Towards Resilient Energy Forecasting: A Robust Optimization Approach

Akylas Stratigakos, Panagiotis Andrianesis, Andrea Michiorri, Georges Kariniotakis

Towards Resilient Energy Forecasting: A Robust Optimization Approach

Akylas Stratigakos, *Graduate Student Member, IEEE,* Panagiotis Andrianesis, *Member, IEEE,* Andrea Michiorri, and Georges Kariniotakis, *Senior Member, IEEE,*

***Abstract*—Energy forecasting models deployed in industrial applications face uncertainty *w.r.t.* data availability, due to network latency, equipment malfunctions or data-integrity attacks. In particular, the case when a subset of features that has been used for model training becomes unavailable when the model is used operationally poses a major challenge to forecasting performance. Ad-hoc solutions, e.g., retraining without the missing features, may work for a small number of features, but they soon become impractical, as the number of models grows exponentially with the number of features. In this work, we present a principled approach to introducing resilience against missing features in energy forecasting applications via robust optimization. Specifically, we formulate a robust regression model that is optimally resilient against missing features at test time, considering both point and probabilistic forecasting. We develop three solution methods for the proposed robust formulation, all leading to Linear Programming problems, with varying degrees of tractability and conservativeness. We provide an extensive empirical validation of the proposed methods in prevalent applications, namely, electricity price, load, wind production, and solar production, forecasting, and we further compare against well-established benchmark models and methods of dealing with missing features, i.e., imputation and retraining. Our results demonstrate that the proposed robust optimization approach outperforms imputation-based models and exhibits similar performance to retraining without the missing features, while also maintaining computational practicality. To the best of our knowledge, this is the first work that introduces resilience against missing features into energy forecasting.**

***Index Terms*—resilient energy forecasting, missing data, missing features, robust optimization, robust regression.**

I. Introduction

SHORT-TERM forecasting, ranging from a few minutes to a few days ahead, is key to ensure the safe, reliable, and economic operation of modern power systems. It pertains to several applications, such as load [1], electricity price [2], wind production [3], and solar production [4], forecasting, which, throughout this paper, will be referred to as *energy forecasting* [5]. The overarching goal in all applications is to estimate some characteristics of a target variable at a future time interval, such as the mean or a set of quantiles, as a function of associated features. For example, wind production is associated with wind speed, load is associated with temperature, etc.

This work was supported in part by the REgions project funded by the ADEME's 'Investissement d'Avenir' program and the ERA-Net SES RegSys Program (Grant No 646039), and in part by the Smart4RES Project (Grant No 864337) funded under the Horizon 2020 Framework Program.

A. Stratigakos, A. Michiorri, and G. Kariniotakis are with Center for processes, renewable energy and energy systems (PERSEE), Mines Paris, PSL University, 06904 Sophia Antipolis, France: {akylas.stratigakos, andrea.michiorri, georges.kariniotakis}@minesparis.psl.eu. P. Andrianesis is with the Department of Wind and Energy Systems, Technical University of Denmark (DTU), Lyngby 2800, Denmark: panosa@dtu.dk.

A. Background and Motivation

Arguably, most research on energy forecasting focuses on improving predictive performance, which largely depends on data *quality* and *availability*. During the development of the forecasting model, i.e., at training time, potential missing data are usually treated in a preprocessing step. The implicit assumption is that input data would be complete and always available during the forecasting model deployment, i.e., at test time. However, real-world industrial applications may face several operational data management challenges that would emerge only after the model is deployed [6]. Undoubtedly, missing features in an operational setting, i.e., when a subset of features used for model training becomes unavailable at test time, may severely affect forecasting performance. Ideally, models deployed in industrial applications should be *resilient* [7], i.e., they should maintain consistent performance, without requiring excessive manual tuning or relying on empirical solutions, in case that data are not available when needed.

There are several reasons that could lead to missing features (or *feature deletion*), e.g., malicious data-integrity attacks, network latency, and sensor failures. In Europe, e.g., system operators must publish, at specific times of day, various day-ahead predictions and system data, which are subsequently used by stakeholders as input to, e.g., electricity price forecasting models. However, an EC survey [8] that assesses the timeliness of data published on the ENTSO-E transparency platform finds that "for every data domain, fewer than 40% of users reported that data were always there when needed." Similarly, an ECMWF survey [9] identifies user dissatisfaction regarding data turnaround of numerical weather prediction (NWPs) that are used as input to short-term renewable production forecasting. But even if data are typically provided in a timely fashion, data availability is not 100% guaranteed, and a robust fallback solution is always desirable if not necessary. Notably, however, uncertainty *w.r.t.* data availability is largely overlooked in the energy forecasting literature.

B. Literature Review

Missing features at test time is a subject that receives scarce attention, contrary to missing data during training, which can be addressed with techniques such as Multiple Imputation [10] or can be directly embedded within the learning model [11],

[12]. In wind power forecasting, [13] examines two methods to handle missing features, namely retraining without the missing features and imputation. Retraining consistently outperforms imputation and the difference is more pronounced when data are missing in batches. However, the number of additional models required is the combination of all features, which renders retraining impractical. Similarly, [14] develops several models to forecast electricity demand at a household level; given data availability at test time, the appropriate model is selected from a decision tree. The same approach, i.e., training several models to deal with uncertain data availability, is also considered in [15] to directly forecast the trading decisions of a renewable producer participating in a day-ahead market. An integrated imputation procedure to replace missing features within a long short-term memory network for solar production forecasting is presented in [16]; the performance, however, deteriorates as the percentage of missing values increases, and no comparison against retraining is provided.

A related stream of research examines energy forecasting under data-integrity attacks, mostly dealing with uncertainty in the target variable and focusing on training data. Several load forecasting models are evaluated in [17] against attacks that affect the training process by permutating historical observations; none of the models considered provides adequate performance under large-scale attacks. A subsequent work [18] leverages robust statistics and shows that selecting the ℓ_1 norm as the loss function proves resilient even under large-scale attacks. Similarly, [19] studies the robustness of short-term wind production forecasting models under false-data injection attacks, considering both point and probabilistic forecasts. Conversely, [20] formulates a poison attack methodology to exploit load forecasting models. Tangentially related to data-integrity attacks on load forecasting are works on outlier detection [21], [22], [23], which focus on identifying attacks that have occurred and replacing any corrupt data. On the other hand, [24] and [25] consider adversarial attacks at test time applied to load forecasting. Specifically, [24] shows that manipulating temperature values at test time leads to a significant decrease in accuracy and increased operational costs, whereas [25] employs Bayesian learning to enhance the robustness of deep-learning-based models under several adversarial attacks.

One way to view data-integrity attacks is as processes that introduce feature uncertainty; the same also applies to the case of missing features. Indeed, advanced forecasting models are typically cognizant of some form of feature uncertainty, even if this is unknown to the forecaster, and address it with regularization, e.g., ℓ_1-regularized (Lasso) regression [26] or ℓ_2-regularized (ridge) regression. Introducing randomness during training also enhances model robustness; popular methods include bagging and sampling a subset of features, as in randomized ensembles such as Random Forests [27], using dropout layers in deep learning models, and generative adversarial networks, among others. In fact, [25] shows that regularization and treating model parameters as random variables increase robustness in load forecasting applications. Interestingly, a big part of the success of regularization methods is their "hidden" robustness. For example, both the ℓ_1-regularized [28] and the ℓ_2-regularized [29] regressions are equivalent to the solution of robust optimization problems [30]. Beyond regularized regression, several applications of robust optimization in different machine learning areas exist [31], such as classification [32] and deep learning [33]. We highlight [34], which describes a robust learning support vector machine algorithm for classification where a different set of features might be missing at each observation, as a core foundation of our current work. Uniform feature deletion, i.e., the same features missing across all observations, is considered as an alternate setting in [34], which is deemed as not efficiently solvable, except for a small number of features through enumeration. Notably, the connection between feature uncertainty, robust optimization, and regularization is rarely discussed in the context of energy forecasting.

C. Aim and Contribution

In this work, we present a robust optimization approach to design energy forecasting models that are optimally resilient when a subset of features used for model training becomes unavailable at test time. We formulate a robust regression model, readily applicable to point and probabilistic forecasting, which minimizes the worst-case loss when a subset of features is missing. We present three solution methods for the resulting robust optimization formulation considering the quantile loss, all leading to Linear Programming (LP) problems: *(i)* a method based on enumeration, which is practical for a small number of features; *(ii)* a deterministic reformulation, which, although tractable, provides conservative results thus being more suitable for the main setting of [34] with different features missing across observations, and *(iii)* an affinely adjustable reformulation [35], which offers an efficient solution method to the uniform feature deletion setting of [34], remains tractable, and is less conservative than the previous method. We further consider extensions to piecewise linear loss functions, which can be used to approximate quadratic, and in general convex, loss functions, whereas [34] only considers the hinge loss. We evaluate the proposed methods in prevalent applications, namely electricity price, load, wind production, and solar production, forecasting, considering a day-ahead horizon. We further compare the proposed approach against established benchmark models, including regularization and randomization-based training, coupled with different methods of handling missing data, i.e., imputation and retraining. We demonstrate that the proposed approach outperforms imputation-based models and exhibits similar performance to retraining without the missing features, while preserving practicality. Notably, by evenly distributing coefficient weights across features during training, it hedges against missing the most important feature at test time. Preliminary results of this work were presented at [36].

