Electric Power Systems Research 212 (2022) 108442

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

Trading data for wind power forecasting: A regression market with lasso regularization☆

Liyang Han [a,*], Pierre Pinson [b], Jalal Kazempour [a]

[a] Department of Wind and Energy Systems, Technical University of Denmark, Kgs. Lyngby, Denmark
[b] Department of Technology, Management and Economics, Technical University of Denmark, Kgs. Lyngby, Denmark

ARTICLE INFO

Keywords:
Data market
Lasso
Linear regression
Time series
Wind power forecasting

ABSTRACT

This paper proposes a regression market for wind agents to monetize data traded among themselves for wind power forecasting. Existing literature on data markets often treats data disclosure as a binary choice or modulates the data quality based on the mismatch between the offer and bid prices. As a result, the market disadvantages either the data sellers due to the overestimation of their willingness to disclose data, or the data buyers due to the lack of useful data being provided. Our proposed regression market determines the data payment based on the least absolute shrinkage and selection operator (lasso), which not only provides the data buyer with a means for selecting useful features, but also enables each data seller to individualize the threshold for data payment. Using both synthetic data and real-world wind data, the case studies demonstrate a reduction in the overall losses for wind agents who buy data, as well as additional financial benefits to those who sell data.

1. Introduction

As the complexity and uncertainty of modern energy systems grow, the value of data for improving system and market operations has become a focus of academic research in recent years [1]. One important usage of data is to inform the forecast of loads and generation [2]. Focusing on wind power forecasting, this paper investigates how the value of external data can be quantified and monetized.

The added value of data for wind power forecasting has been well examined under the assumption that data from external sources is a free and highly accessible commodity [3–5]. However, this assumption becomes unrealistic in applications where data privacy is highly valued [6] or where a conflict of interest exists between the data owners and the data users [7].

To incentivize data exchange, recent literature has been exploring the idea of a data market, in which the data owners (sellers) receive financial compensation from the data users (buyers) for their disclosure of data [8]. In general, these market frameworks can be put into two categories: buyer-centric markets and seller-centric markets. In a buyer-centric market, the data buyer has full control over the data price, while each data seller is either assumed to always accept the offer [9], or given a binary choice on the offer while suffering a privacy loss [10,11]. By contrast, a seller-centric market gives the data sellers the authority to directly add noise to the data [12], or has a third party set the price based on the added value of previously traded data [13]. In the latter, noise is also added to the data if the offering price from the data buyer is lower than the set price. As these market frameworks are often based on game theory, it is often computationally intensive to find the market equilibrium in a noncooperative game setting [10–12] or to derive the market payoffs in a cooperative game setting [13].

In the context of wind power forecasting, it has been shown in the literature that data from surrounding sites, such as other wind farms, can help improve forecast accuracy substantially due to the significant spatio-temporal correlations [4,14]. A wind agent's losses from forecast errors are commonly measured by the mean squared error (MSE) of the forecast compared to the target values of power generation [15,16]. Using the reduction of MSE as the measurement for a wind agent's profit as a data buyer, [7,9] adopt the ordinary least squares (OLS) regression to estimate the forecasting parameters, and construct data markets to incentivize wind agents as market competitors to trade data among themselves. The framework in [7] adopts the pricing scheme from [13]. where the price of data in each trade is determined by the added value of the sellers' data to the previous buyers. As a result, the market outcomes are dependent on the ordering of the buyers, leading to potential large suboptimality gaps and unfairness. Instead of

☆ The research leading to this work is being carried out as a part of the Smart4RES project (European Union's Horizon 2020, No. 864337). The sole responsibility of this publication lies with the author. The European Union is not responsible for any use that may be made of the information contained therein.

* Corresponding author.
E-mail addresses: liyangh@alumni.stanford.edu (L. Han), ppin@dtu.dk (P. Pinson), seykaz@dtu.dk (J. Kazempour).

https://doi.org/10.1016/j.epsr.2022.108442
Received 3 October 2021; Received in revised form 11 April 2022; Accepted 2 July 2022
Available online 15 July 2022

pricing the data prior to each trade, the framework in [9] quantifies the reduction of the losses for each data buyer and allocates the resulting savings to the sellers based on their contribution to the buyer's task. The caveat of this market is the assumption that the sellers have no revenue requirements, when in reality this might not hold true.

