \hat{d}_t = B \theta_t^{da} \quad \forall t \in [T], \quad (10b)$$

$$0 \leq p_t \leq \bar{p} \quad \forall t \in [T], \quad (10c)$$

$$-\bar{R}^d \leq p_t - p_{t-1} \leq \bar{R}^u \quad \forall t \in [T], \quad (10d)$$

$$-\bar{f} \leq b^{diag} A \theta_t^{da} \leq \bar{f} \quad \forall t \in [T], \quad (10e)$$

$$\theta_t^{da,ref} = 0 \quad \forall t \in [T], \quad (10f)$$

where $Z = \{p, \theta^{da}\}$ the set of DA decisions and \hat{d}_t the expected nodal demand (forecasts). The objective (10a) minimizes the DA dispatch costs, subject to nodal balance constraints (10b), generator technical limits (10c), ramping constraints (10d), and line capacity limits (10e), with (10f) setting the reference node.

2) *Redispatch problem*: When uncertainty is realized during RT operation, the operator achieves the demand-supply equilibrium by solving the redispatch problem. Here, the optimal solutions (p_t^*, θ_t^{da*}) define problem parameters. For a specific realization ξ' of uncertainty d , the problem is formulated as follows

$$\underset{\Psi}{\text{minimize}} \quad \sum_{t \in [T]} c^\top \langle r_{\xi't}^u, r_{\xi't}^d, s_{\xi't}^u, s_{\xi't}^d \rangle \quad (11a)$$

subject to

$$A^G (r_{\xi't}^u - r_{\xi't}^d - s_{\xi't}^d) + A^L (s_{\xi't}^u - d_{\xi't} + \hat{d}_t) = B (\theta_{\xi't}^{rt} - \theta_t^{da*}) \quad \forall t \in [T], \quad (11b)$$

$$0 \leq r_{\xi't}^u \leq \min(\bar{p} - p_t^*, \bar{R}^+) \quad \forall t \in [T], \quad (11c)$$

$$0 \leq r_{\xi't}^d \leq \min(p_t^*, \bar{R}^-) \quad \forall t \in [T], \quad (11d)$$

$$0 \leq s_{\xi't}^d \leq p_t^* - r_{\xi't}^d \quad \forall t \in [T], \quad (11e)$$

$$0 \leq s_{\xi't}^u \leq d_{\xi't} \quad \forall t \in [T], \quad (11f)$$

$$-\bar{f} \leq b^{diag} A \theta_{\xi't}^{rt} \leq \bar{f} \quad \forall t \in [T], \quad (11g)$$

$$\theta_{\xi't}^{rt,ref} = 0 \quad \forall t \in [T], \quad (11h)$$

where $\Psi = \{r_{\xi't}^u, r_{\xi't}^d, s_{\xi't}^u, s_{\xi't}^d, \theta_{\xi't}^{rt}\}$ defines the set RT decisions under scenario ξ' . Specifically, $r^u, r^d \in \mathbb{R}^{|\mathcal{G}|}$ denote the generators upward/downward regulation, $s^u \in \mathbb{R}^{|\mathcal{L}|}$ denotes a slack variable for upward regulation (i.e., load shedding), $s^d \in \mathbb{R}^{|\mathcal{G}|}$ denotes a slack variable for downward regulation, and $\theta^{rt} \in \mathbb{R}^{|\mathcal{V}|}$ denotes the RT voltage angles. The RT redispatch problem minimizes the linear balancing costs (11a), subject to RT nodal balance constraints (11b), technical constraints for the corrective actions (11c)-(11f), and RT flows (11g), (11h). Slack s^d is used to avoid infeasible solutions, by imposing

an additional cost for downward regulation exceeding the maximum downward reserve capacity \bar{R}^- .

3) *Stochastic Market Clearing*: The stochastic formulation [22] explicitly models recourse actions, thus reserves are estimated endogenously. For $\xi \in \Xi$, where $\Xi = \{\xi_i\}_{i=1}^n$ is a finite set of scenarios, the two-stage linear program is defined as

$$\underset{Z, \Psi_\xi}{\min} \quad \sum_{t \in [T]} \left[c^\top p_t + \frac{1}{n} \sum_{\xi \in \Xi} c^\top \langle r_{\xi t}^u, r_{\xi t}^d, s_{\xi t}^u, s_{\xi t}^d, s_{\xi t}^u \rangle \right] \quad (12a)$$

s.t.

$$\text{constraints (10b) - (10f)} \quad \forall t \in [T] \quad (12b)$$

$$\text{constraints (11b) - (11h)} \quad \forall \xi \in \Xi, t \in [T] \quad (12c)$$

The predictive prescriptions generated with the proposed PF approach are based on the same formulation, with the objective (12a) being modified according to the learned weights $\omega_{n,i}(x)$, as follows

$$\underset{Z, \Psi_\xi}{\min} \quad \sum_{t \in [T]} \left[c^\top p_t + \omega_{n,i}(x) \sum_{\xi \in \Xi} c^\top \langle r_{\xi t}^u, r_{\xi t}^d, s_{\xi t}^u, s_{\xi t}^d, s_{\xi t}^u \rangle \right] \quad (13a)$$

s.t.

$$\text{constraints (10b) - (10f)} \quad \forall t \in [T], \quad (13b)$$

$$\text{constraints (11b) - (11h)} \quad \forall \xi \in \Xi, t \in [T]. \quad (13c)$$