Our main contribution is two-fold. Firstly, we propose a robust regression model that is, by design, resilient against missing features at test time, with the following key advantages: *(i)* consistent performance and lower model degradation when features are missing, including the worst-case scenario of missing the most important feature, and *(ii)* computational

tractability through LP reformulations, which can also be applied to approximations of quadratic, and in general convex, loss functions. Secondly, we benchmark against current state-of-the-art forecasting models and methods to handle feature uncertainty for both point and probabilistic forecasting, and quantify the aforementioned advantages in several prevalent energy forecasting applications. To the best of our knowledge, this is the first work that introduces resilient energy forecasting and benchmarks against missing features at test time, a situation that may emerge in industrial applications after the forecasting model is deployed.

D. Paper Organization

The remainder of the paper is organized as follows. Section II presents the mathematical background and the proposed model. Section III develops the solution methodology. Section IV presents the experimental setup and the input data, and Section V discusses the numerical results. Section VI concludes and provides directions for future work.

II. Preliminaries and Proposed Model

In this section, we present preliminaries on linear regression (in Subsection II-A), describe the process of modeling feature uncertainty (in Subsection II-B), and present the proposed robust formulation (in Subsection II-C).

A. Preliminaries

Let $y_i \in \mathbb{R}$ be the target variable (e.g., electricity prices, load, wind/solar production) and $\boldsymbol{x}_i \in \mathbb{R}^p$ be a p-size vector of associated features from a set $\mathcal{P} = \{1, \dots, p\}$ (e.g., weather data, historical data), with subscript i denoting an observation from a training data set $\{(y_i, \boldsymbol{x}_i)\}_{i=1}^n$ of n observations. Throughout, the term $[n]$ is used as shorthand for $\{1, \dots, n\}$. A regression model is defined as a mapping function $f \in \mathcal{F} : \boldsymbol{x} \in \mathbb{R}^p \rightarrow y \in \mathbb{R}$, where $\mathcal{F}$ is a hypothesis space. Here, we focus exclusively on linear models parameterized by a set of coefficients $\boldsymbol{w} \in \mathbb{R}^p$. To ease the notation, we assume that the bias term is modeled by appending a constant vector of ones to $\boldsymbol{x}$. The problem of estimating the parameters of a linear regression model is given by:

$$\min_{\boldsymbol{w}} \sum_{i \in [n]} l(y_i - \boldsymbol{w}^\intercal \boldsymbol{x}_i), \tag{1}$$

where l is the selected loss function to be minimized [1]. Typical choices are the quadratic loss $l(\cdot) = (\cdot)^2$ (least squares or LS) and the ℓ_1 norm $l(\cdot) = |\cdot|$ (least absolute deviations or LAD).

Both the LS and the LAD models are employed to derive point estimates of the target variable. Dealing, however, with uncertainty necessitates the usage of probabilistic forecasts as an input in many decision-making processes. Quantile regression (QR) [37] is a general approach to derive probabilistic forecasts in the form of predictive quantiles. A QR model minimizes the quantile (pinball) loss for a specific quantile τ, defined as:

$$\begin{aligned} l(y_i - \boldsymbol{w}^\intercal \boldsymbol{x}_i; \tau) &= \tau(y_i - \boldsymbol{w}^\intercal \boldsymbol{x}_i)^+ + (1-\tau)(\boldsymbol{w}^\intercal \boldsymbol{x}_i - y_i)^+ \\ &= \max\left(\tau(y_i - \boldsymbol{w}^\intercal \boldsymbol{x}_i), (\tau - 1)(y_i - \boldsymbol{w}^\intercal \boldsymbol{x}_i)\right), \end{aligned} \tag{2}$$

where $(t)^+ = \max(0, t)$. In fact, the ℓ_1 loss can be viewed as a special case of the quantile loss estimating the 50th quantile (median). This is straightforward to show considering that $|x| = \max(x, -x)$, $\tau = 0.5$, and that scaling the objective does not affect the solution.

B. Modeling Feature Uncertainty

Our goal is to formulate a robust regression model, which accounts for missing features after model deployment (i.e., at test time) and maintains consistent performance. To this end, we introduce binary variables $\boldsymbol{\alpha} \in \{\mathbf{0}, \mathbf{1}\}^p$ and model the availability of the i-th feature observation as $\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})$, where $\odot$ is the element-wise multiplication, and α_j equals 1 if the j-th feature is missing (i.e., missing features are set to zero).

At this point, there are two issues that relate to energy forecasting applications that warrant a discussion.

First, in practice, some features cannot be deleted at test time. It makes little sense to delete, e.g., calendar variables, which are regularly employed in energy forecasting. Let $\mathcal{J} \subseteq \mathcal{P}$ denote the subset of features that *can* be deleted at test time, and $\mathcal{C} = \mathcal{P} - \mathcal{J}$ denote the set of features that *cannot* be deleted. It is straightforward to account for this case by setting $\alpha_j = 0 \ \forall j \in \mathcal{C}$, therefore features in $\mathcal{C}$ cannot go missing.

Second, a standard technique to model nonlinear relationships within a linear regression is to include polynomial and interaction terms of associated features. A classic example in energy forecasting is to add quadratic and cubic terms of temperature in load forecasting models [38]. It follows that all features derived from the same variable should be treated as a group of features (i.e., if missing, they are all missing).

We address both the aforementioned issues by enforcing a set of equality constraints, $\boldsymbol{M}\boldsymbol{\alpha} = \mathbf{0}$, where $\boldsymbol{M} \in \mathbb{R}^{m \times p}$. Namely, if the first feature cannot be deleted (i.e., $\alpha_1 = 0$), then the row vector $[1, \mathbf{0}]$ is appended to $\boldsymbol{M}$. Similarly, if $\alpha_1 = \alpha_2$, i.e., they represent a group of features, then $[-1, 1, \mathbf{0}]$ is appended to $\boldsymbol{M}$.

Following the above, we consider the discrete uncertainty set:

$$\mathcal{U} = \{\boldsymbol{\alpha} \ \big| \ \boldsymbol{\alpha} \in \{\mathbf{0}, \mathbf{1}\}^p, \sum_{j \in [p]} \alpha_j = \Gamma, \boldsymbol{M}\boldsymbol{a} = \mathbf{0}\}, \tag{3}$$

that models the representation of feature availability, where Γ (integer) is the budget of robustness (for $\Gamma = 0$, all features are present, whereas for $\Gamma = p$ all features are missing).

C. Proposed Robust Formulation

The proposed robust formulation employs the representation of the availability of the i-th feature observation $\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})$,

[1] Note that the linear regression model can straightforwardly accommodate nonlinear dependencies by considering polynomial terms, local weights, etc.

and builds a robust regression model using the uncertainty set (3), as follows:

$$\min_{\boldsymbol{w}} \max_{\boldsymbol{\alpha}\in\mathcal{U}} \sum_{i\in[n]} l\big(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha}))\big). \tag{4}$$

Inspired by [34], we refer to model (4) as *feature-deletion robust regression* (*FDRR*). The problem objective is to minimize the worst-case loss when Γ features are missing, assuming that the *same* features are missing across all observations[2], while also respecting additional constraints arising from the fact that a subset of features could not be deleted or that different features might be grouped. In the latter case, Γ is selected appropriately to account for feature groups.

Interestingly, minimizing the worst-case loss when a subset of features is missing (4) shares many similarities with feature selection and feature importance. On one hand, feature selection concerns methods to improve out-of-sample predictive accuracy by optimally selecting a feature vector. Usually, this involves gradually adding features to the model. Intuitively, a feature that improves performance will also have significant impact when deleted; however, the problems are not equivalent. Feature importance, on the other hand, concerns post-hoc methods to assess the individual feature contribution to model performance, with the goal to improve explainability — see, e.g., the permutation importance metric proposed in [27]. Notably, our proposal effectively optimizes the model based on feature importance by design.

Next, we consider (4), using the quantile loss, which, along with its special case — the ℓ_1 loss — are of particular interest in energy forecasting applications. Hence, using the quantile loss representation (2) in (4), we obtain the following robust optimization problem:

$$\min_{\boldsymbol{w}} \max_{\boldsymbol{\alpha}\in\mathcal{U}} \sum_{i\in[n]} \max\Big(\tau(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha}))), (\tau-1)(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha})))\Big). \tag{5}$$

Note that setting $\tau = 0.5$ and scaling the objective would yield the robust formulation for the ℓ_1 regression:

$$\min_{\boldsymbol{w}} \max_{\boldsymbol{\alpha}\in\mathcal{U}} \sum_{i\in[n]} \big|y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha}))\big|.$$

For practical reasons, we can recast (5) using a robust constraint, introducing auxiliary $t \in \mathbb{R}$, as follows:

$$\min_{\boldsymbol{w},t} t, \tag{6a}$$

$$\text{s.t.} \sum_{i\in[n]} \max\Big(\tau(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha}))), (\tau-1)(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha})))\Big) \le t,\ \forall \boldsymbol{\alpha}\in\mathcal{U}, \tag{6b}$$

which involves an inequality that contains the sum of maxima of linear functions. Indeed, in a deterministic setting, i.e., in the absence of $\forall\boldsymbol{\alpha}\in\mathcal{U}$, constraint (6b) could be straightforwardly and, most importantly exactly, reformulated using auxiliary variables. Consider a specific instance of $\boldsymbol{\alpha}$, say $\boldsymbol{\alpha}_k$. Then, the deterministic reformulation of (6b) would be:

$$\min_{\boldsymbol{w},t,\boldsymbol{\xi}} t, \tag{7a}$$

$$\text{s.t.} \sum_{i\in[n]} \xi_i \le t, \tag{7b}$$

$$\tau(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha}_k))) \le \xi_i, \quad i\in[n], \tag{7c}$$

$$(\tau-1)(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1}-\boldsymbol{\alpha}_k))) \le \xi_i, \quad i\in[n], \tag{7d}$$

where $\xi_i \in \mathbb{R}$ is an auxiliary variable, and $\boldsymbol{\xi}$ an appropriate vector. However, care must be given when applying deterministic reformulations in a robust setting, as they could lead to over-conservative solutions [35]. It is interesting to note that (7) is essentially equivalent to "retraining" for a specific combination of missing features. In fact, repeating (7) for all elements of all sets $\mathcal{U}$ constructed by the admissible values of $\Gamma = \{0,\dots,|\mathcal{J}|\}$ retrieves the solution proposed in [13], [14], [15], i.e., retraining without the missing features.