OLS regression is known to be highly sensitive to outliers and prone to overfitting when the supporting features, sets of data that are fitted to the target variable in the regression, are highly correlated [17]. Since wind power data for closely located wind farms are likely highly correlated, feature selection methods such as the least absolute shrinkage and selection operator (lasso) [18] have been widely adopted for parameter estimation in wind power forecasting [16,19,20]. It helps eliminate the impact of less correlated data in the forecasting task through adding to the loss function the product of a positive regularization parameter and the absolute value of the coefficient for each feature, referred to as the lasso term.

We observe that not only is the lasso term effective in preventing overfitting, but it can also provide a tool for thresholding the selection of features through customizing their corresponding regularization parameters: the higher the regularization parameter, the more correlated a feature has to be to the target to be selected. This observation serves as the inspiration for our proposed market framework.

The main contribution of this paper is the design of a *lasso regression market* for wind power forecasting, in which the lasso term is exploited as a measure for direct payments from data buyers to data sellers. This market framework has three main advantages: (1) compared to the seller-centric model, the lasso provides the buyers with better quality data through selecting from the sellers' data the features that are most likely to improve the forecast without additional noise; (2) compared to the buyer-centric model, sellers are given the authority to threshold their financial reward through individualizing the lasso regularization parameter; and (3) compared to data markets that are set up based on game theory, the computation of the market outcomes is simpler as the lasso term as an output from the regression directly represents the payment.

The paper is structured as follows: Section 2 introduces the market participants. Section 3 describes the OLS regression and the lasso regression for the data buyer's analytics task. Section 4 formulates the lasso regression market for wind agents and proves its financial viability for both data buyers and data sellers. Section 5 uses synthetic data and measured data from Nord Pool[1] to demonstrate the effectiveness of the proposed market. Section 6 concludes the paper with discussions on the key findings and future work.

2. Market participants

The use case for the proposed regression market is data trading among wind agents. The data buyer is called a *central agent* who has an analytics task to estimate parameters for forecasting wind power. The other wind agents are data sellers, referred to as *support agents*, hold data that could potentially improve the central agent's forecast, and expect to be remunerated for sharing those data. We focus on a single-buyer setup, but it can be easily extended to multiple buyers since the agents' analytics tasks are independent of each other in the sense that the outcome of a central agent's forecasting task does not affect any concurrent task of another agent.

[1] https://www.nordpoolgroup.com/Market-data1/Power-system-data/Production1/Wind-Power/ALL/Hourly1/.

2.1. Agents and the analytics task

We denote the full set of wind agents by $\mathcal{N} = \{1, 2, \dots, N\}$, indexed by i, so $i \in \mathcal{N}$. The data buyer, or the central agent $c \in \mathcal{N}$, has an analytics task to estimate parameters for forecasting wind power as a time series target variable, denoted by vector $\mathbf{y}_c = \begin{bmatrix} y_{c,1} & y_{c,2} & \cdots & y_{c,T} \end{bmatrix}^\top$, where T is the total number of time steps of concern. The support agents are gathered in set $\mathcal{N}_{-c} = \mathcal{N} \setminus \{c\} = \{i \in \mathcal{N} | i \neq c\}$, and the data owned by each agent i are in set $\mathcal{X}_i = \{\mathbf{x}_i^d | d \in [1, D_{i\to c}]\}$, where $D_{i\to c}$ is the total number of relevant features from agent i for the central agent's analytics task, and vector $\mathbf{x}_i^d$ represents agent i's dth feature (e.g., l-hour-ahead wind power data: $\mathbf{x}_i^d = [x_{i,1-l}^d \; x_{i,2-l}^d \; \cdots \; x_{i,T-l}^d]^\top$). The analytics task with all agents' data is given by

$$\hat{\mathcal{B}}_{\mathcal{N}} = F\left(\mathcal{X}_{\mathcal{N}}, \mathbf{y}_c\right), \tag{1}$$

where $\mathcal{X}_{\mathcal{N}} = \cup_{i\in\mathcal{N}} \mathcal{X}_i$ gathers the relevant input features of the central agent and support agents, and $\hat{\mathcal{B}}_{\mathcal{N}}$ is the set of estimated parameter values for the analytics task. If the analytics task follows a linear regression framework, $\hat{\mathcal{B}}_{\mathcal{N}} = \{\beta_i^d | i \in \mathcal{N}, d = 1, \dots, D_{i\to c}\} \cup \{\beta_{\mathcal{N}}^0\}$ is the set of linear coefficients, with each element corresponding to a feature of an agent and $\beta_{\mathcal{N}}^0$ representing the intercept term.