Before proceeding to the solution methods of (6), let us revisit the uncertainty set, $\mathcal{U}$, and consider its convex hull, represented by the polyhedral uncertainty set, $\mathcal{A}$,

$$\mathcal{A} = \{\boldsymbol{\alpha} \mid \mathbf{0} \le \boldsymbol{\alpha} \le \mathbf{1}, \sum_{j\in[p]} \alpha_j = \Gamma, \boldsymbol{M}\boldsymbol{a} = \mathbf{0}\}. \tag{8}$$

Note that $\boldsymbol{M}$ is unimodular, as all of its entries are $0, 1$ or -1, and at most two entries per column are non-zero, at which case the column-wise sum is zero. Since Γ is also integer, all vertices of $\mathcal{A}$ occur at integer values, therefore the LP relaxation of the inner $\max$ problem over $\boldsymbol{\alpha}$ in (5) is exact. Evidently, replacing $\mathcal{U}$ by its convex hull $\mathcal{A}$ in constraint (6b) also yields equivalent solutions [39, Ch. 10].

III. Solution Methods

In this section, we present three methods to solve the robust optimization problem (6). In Subsection III-A, we describe a method suitable for a small number of features, whereas in Subsections III-B and III-C we present reformulations that lead to tractable problems. Finally, in Subsection III-D, we discuss an extension to piecewise linear loss functions.

A. Vertex Enumeration of FDRR (FDRR-V)

Typically, most energy forecasting problems have relatively large sample sizes (e.g., n is in the order of 10^4 for series with hourly resolution) compared to the number of features, i.e., $n >> p$. Hence, if the number of features is small, problem (6) could be solved by vertex enumeration of the uncertainty set $\mathcal{A}$. In fact, since all vertices of $\mathcal{A}$ are contained in the original finite set $\mathcal{U}$, vertex enumeration of $\mathcal{A}$ is equivalent to an enumeration of the elements of $\mathcal{U}$.

Let V denote the number of elements of $\mathcal{U}$, equivalently the number of vertices of $\mathcal{A}$; assuming no grouping constraints, $V = \binom{|\mathcal{J}|}{\Gamma}$ (grouping constraints would further reduce V). Let

[2]Note that [34] considers the case where *different* features are missing across observations, which leads to a more conservative problem.

ξ_i^k be an auxiliary variable, for each $i \in [n]$ and each vertex $k \in [V]$. Constraint (6b) is equivalently written as:

$$\sum_{i\in[n]} \xi_i^k \le t, \qquad k \in [V], \quad (9a)$$

$$y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}_k)) \le \tfrac{1}{\tau}\xi_i^k, \;\; i \in [n], k \in [V], \quad (9b)$$

$$-y_i + \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}_k)) \le \tfrac{1}{1-\tau}\xi_i^k, i \in [n], k \in [V], \quad (9c)$$

where constraints (9a)–(9c) essentially enumerate the deterministic reformulation (7b)–(7d) for all vertices. Hence, the solution of FDDR by vertex enumeration, referred to as FDRR-V, is given by the following deterministic LP:

$$\text{FDRR-V:} \quad \min_{\boldsymbol{w},t,\boldsymbol{\xi}} t, \quad \text{s.t.} \quad (9a) - (9c), \qquad (10)$$

where $\boldsymbol{\xi}$ is an appropriate vector that represents variables ξ_i^k. FDRR-V ensures that the worst-case $\boldsymbol{\alpha}$ remains the same across all observations and leads to an exact solution of (6). Evidently, for a specific realization of uncertainty, say $\boldsymbol{\alpha}_k$, retraining — see (7) — sets a lower bound to the in-sample error of FDRR-V, which subsumes all individual cases. However, if the number of features is not small enough, unavoidably V gets large enough to render both retraining and FDRR-V at least impractical, in terms of models to be trained and LPs to be solved, respectively.

B. Reformulation of FDRR (FDRR-R)

An alternative approach is to first apply deterministic reformulation to the maxima terms in (6b), leading to:

$$\sum_{i\in[n]} \xi_i \le t, \qquad (11a)$$

$$y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) \le \tfrac{1}{\tau}\xi_i, \; i \in [n], \forall \boldsymbol{\alpha} \in \mathcal{A}, \quad (11b)$$

$$-y_i + \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) \le \tfrac{1}{1-\tau}\xi_i, \; i \in [n], \forall \boldsymbol{\alpha} \in \mathcal{A}. \quad (11c)$$

In turn, (11b)-(11c) are further reformulated to deterministic constraints. Since both constraints are similar, we illustrate the reformulation for (11b).

Changing the order of multiplication in the left-hand side (*lhs*) of (11b), and considering that the inequality holds $\forall \boldsymbol{\alpha} \in \mathcal{A}$, i.e., the worst-case of $\boldsymbol{\alpha}$, constraint (11b) is equivalent to:

$$y_i - \boldsymbol{w}^\intercal \boldsymbol{x}_i + \max_{\boldsymbol{\alpha}\in\mathcal{A}} (\boldsymbol{w} \odot \boldsymbol{x}_i)^\intercal \boldsymbol{\alpha} \le \tfrac{1}{\tau}\xi_i, \quad i \in [n]. \qquad (12)$$

The inner $\max$ in (12) can be written with explicit constraints, for the i-th observation, as follows:

$$\max_{\boldsymbol{\alpha}} \quad (\boldsymbol{w} \odot \boldsymbol{x}_i)^\intercal \boldsymbol{\alpha}, \qquad (13a)$$

$$\text{s.t.} \quad \boldsymbol{\alpha} \le \mathbf{1} : \;\; \boldsymbol{\mu}_i^+ \ge \mathbf{0}, \qquad (13b)$$

$$\sum_{j\in[p]} \alpha = \Gamma : \;\; \zeta_i^+, \qquad (13c)$$

$$\boldsymbol{M}\boldsymbol{\alpha} = \mathbf{0} : \;\; \boldsymbol{\pi}_i^+, \qquad (13d)$$

$$\boldsymbol{\alpha} \ge \mathbf{0}, \qquad (13e)$$

where $\boldsymbol{\mu_i^+}, \zeta_i^+, \pi_i^+$ are dual variables of appropriate size. Since problem (13a) is linear in $\boldsymbol{\alpha}$, it can be replaced by its dual:

$$\min_{\boldsymbol{\mu}_i^+ \ge \mathbf{0}, \zeta_i^+, \boldsymbol{\pi}_i^+} \sum_{j\in[p]} \mu_{ij}^+ + \Gamma\zeta_i^+, \qquad (14a)$$

$$\text{s.t.} \quad \boldsymbol{\mu}_i^+ + \zeta_i^+ + \boldsymbol{M}^\intercal \boldsymbol{\pi}_i^+ \ge \boldsymbol{x}_i \odot \boldsymbol{w}, \qquad (14b)$$

and hence, the inner $\max$ in (12) can be replaced by (14). Evidently, the $\min$ operator becomes redundant. Hence, constraint (11b) is replaced by the following inequalities:

$$y_i - \boldsymbol{w}^\intercal \boldsymbol{x}_i + \sum_{j\in[p]} \mu_{ij}^+ + \Gamma\zeta_i^+ \le \tfrac{1}{\tau}\xi_i, \; i \in [n], \qquad (15a)$$

$$\boldsymbol{\mu}_i^+ + \zeta_i^+ + \boldsymbol{M}^\intercal \boldsymbol{\pi}_i^+ \ge \boldsymbol{x}_i \odot \boldsymbol{w}, \; i \in [n], \qquad (15b)$$

$$\boldsymbol{\mu}_i^+ \ge \mathbf{0}, \; i \in [n]. \qquad (15c)$$

Similarly, constraint (11c) is replaced by:

$$-y_i + \boldsymbol{w}^\intercal \boldsymbol{x}_i + \sum_{j\in[p]} \mu_{ij}^- + \Gamma\zeta_i^- \le \tfrac{1}{1-\tau}\xi_i, \; i \in [n], \qquad (15d)$$

$$\boldsymbol{\mu}_i^- + \zeta_i^- + \boldsymbol{M}^\intercal \boldsymbol{\pi}_i^- \ge -\boldsymbol{x}_i \odot \boldsymbol{w}, \; i \in [n], \qquad (15e)$$

$$\boldsymbol{\mu}_i^- \ge \mathbf{0}, \; i \in [n]. \qquad (15f)$$

Summarizing, the reformulation of the FDRR, referred to as FDRR-R, yields the following deterministic LP:

$$\text{FDRR-R:} \quad \min_{\substack{\boldsymbol{w},t,\boldsymbol{\xi},\\ \boldsymbol{\mu}^+,\boldsymbol{\mu}^-,\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-,\boldsymbol{\pi}^+,\boldsymbol{\pi}^-}} t, \quad \text{s.t.} \quad (15a) - (15f). \quad (16)$$

Note, however, that the uncertainty is now spread over several constraints, separately optimizing the worst-case loss of each observation. This worst-case loss may occur for different $\boldsymbol{\alpha}$ per observation, i.e., different features might be missing at each observation, which leads to the representation of uncertainty considered in [34]. When modeling feature uncertainty in Section II-B, however, we assumed the same $\boldsymbol{\alpha}$ across all observations. Evidently, FDRR-R considers a more general case and thus provides a conservative approximation of (6), which is more pessimistic.

C. Affinely Adjustable Reformulation of FDRR (FDRR-AAR)

The conservativeness introduced by the reformulation of the maxima terms is reduced using adjustable auxiliary variables [35]. As ξ_i is not a true decision variable, it may be adjusted to the realization of $\boldsymbol{\alpha}$ as long as inequalities (11b) and (11c) hold. To this end, we introduce linear decision rules $v_i \in \mathbb{R}, \boldsymbol{u}_i \in \mathbb{R}^p$, and substitute $\xi_i = v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}$, i.e., ξ_i is an affine function of uncertainty. Constraint (6b) is written as:

$$\sum_{i\in[n]} (v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}) \le t, \qquad \forall \boldsymbol{\alpha} \in \mathcal{A}, \quad (17a)$$

$$y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) \le \tfrac{1}{\tau}(v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}), \; i \in [n], \; \forall \boldsymbol{\alpha} \in \mathcal{A}, \quad (17b)$$

$$-y_i + \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) \le \tfrac{1}{1-\tau}(v_i + \boldsymbol{u}_i^\intercal \boldsymbol{\alpha}), \; i \in [n], \; \forall \boldsymbol{\alpha} \in \mathcal{A}, \quad (17c)$$

Similarly to (12), constraint (17a) is equivalent to:

$$\sum_{i\in[n]} v_i + \max_{\boldsymbol{\alpha}\in\mathcal{A}} \sum_{i\in[n]} \boldsymbol{u}_i^\intercal \boldsymbol{\alpha} \le t,$$

and introducing dual variables $\boldsymbol{\mu}$, ζ, and $\boldsymbol{\pi}$ (similarly to (13b), (13c), and (13d), respectively), constraint (17a) is replaced by:

$$\sum_{i\in[n]} v_i + \sum_{j\in[p]} \mu_j + \Gamma\zeta \leq t, \tag{18a}$$

$$\boldsymbol{\mu} + \zeta + \boldsymbol{M}^\intercal\boldsymbol{\pi} \geq \sum_{i\in[n]} \boldsymbol{u}_i, \tag{18b}$$

$$\boldsymbol{\mu} \geq \mathbf{0}. \tag{18c}$$

Constraint (17b) is equivalent to:

$$y_i - \boldsymbol{w}^\intercal\boldsymbol{x}_i + \max_{\boldsymbol{\alpha}\in\mathcal{A}}(\boldsymbol{x}_i \odot \boldsymbol{w} - \tfrac{1}{\tau}\boldsymbol{u}_i)^\intercal\boldsymbol{\alpha} \leq \tfrac{1}{\tau}v_i,\ i\in[n],$$

and similarly to (12), constraint (17b) is replaced by:

$$y_i - \boldsymbol{w}^\intercal\boldsymbol{x}_i + \sum_{j\in[p]} \mu_{ij}^+ + \Gamma\zeta_i^+ \leq \tfrac{1}{\tau}v_i, \quad i\in[n], \tag{19a}$$

$$\boldsymbol{\mu}_i^+ + \zeta_i^+ + \boldsymbol{M}^\intercal\boldsymbol{\pi}_i^+ \geq \boldsymbol{x}_i \odot \boldsymbol{w} - \tfrac{1}{\tau}\boldsymbol{u}_i, \quad i\in[n], \tag{19b}$$

$$\boldsymbol{\mu}_i^+ \geq \mathbf{0}, \quad i\in[n], \tag{19c}$$

whereas constraint (17c) is replaced by:

$$-y_i + \boldsymbol{w}^\intercal\boldsymbol{x}_i + \sum_{j\in[p]} \mu_{ij}^- + \Gamma\zeta_i^- \leq \tfrac{1}{1-\tau}v_i, \quad i\in[n], \tag{20a}$$

$$\boldsymbol{\mu}_i^- + \zeta_i^- + \boldsymbol{M}^\intercal\boldsymbol{\pi}_i^- \geq -\boldsymbol{x}_i \odot \boldsymbol{w} - \tfrac{1}{1-\tau}\boldsymbol{u}_i, \quad i\in[n], \tag{20b}$$

$$\boldsymbol{\mu}_i^- \geq \mathbf{0}, \quad i\in[n]. \tag{20c}$$

Lastly, the affinely adjustable reformulation of the FDRR (FDRR-AAR) is equivalent to the following deterministic LP:

$$\text{FDRR-AAR:} \quad \min_{\substack{\boldsymbol{w},t,\boldsymbol{v},\boldsymbol{u},\boldsymbol{\mu},\boldsymbol{\mu}^+,\boldsymbol{\mu}^-,\\ \zeta,\boldsymbol{\zeta}^+,\boldsymbol{\zeta}^-,\boldsymbol{\pi},\boldsymbol{\pi}^+,\boldsymbol{\pi}^-}} t, \quad \text{s.t.} \quad (18a)-(20c). \tag{21}$$

Note that we are still optimizing over the worst-case loss per observation, hence FDRR-AAR is a conservative approximation of (6). However, allowing for adjustable auxiliary variables reduces the induced conservativeness compared to FDRR-R. On the other hand, FDRR-AAR leads to a tractable LP, contrary to FDRR-V that leads to an LP whose size grows combinatorially. This trade-off between tractability and conservativeness places FDRR-AAR as an intermediate solution between FDRR-V and FDRR-R.

D. Extension to Piecewise Linear Loss Functions

In what follows, we discuss an extension of our proposal to piecewise linear loss functions.

Consider a piecewise linear loss function

$$l(y - \boldsymbol{w}^\intercal\boldsymbol{x}; \boldsymbol{c}, \boldsymbol{b}) = \max_{j=1,\ldots,m}(c_j(y - \boldsymbol{w}^\intercal\boldsymbol{x} + b_j)), \tag{22}$$

parameterized by the m-size vectors $\boldsymbol{c}, \boldsymbol{b}$. Note that the quantile loss is a special case of (22), where $m = 2$, $\boldsymbol{c} = [\tau,\ \tau-1]^\intercal$, and $\boldsymbol{b} = \mathbf{0}$. Using the piecewise linear loss function (22), the FDRR model (6) becomes

$$\min_{\boldsymbol{w},t}\ t,$$

$$\text{s.t.} \sum_{i\in[n]} \max_{j\in[m]}\Big(c_j(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) + b_j)\Big) \leq t, \quad \forall\boldsymbol{\alpha}\in\mathcal{U},$$

which can be solved with any of the proposed solution methods. For the solution with vertex enumeration, FDRR-V, we enumerate the deterministic reformulation for all vertices and all m vectors; hence, (9b)-(9c) are replaced by

$$c_j(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha}_k)) + b_j) \leq \xi_i^k,\ \ i\in[n], \quad k\in[V], j\in[m].$$

For FDRR-R, (11b)-(11c) are replaced by

$$c_j(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) + b_j) \leq \xi_i,\ \ i\in[n],\ j\in[m], \quad \forall\boldsymbol{\alpha}\in\mathcal{A},$$

which are further reformulated to deterministic constraints similarly to (11b)-(11c) — see (15). For FDRR-AAR, (17b)-(17c) are replaced by

$$c_j(y_i - \boldsymbol{w}^\intercal(\boldsymbol{x}_i \odot (\mathbf{1} - \boldsymbol{\alpha})) + b_j) \leq (v_i + \boldsymbol{u}_i^\intercal\boldsymbol{\alpha}),\ i\in[n], \quad j\in[m], \forall\boldsymbol{\alpha}\in\mathcal{A},$$

which are further reformulated similarly to (11b)-(11c) — see (19) and (20).

The piecewise linear loss functions can be used to approximate quadratic, and in general convex, loss functions. Consider for example an FDRR model with a quadratic loss (LS). It is straightforward to solve the robust regression model with vertex enumeration, but this approach is only be practical for a small number of features. For a larger number of features, it is not straightforward to reformulate the robust problem, as the quadratic loss leads to robust constraints that are quadratic in $\boldsymbol{\alpha}$ and thus more challenging to handle — see [40, Ch. 16]. Hence, a reasonable approach would be to use a piecewise linear function to approximate the quadratic loss and solve the resulting robust problem as described above. In general, the piecewise linearization becomes relevant in first order approximations of the loss function, e.g., in the context of adversarial training [41].

IV. Experimental Setup and Input Data

In this section, we present the setup of our numerical experimentation (in Subsection IV-A) and list the input data for several energy forecasting applications (in Subsection IV-B).

A. Experimental Setup

Our experimental setup involves four prevalent day-ahead energy forecasting applications, namely *(i)* electricity price, *(ii)* load, *(iii)* wind production, and *(iv)* solar production, forecasting. First, we select a set of features that lead to good performance in a linear regression model following known best practices. We then train several benchmarks with the same set of features, including both linear regression models and machine learning models with randomization-based training (e.g., Random Forest),[3] which are known to perform well in energy forecasting applications. We compare their out-of-sample performance under feature deletion to the proposed

[3]We opt for tree-based ensembles over other machine learning models (e.g., neural networks) as they showcase exceptionally good performance in regression settings with minimal tuning effort, which makes them ideal benchmarks [42].

FDRR and retraining without the missing features. Evidently, our goal is not to search for improved forecast accuracy, but rather for resilient energy forecasting, i.e., to examine the robustness of the models.

For point forecasting, we test the following models:

- `LS`: LS regression.
- `LAD`: LAD regression.
- `LS-`$\ell_1\backslash\ell_2$: LS regression with ℓ_1 (lasso) or ℓ_2 (ridge) regularization penalties.
- `RF`: Random Forest.
- `RETRAIN` [13]: It involves retraining an LAD model for each combination of missing features, in total $\sum_{k=0}^{|\mathcal{J}|}\binom{|\mathcal{J}|}{k}$ times. To facilitate comparisons with the proposed approach, we use LAD instead of LS models to derive equivalent performance when Γ is 0 or $|\mathcal{J}|$.
- `FDRR`(Γ): Robust regression with ℓ_1 loss, and robustness budget Γ.

For probabilistic forecasting, we test the following models:

- `QR`: Quantile regression.
- `QR-`ℓ_1: Quantile regression with ℓ_1 regularization.
- `QRF`: Quantile Regression Forests [43], a generalization of Random Forests.
- `FDRR`(Γ): Robust regression with quantile loss, and robustness budget Γ.

For the models that cannot handle missing values directly, i.e., `LS`-type, `LAD`, `RF`, `QR`-type, and `QRF`, we follow the impute-then-regress approach with mean imputation, setting missing features at their in-sample mean. We purposefully choose mean imputation as a simple method that is suitable for an operational setting, [4] thus avoiding complicated and computationally costly methods, which may not add in terms of predictive performance — see e.g., [44] for a discussion in a similar context with missing data. For `LS-`$\ell_1\backslash\ell_2$ and `RF`, we use 5-fold cross-validation on the training data for hyperparameter tuning. We select the hyperparameters with lowest cross-validation error via grid search, and we retrain each model using the full train set. The same hyperparameter values are subsequently used in the probabilistic case for `QR-`ℓ_1 and `QRF`, respectively. For `FDRR`(Γ) missing values are set to zero, and a different model is trained for each value of Γ. To ease the notation, `FDRR` refers to the group of models trained over all Γ. Clearly, as the number of missing features is known prior to deriving out-of-sample forecasts, we use `FDRR` with Γ set at the exact number of missing features. By definition, `FDRR`(0) is equivalent to an LAD model. In addition, `FDRR` and `RETRAIN` are equivalent for $\Gamma = 0$ and $\Gamma = |\mathcal{J}|$. In all cases, data are scaled between $[0, 1]$ prior to training. Lastly, to derive probabilistic forecasts a different model is trained per quantile τ in all cases except for `QRF`.

To evaluate performance we use standard error metrics. For point forecasting, we use the mean absolute error (MAE) for electricity price and wind/solar production (both normalized *w.r.t.* nominal capacity), and the mean absolute percentage error (MAPE) for load. For probabilistic forecasting, we use the average pinball loss on 9 equally spaced quantiles, i.e., $\tau \in \{0.1, \ldots, 0.9\}$.

[4] In practice, missing data might be replaced by correlated features (which may have been removed during feature selection), if such are available, e.g., data from nearby locations. Practitioners may also apply imputation methods that rely on their experience, whose performance is assessed empirically for a specific forecasting application.

TABLE I
OVERVIEW OF THE DATA SETS.

Data set (# series)	Source	n	$\lvert\mathcal{P}\rvert$	$\lvert\mathcal{J}\rvert$
Electricity Prices (1)	[45]	13140	9	5
Load (21)	[46]	16200	625	4×111
Wind (10)	[47]	8807	13	2×4
Solar (3)	[47]	8784	13	12

B. Input Data for Energy Forecasting Applications

Table I provides an overview of the selected data sets. For each energy forecasting application, it shows the number of series, the source, the training sample size, n, and the sizes of the sets $\mathcal{P}$ and $\mathcal{J}$. Note that the bias (intercept) term is included in $\mathcal{P}$ and cannot be deleted. Further, all cases involve features that capture seasonality and cannot be deleted. Thus, when $\Gamma = |\mathcal{J}|$, FDRR leads to a model that captures the seasonal component of each series.

1) Electricity Prices: We use hourly data from the French electricity market, spanning the period 2017-2019, with a $50/50$ train/test split. Features include calendar variables (cannot be deleted), historical price lags, and TSO published data, namely net load forecast (demand minus renewable production) and system margin (ratio of net load and available thermal generation). For historical lags, we examine the partial autocorrelation function and select lags that are significant at the 5% level.

2) Load: We use data from GEFCom 2012 [46], comprising 4.5 years of hourly load and temperature data from a US utility with 21 zones. Following [17], [18], 3 full years of data are used with a $75/25$ train/test split. We construct the input feature vector according to the vanilla model [38], which includes a linear trend, calendar variables (one-hot encoded), polynomial terms of temperature, and interaction terms of the above, with a total of 292 features. We consider 4 distinct groups of features based on temperatures from different stations and examine performance under group deletion; this leads to 625 features in total and 111 features per group. Clearly, the subset of features that cannot be deleted (trend and calendar variables) is included only once. The results presented concern zone 21 (aggregate demand) using temperatures from stations 1-3 and a fictitious station with the average temperature across all stations.

3) Wind Production: We use data from GEFCom2014 [47], comprising 2 years of hourly production data from 10 wind farms, and apply a $50/50$ train/test split. Following [19], the selected features include wind speed forecasts, with quadratic and cubic terms, wind direction forecasts (both at 10m and 100m), and Fourier terms to model the diurnal patterns (these cannot be deleted). Forecasts of both wind speed and direction are derived from forecasts of the U- and V-speed components for each height level; thus, if either is missing, all derivative

features will be missing. We consider two groups of features that include wind speed and wind direction at 10m and 100m and assume that these can be missing independently. The results presented concern zone 1 of the data set.

4) Solar Production: We use data from GEFCom2014 [47], comprising 2 years of hourly production data from 3 PV plants located in Australia and 12 NWP variables, including precipitation, solar radiation, and temperature — see [47] for detail, and apply a $50/50$ train/test split. We train a separate model for each hour of the day (except for `RF`, `QRF`) using the respective NWPs as input features, and assume that each NWP variable could be missing independently. Only hours with non-zero solar radiation are considered. The results presented concern zone 1 of the data set.

V. Numerical Results

In this section, we evaluate the FDRR solution methods (in Subsection V-A), we compare FDRR with various benchmarks (in Subsection V-B), and we perform a sensitivity analysis *w.r.t.* the number of observations with missing features (in Subsection V-C). All FDRR solutions are solved with GUROBI using the Python API.

A. Evaluation of FDRR Solution Methods

In this subsection, we assess the solution methods presented in Section III, namely `FDRR-V`, `FDRR-R`, and `FDRR-AAR`, by iterating over all eligible combinations of missing features and deleting the respective feature observations from the test set.

Fig. 1 plots the average value (per Γ) and range of the point forecast error metrics, for each solution method, in the four energy forecasting applications. Note that for each value of Γ, we evaluate the methods for the same number of features missing at test time. To avoid cluttering, we only show the odd (and omit the even) values of Γ in the solar production forecasting plot. Unsurprisingly, we observe that the accuracy for each solution method decreases on average as Γ increases, i.e., as more features are missing. Recall that for $\Gamma = 0$, FDRR is a standard LAD, whereas for $\Gamma = |\mathcal{J}|$ all features in $\mathcal{J}$ are ignored, i.e., coefficients are set to zero; hence the three methods are equivalent in these cases (not shown in the plots).

The results in Fig. 1 indicate a similar performance on average for the three methods, with the exception of `FDRR-R` in electricity price forecasting — see top for $\Gamma = 4$ — and solar production forecasting — see bottom. As the number of eligible combinations increases, `FDRR-R` becomes overly conservative, setting all coefficients in $\mathcal{J}$ to zero, which in turn decreases the accuracy. For example, in solar production forecasting, `FDRR(3)-R` becomes equivalent to `FDRR(`$|\mathcal{J}|$`)-R`, which explains the plateau as Γ increases further. Notably, `FDRR-V` and `FDRR-AAR` provide similar performance in terms of average value and range in all applications. Overall, `FDRR-V` ranks higher in solar production forecasting (in about 90% of the combinations) but the differences are very small. `FDRR-AAR` yields slightly better results compared to `FDRR-V`, in electricity price and load forecasting, whereas the results are the same in wind production forecasting.

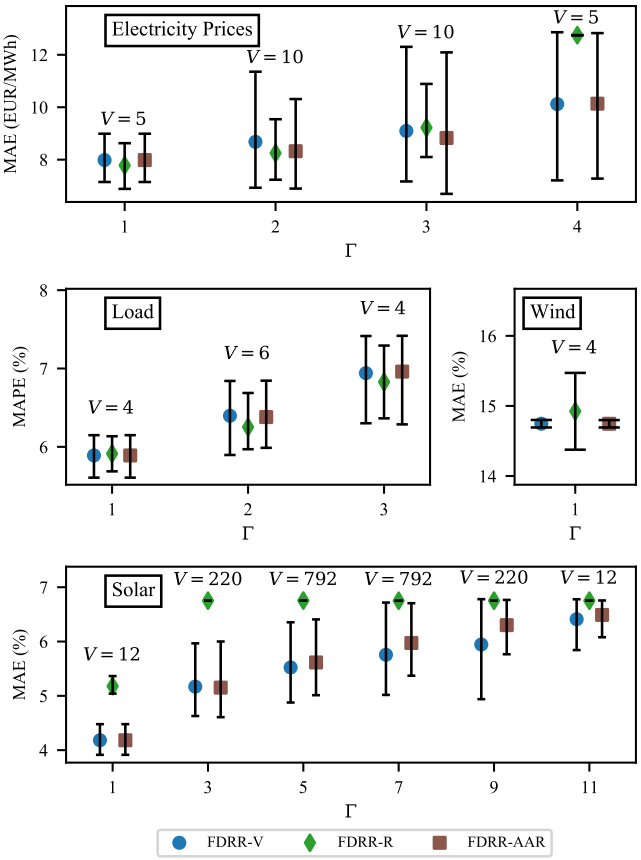

Fig. 1. Average point forecasting error for all combinations of missing features. Bars indicate the range, V indicates the number of vertices per Γ.

We further evaluate the three solution methods in terms of computational cost, by comparing the required CPU time on an Intel Core i7 at 2.7 GHz with 16GB of RAM, using default solver settings. Our results indicate that when the number of vertices V is relatively small, all methods incur a similar cost. However, as V increases, `FDRR-V` incurs a computational cost that is several orders of magnitude larger than the other methods. For example, in solar production forecasting, for $\Gamma = 6$, the CPU time ranges from around 200 to over 27×10^3 seconds for `FDRR(6)-V`, whereas the worst case is less than 1 second and 3.5 seconds, for `FDRR(6)-R` and `FDRR(6)-AAR`, respectively. Clearly, `FDRR-V` incurs a much higher computational cost, which renders this method at least impractical, even for a modest number of features.

We also evaluated the performance on probabilistic forecasts, by repeating the above experiment and training a separate model for each quantile. The obtained results and remarks were very similar to the point forecasts. Pinball loss values increased with Γ, `FDRR-R` yielded high pinball loss values, similarly to the errors in Fig. 1, whereas `FDRR-V` and `FDRR-AAR` yielded quite similar performance.

Henceforth, we shall further consider only `FDRR-AAR`, which stands out as the best FDRR representative with good out-of-sample performance and low computational cost.

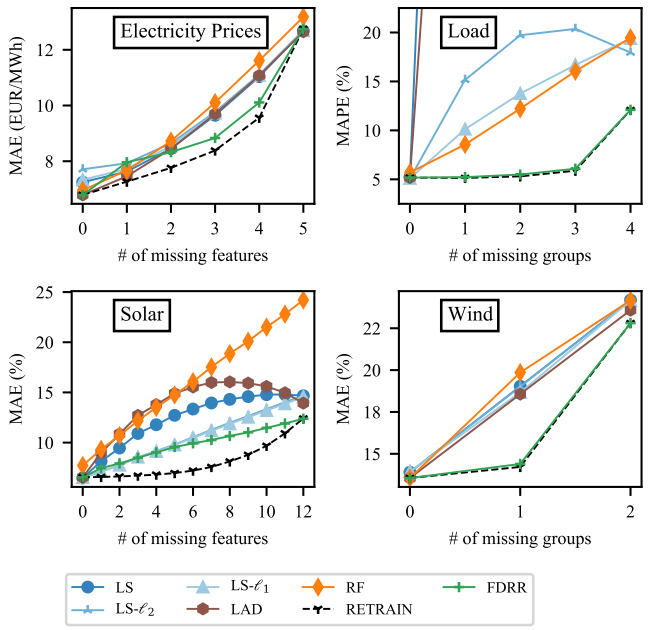

Fig. 2. Point forecasting error metrics versus number of missing features.

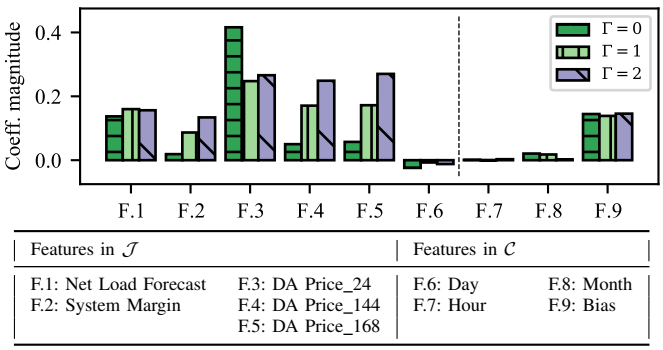

Fig. 3. FDRR(Γ) coefficients for point forecasting of electricity prices.

B. Comparison of FDRR with Benchmark Models

In this subsection, we compare FDRR with the benchmark models presented in Subsection IV-A. For all applications, we iterate over each day of the test set, sample a subset of features, and delete it, repeating the process 10 times.

Fig. 2 presents the average error metrics for point forecasting as a function of the number of missing features. In the nominal case, i.e., without missing features, performance is on par with previous works. Specifically, for each application, the best performing model is: LAD, for electricity price forecasting, with MAE 6.79 EUR/MWh; LS-ℓ_2, for load forecasting, with MAPE 5.07%; LAD, for wind production forecasting, with MAE 13.55%, and LS, LS-ℓ_2, for solar production forecasting, with MAE 6.47%.

Overall, RETRAIN yields the best results in terms of accuracy when features are missing, followed by FDRR, which is clearly a second best. The relative average (maximum) error increase of FDRR compared to RETRAIN is 4.7% (10%) for electricity price, 1.6% (4%) for load, 0.4% (1.7%) for wind production, and 21% (38%) for solar production forecasting. The underlying trend suggests that the gap between FDRR and RETRAIN increases as the number of eligible combinations increases, with its worst case observed for solar production forecasting with 6 missing features, i.e., $\binom{12}{6} = 924$ combinations. FDRR outperforms imputation-based benchmarks, namely LS-type, LAD, and RF, in almost all cases, with an average error reduction of 2% for electricity price, 37% for load, 9% for wind production, and 5% for solar production forecasting. A few exceptions appear, although the differences are small — see top left plot for $\Gamma = 1$ (7% worse than LAD) and bottom left plot for $\Gamma = \{2, 3\}$ (5% worse than LS-ℓ_2).

Taking a closer look at the imputation-based benchmarks, we observe that LS and LAD exhibit a similar performance, in all applications. Note that for load forecasting (top right plot), although both LS and LAD perform on par with [17] in the nominal case, they suffer from bad conditioning, which leads to very large coefficients, and, in turn, to bad performance when features are missing (not shown in plot). The regularized models LS-ℓ_1/ℓ_2, in general, improve the performance of the LS model — see ,e.g., LS-ℓ_1/ℓ_2 for load (top right) and solar production (bottom left) forecasting. Lastly, RF exhibits the worst performance on average amongst the benchmarks, with the exception of the load forecasting case.

To gain further insight, we focus on point forecasting of electricity prices and examine the effect of Γ. Fig. 3 presents the learned coefficients for $\Gamma = \{0, 1, 2\}$. Considering FDRR(0), i.e., LAD, the plot suggests that the price at lag 24 (DA Price_24 or F.3), i.e., same hour of the previous day, is the most important feature, followed by the Net Load Forecast (F.1); therefore, if any of them is missing, the impact on performance is expected to be significant. On the other hand, the coefficients for prices at lag 144 (F.4), and lag 168 (F.5) are small, therefore their deletion has a smaller impact. Intuitively, F.3, F.4, and F.5 carry similar information pertaining to the autoregressive and seasonal nature of electricity prices. For $\Gamma = 0$, these three coefficients vary significantly, with a standard deviation of approximately 17%. For $\Gamma = 1$ we observe that the values of the coefficients come closer, and their standard deviation decreases to 3.5%, while for $\Gamma = 2$ their standard deviation further decreases to 0.09%. Effectively, FDRR(Γ) hedges against feature uncertainty by assigning similar coefficients to these features, which, in turn, mitigates the adverse effect of deleting F.3 from the test set. Moreover, we observe that the total weight of the coefficients increases with Γ to compensate for the larger number of features set to zero during training. Similar results are also observed for the other applications, but omitted due to space limitations. For solar production, e.g., FDRR(1) hedges against the deletion of the surface solar radiation down (SSRD) forecast, which is arguably the most important feature.

We further examine performance for probabilistic forecasting, and illustrate in Fig. 4 the average pinball loss for all applications. Note that we do not examine RETRAIN in this case, as applying it for each quantile becomes prohibitive. Indeed, the results closely resemble the ones presented in Fig. 2. The ranking of the models is generally maintained, with FDRR outperforming the benchmarks in all cases except for

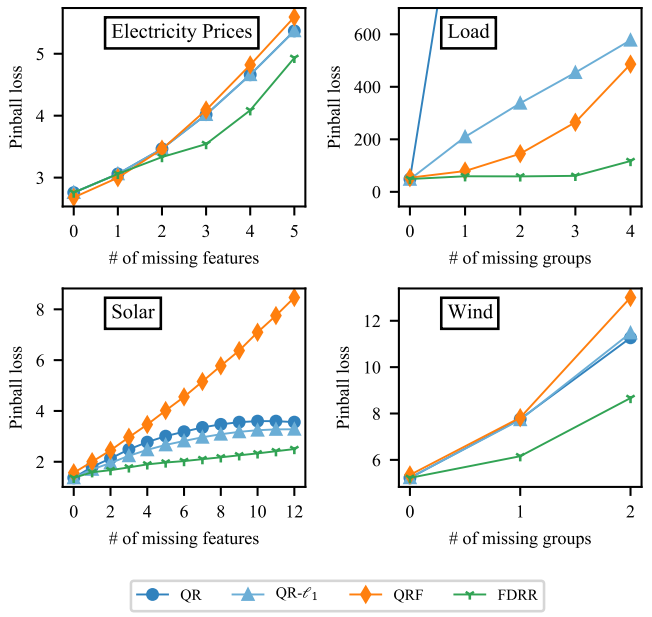

Fig. 4. Pinball loss versus number of missing features.

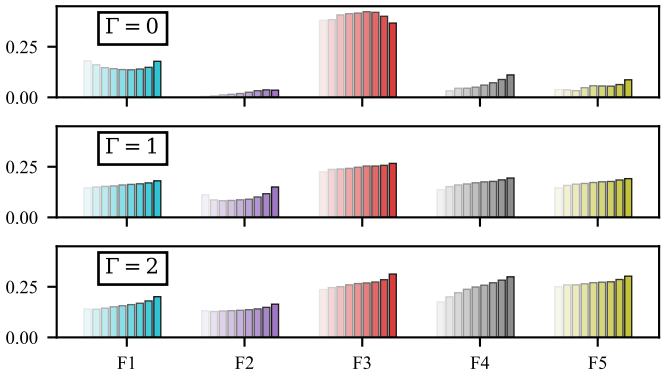

Fig. 5. FDRR(Γ) coefficients for probabilistic forecasting of electricity prices. Higher transparency indicates lower quantiles (a 10% step is considered).

electricity price forecasting for $\Gamma = 1$ (7% worse than QRF), with an average pinball loss reduction of 5% for electricity price, 46% for load, 15% for wind production, and 21% for solar production forecasting. Moreover, as the number of missing features increases, the pinball loss increases in a qualitatively similar fashion as the respective error metrics for point forecasts. Lastly, as Γ increases, the values of the coefficients for all quantiles come closer — see, e.g., Fig. 5 for an illustration of probabilistic forecasting of electricity prices, for $\Gamma = \{0, 1, 2\}$.

C. Sensitivity Analysis

In this subsection, we perform sensitivity analysis *w.r.t.* the number of observations with missing features. Specifically, we sample a percentage of test observations that have missing features, we draw the number of missing features for each observation from a uniform distribution, and we subsequently sample the feature subset that is missing.

Table II presents the average point forecasting errors over 10 runs. The parentheses indicate the difference from the lowest nominal error, which is used to measure performance degradation. The best model is underlined in bold and the second best is in bold. As expected, RETRAIN leads to the smallest error when features are missing and is also the most consistent, i.e., it has the smallest degradation. FDRR typically ranks second both in terms of expected error and performance degradation, with generally small differences from RETRAIN (with the exception of solar production forecasting, where the performance degradation of FDRR is about twice higher compared to RETRAIN). Compared to imputation-based benchmarks, FDRR leads to both smaller error and smaller degradation in all cases except for the lower percentages in solar production forecasting, where it is worse than LS-ℓ_2 but only for up to 0.04%. Further, the relative improvement of FDRR over the benchmarks increases with the percentage of observations with missing features. Considering only imputation-based benchmarks, all models exhibit similar performance for electricity price and wind production forecasting, whereas LS-ℓ_1 and LS-ℓ_2 are significantly better than the rest for load and solar production forecasting.

We further investigate how FDRR performs with an approximation of the quadratic loss function for solar production forecasting, which is the only application where LS ranks first without missing features. We use a piecewise linear function with 20 equally spaced breakpoints within $[-1, 1]$ — recall that the production is normalized between $[0, 1]$ — to approximate the quadratic loss and solve the robust problem using the affinely adjustable reformulation. Results are shown in the last row of Table II (FDRR−PWL). Without missing features, FDRR−PWL and LS have the same error, indicating that the piecewise linearization approximates the quadratic loss well. However, when features are missing, FDRR−PWL significantly outperforms LS (similarly to the way FDRR outperforms LAD). Furthermore, we note that for the lowest percentage (5%) of observations with missing features, where LS performs better than LAD, FDRR−PWL slightly outperforms FDRR.

VI. Conclusions

This work provided a principled approach to enhance resilience against missing features in energy forecasting applications via robust optimization. We formulated a robust regression model that is optimally resilient against missing features at test time, considering both point and probabilistic forecasting, and we developed three solution methods for the resulting robust formulation, leading to LP problems. The numerical results indicated that the affinely adjustable reformulation method provides the best trade-off between accuracy and computational cost. In a comprehensive evaluation against several benchmarks coupled with imputation, the proposed approach improved point (probabilistic) forecasting performance in the presence of missing features by 2% (5%) for electricity price, 37% (46%) for load, 9% (15%) for wind production, and 5% (21%) for solar production. Moreover, the proposed approach performed comparable to retraining without the missing features, while avoiding a large number of additional

TABLE II
POINT FORECASTING ERROR VERSUS PERCENTAGE (%) OF OBSERVATIONS WITH MISSING FEATURES.

% of obs.		0 %	5 %	10 %	25%	50 %
El. Prices	LS	7.25 (0.46)	7.39 (0.60)	7.52 (0.73)	7.91 (1.12)	8.57 (1.78)
	LS-ℓ_2	7.71 (0.92)	7.83 (1.04)	7.95 (1.16)	8.29 (1.50)	8.87 (2.08)
	LS-ℓ_1	7.33 (0.54)	7.47 (0.68)	7.60 (0.81)	7.99 (1.19)	8.65 (1.86)
	LAD	**6.79 (0.00)**	6.95 (0.16)	7.10 (0.31)	7.56 (0.77)	8.33 (1.54)
	RF	**6.90 (0.10)**	7.07 (0.28)	7.23 (0.44)	7.73 (0.94)	8.58 (1.79)
	RETRAIN	**6.79 (0.00)**	**6.92 (0.12)**	**7.03 (0.24)**	**7.38 (0.59)**	**7.97 (1.18)**
	FDRR	**6.79 (0.00)**	**6.94 (0.15)**	**7.08 (0.28)**	**7.48 (0.69)**	**8.20 (1.41)**
Load	LS	5.22 (0.14)	13.65 (8.58)	22.35 (17.28)	46.87 (41.79)	89.07 (84.0)
	LS-ℓ_2	**5.07 (0.00)**	5.74 (0.67)	6.38 (1.31)	8.39 (3.32)	11.69 (6.62)
	LS-ℓ_1	**5.09 (0.02)**	5.60 (0.53)	6.10 (1.03)	7.58 (2.51)	10.03 (4.96)
	LAD	5.18 (0.10)	10.60 (5.53)	15.90 (10.83)	31.58 (26.51)	56.79 (51.72)
	RF	5.72 (0.65)	6.13 (1.06)	6.55 (1.48)	7.81 (2.74)	9.88 (4.81)
	RETRAIN	5.18 (0.10)	**5.27 (0.20)**	**5.38 (0.31)**	**5.66 (0.58)**	**6.13 (1.06)**
	FDRR	5.18 (0.10)	**5.28 (0.21)**	**5.39 (0.31)**	**5.69 (0.62)**	**6.18 (1.11)**
Wind	LS	13.90 (0.36)	14.29 (0.75)	14.65 (1.11)	15.85 (2.31)	17.78 (4.24)
	LS-ℓ_2	13.90 (0.36)	14.29 (0.75)	14.65 (1.11)	15.85 (2.31)	17.78 (4.24)
	LS-ℓ_1	13.95 (0.41)	14.32 (0.79)	14.67 (1.14)	15.83 (2.29)	17.71 (4.18)
	LAD	**13.55 (0.00)**	13.92 (0.39)	14.29 (0.75)	15.46 (1.92)	17.36 (3.82)
	RF	**13.56 (0.01)**	13.95 (0.41)	14.34 (0.80)	15.64 (2.11)	17.66 (4.12)
	RETRAIN	**13.55 (0.00)**	**13.84 (0.30)**	**14.06 (0.52)**	**14.78 (1.24)**	**16.09 (2.55)**
	FDRR	**13.55 (0.00)**	**13.85 (0.31)**	**14.07 (0.53)**	**14.80 (1.26)**	**16.15 (2.61)**
Solar	LS	**6.47 (0.00)**	6.79 (0.32)	7.10 (0.63)	8.04 (1.57)	9.65 (3.18)
	LS-ℓ_2	**6.47 (0.00)**	6.71 (0.23)	**6.92 (0.45)**	7.58 (1.11)	8.73 (2.26)
	LS-ℓ_1	**6.51 (0.04)**	6.74 (0.27)	6.95 (0.48)	7.58 (1.11)	8.70 (2.23)
	LAD	6.54 (0.07)	6.91 (0.44)	7.29 (0.82)	8.42 (1.95)	10.35 (3.88)
	RF	7.71 (1.24)	8.20 (1.72)	8.62 (2.15)	10.03 (3.56)	12.38 (5.91)
	RETRAIN	6.54 (0.07)	**6.62 (0.15)**	**6.71 (0.24)**	**6.94 (0.47)**	**7.37 (0.90)**
	FDRR	6.54 (0.07)	6.74 (0.27)	6.95 (0.48)	**7.53 (1.06)**	**8.51 (2.04)**
	FDRR-PWL	**6.47 (0.00)**	**6.69 (0.22)**	6.94 (0.47)	7.64 (1.17)	8.83 (2.36)

models, and provided resilience in the adverse scenario where the most important feature is missing in an operational setting. A sensitivity analysis *w.r.t.* the number of observations with missing features further validated the practical applicability of the proposed approach. Overall, our results highlight the importance of moving beyond standard accuracy metrics to also consider resilience in adverse scenarios, prior to model deployment.

This work can be extended in several directions. In terms of applications, future work can focus on enhancing resilience within an intra-day forecasting horizon. Jointly considering resilience against missing features and noisy data due to cyber-attacks also presents an interesting methodological challenge.

REFERENCES

[1] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecasting*, vol. 32, no. 3, pp. 914–938, 2016.

[2] J. Nowotarski and R. Weron, "Recent advances in electricity price forecasting: A review of probabilistic forecasting," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 1548–1568, 2018.

[3] Y. Zhang, J. Wang, and X. Wang, "Review on probabilistic forecasting of wind power generation," *Renew. Sustain. Energy Rev.*, vol. 32, pp. 255–270, 2014.

[4] D. W. Van der Meer, J. Widén, and J. Munkhammar, "Review on probabilistic forecasting of photovoltaic power production and electricity consumption," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 1484–1512, 2018.

[5] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," *IEEE Open Access J. Power Energy*, vol. 7, pp. 376–388, 2020.

[6] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data management challenges in production machine learning," *Proc. ACM SIGMOD Int. Conf. Management Data*, vol. Part F1277, pp. 1723–1726, 2017.

[7] M. Bohlke-Schneider, S. Kapoor, and T. Januschowski, "Resilient neural forecasting systems," in *Proc. 4th Int. Workshop Data Management for End-to-End Mach. Learn.*, 2020, pp. 1–5.

[8] European Commission, "A review of the ENTSO-E transparency platform," 2017. [Online]. Available: https://energy.ec.europa.eu/system/files/2018-05/review_of_the_entso_e_plattform_0.pdf

[9] European Centre for Medium-Range Weather Forecasts, "2016 Survey: MARS," 2016. [Online]. Available: https://confluence.ecmwf.int/display/UDOC/2016+Survey%3A+MARS

[10] I. R. White, P. Royston, and A. M. Wood, "Multiple imputation using chained equations: Issues and guidance for practice," *Stat. Med.*, vol. 30, no. 4, pp. 377–399, 2011.

[11] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, 2018.

[12] A. Stratigakos, D. van der Meer, S. Camal, and G. Kariniotakis, "End-to-end learning for hierarchical forecasting of renewable energy production with missing values," in *2022 17th Int. Conf. Probabilistic Methods Applied to Power Syst. (PMAPS)*, 2022, pp. 1–6.

[13] R. Tawn, J. Browell, and I. Dinwoodie, "Missing data in wind farm time series: Properties and effect on forecasts," *Electr. Power Syst. Res.*, vol. 189, 2020.

[14] A. Gerossier, R. Girard, A. Bocquet, and G. Kariniotakis, "Robust day-ahead forecasting of household electricity demand and operational challenges," *Energies*, vol. 11, no. 12, 2018.

[15] M. A. Munoz, J. M. Morales, and S. Pineda, "Feature-driven improvement of renewable energy forecasting and trading," *IEEE Trans. Power Syst.*, vol. 35, no. 5, pp. 3753–3763, 2020.

[16] Q. Li, Y. Xu, B. Chew, H. Ding, and L. Zhao, "An integrated missing-data tolerant model for probabilistic pv power generation forecasting," *IEEE Trans. Power Syst.*, vol. 8950, no. c, pp. 1–1, 2022.

[17] J. Luo, T. Hong, and S.-C. Fang, "Benchmarking robustness of load forecasting models under data integrity attacks," *Int. J. Forecasting*, vol. 34, no. 1, pp. 89–104, 2018.

[18] ——, "Robust regression models for load forecasting," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5397–5404, 2019.

[19] Y. Zhang, F. Lin, and K. Wang, "Robustness of short-term wind power forecasting against false data injection attacks," *Energies*, vol. 13, no. 15, 2020.

[20] Y. Liang, D. He, and D. Chen, "Poisoning attack on load forecasting," in *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, pp. 1230–1235.

[21] J. Luo, T. Hong, and M. Yue, "Real-time anomaly detection for very short-term load forecasting," *J. Mod. Power Syst. Clean Energy*, vol. 6, no. 2, pp. 235–243, 2018.

[22] M. Yue, T. Hong, and J. Wang, "Descriptive analytics-based anomaly detection for cybersecure load forecasting," *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 5964–5974, 2019.

[23] M. Cui, J. Wang, and M. Yue, "Machine learning-based anomaly detection for load forecasting under cyberattacks," *IEEE Trans. Smart Grid*, vol. 10, no. 5, pp. 5724–5734, 2019.

[24] Y. Chen, Y. Tan, and B. Zhang, "Exploiting vulnerabilities of load forecasting through adversarial attacks," in *e-Energy 2019 - Proc. 10th ACM Int. Conf. Future Energy Syst.*, 2019, pp. 1–11.

[25] Y. Zhou, Z. Ding, Q. Wen, and Y. Wang, "Robust load forecasting towards adversarial attacks via bayesian learning," *IEEE Trans. Power Syst.*, pp. 1–1, 2022.

[26] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Stat. Soc. Ser. B Methodol.*, vol. 58, no. 1, pp. 267–288, 1996.

[27] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[28] H. Xu, C. Caramanis, and S. Mannor, "Robust regression and lasso," *IEEE Trans. Inf. Theory*, vol. 56, no. 7, pp. 3561–3574, 2010.

[29] L. El Ghaoui and H. Lebret, "Robust solutions to least-squares problems with uncertain data," *SIAM J. Matrix Anal. Appl.*, vol. 18, no. 4, pp. 1035–1064, 1997.

[30] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Rev.*, vol. 53, no. 3, pp. 464–501, 2011.

[31] C. Caramanis, S. Mannor, and H. Xu, "Robust optimization in machine learning," in *Optimization for machine learning*, S. Sra, S. Nowozin, and S. J. Wright, Eds. MIT Press, 2012, pp. 369–402.

[32] D. Bertsimas, J. Dunn, C. Pawlowski, and Y. D. Zhuo, "Robust classification," *INFORMS J. on Optimization*, vol. 1, no. 1, pp. 2–34, 2019.

[33] D. Bertsimas, X. Boix, K. V. Carballo, and D. den Hertog, "A robust optimization approach to deep learning," *arXiv:2112.09279*, 2021.

[34] A. Globerson and S. Roweis, "Nightmare at test time: Robust learning by feature deletion," *ACM Int. Conf. Proc. Series*, vol. 148, pp. 353–360, 2006.

[35] B. L. Gorissen and D. den Hertog, "Robust counterparts of inequalities containing sums of maxima of linear functions," *Eur. J. Oper. Res.*, vol. 227, no. 1, pp. 30–43, 2013.

[36] A. Stratigakos, P. Andrianesis, A. Michiorri, and G. Kariniotakis, "Making Energy Forecasting Resilient to Missing Features: a Robust Optimization Approach," in *42nd Int. Symp. on Forecasting*, 2022.

[37] R. Koenker and K. F. Hallock, "Quantile regression," *J. Econ. Perspect.*, vol. 15, no. 4, pp. 143–156, 2001.

[38] T. Hong, "Short term electric load forecasting," Ph.D. dissertation, North Carolina State University, 2010.
[39] D. Bertsimas and J. N. Tsitsiklis, *Introduction to linear optimization*. Athena Scientific Belmont, MA, 1997, vol. 6.
[40] D. Bertsimas and D. den Hertog, *Robust and adaptive optimization*. Dynamic Ideas LLC, 2020, vol. 958.
[41] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd Int. Conf. on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.
[42] T. Januschowski, Y. Wang, K. Torkkola, T. Erkkilä, H. Hasson, and J. Gasthaus, "Forecasting with trees," *Int. J. Forecasting*, vol. 38, no. 4, pp. 1473–1481, 2022.
[43] N. Meinshausen, "Quantile regression forests," *J. Mach. Learn. Res.*, vol. 7, no. 35, pp. 983–999, 2006.
[44] D. Bertsimas, A. Delarue, and J. Pauphilet, "Beyond impute-then-regress: Adapting prediction to missing data," *arXiv:2104.03158*, 2021.
[45] ENTSO-E. Transparency platform. [Online]. Available: https://transparency.entsoe.eu/
[46] T. Hong, P. Pinson, and S. Fan, "Global energy forecasting competition 2012," *Int. J. Forecasting*, vol. 30, no. 2, pp. 357–363, 2014.
[47] T. Hong, P. Pinson, S. Fan, H. Zareipour, A. Troccoli, and R. J. Hyndman, "Probabilistic energy forecasting: Global energy forecasting competition 2014 and beyond," *Int. J. Forecasting*, vol. 32, no. 3, pp. 896–913, 2016.