Similarly, if the central agent only relies on their own data for the analytics task, it solves

$$\hat{\mathcal{B}}_c = F\left(\mathcal{X}_c, \mathbf{y}_c\right), \tag{2}$$

where $\hat{\mathcal{B}}_c = \{\beta_c^d | d = 0, 1, \dots, D_{c\to c}\}$ gathers the linear coefficients corresponding to the features owned by the central agent, with β_c^0 being the intercept. Using results from (1) and (2), the central agent can forecast the target variable as $\hat{\mathbf{y}}_c\left(\mathcal{X}_{\mathcal{N}}, \hat{\mathcal{B}}_{\mathcal{N}}\right)$ and $\hat{\mathbf{y}}_c\left(\mathcal{X}_c, \hat{\mathcal{B}}_c\right)$, respectively. Details on linear regression for the analytics task and forecasting are explained in Section 3.

2.2. Support agents and reservation to sell data

Note that the support agents and the central agent can be competitors in the same energy market, which may discourage the support agents from disclosing their data. We measure this barrier in the form of a payment threshold denoted by

$$H_i^d\left(u_i^d, \beta_i^d\right) = \left|u_i^d \beta_i^d\right|, \tag{3}$$

representing the payment required by support agent $i \in \mathcal{N}_{-c}$ for disclosing data associated with their dth feature. It is a function of u_i^d and β_i^d, measuring, respectively, the reservation of agent i to sell their dth feature, and how correlated this feature is to the central agent's target variable. Therefore, the higher u_i^d and β_i^d are, the higher the payment needs to be for agent i to disclose their dth feature to the central agent. In other words, the payment threshold increases as the support agent becomes less willing to sell their data, and as the correlation between the support agent's data and the central agent's target variable becomes stronger.

Some factors that a support agent may take into account when determining their revenue threshold includes the valuation of their loss of privacy, the valuation of their losses due to an increase in their competitor's profit, the cost of collecting data and offering into the regression market, etc. In our proposed framework, the revenue threshold is non-negative. In practice, however, there could be scenarios where the support agent is able to benefit directly from the improvement of the central agent's analytics task (e.g., an overall improvement of the forecast of renewable generation in the energy market can lead to a decrease in the imbalance price that is eventually passed down to all the renewable agents), and is thus willing to receive a negative payment in the regression market. We choose to leave the latter scenario for future work.

2.3. Central agent and willingness to buy data

We assume that the central agent uses the MSE to measure the average losses from the forecast for each time step:

$$S_c^{\mathrm{MSE}}\left(\hat{\mathbf{y}}_c\right)=\frac{1}{T}\sum_{t=1}^{T}\left(y_{c,t}-\hat{y}_{c,t}\right)^2, \tag{4}$$

where $\hat{\mathbf{y}}_c=\left[\hat{y}_{c,1}\ \hat{y}_{c,2}\ \dots\ \hat{y}_{c,T}\right]^\top$. In practice, the central agent may assign a scaling factor to the MSE to represent their actual losses in monetary terms, but we assume this scaling factor to be "1" in this paper for simplicity. Therefore, S_c^{MSE} can be considered to directly represent the central agent's mean financial losses at each time step.

Given the payment threshold H_i^d to obtain agent i's dth feature, in order for the central agent to financially benefit from purchasing data in the regression market, it requires

$$S_c^{\mathrm{MSE}}\left(\hat{\mathbf{y}}_c\left(\mathcal{X}_c,\hat{\mathcal{B}}_c\right)\right)-S_c^{\mathrm{MSE}}\left(\hat{\mathbf{y}}_c\left(\mathcal{X}_{\mathcal{N}},\hat{\mathcal{B}}_{\mathcal{N}}\right)\right)\geq\sum_{i\in\mathcal{N}_{-c}}\sum_{d=1}^{D_{i\to c}}H_i^d. \tag{5}$$

3. Analytics task under linear regression

Given (5), we anticipate there to be an opportunity for the central agent to purchase data from the support agents while making sure the total payment is lower than the reduced losses for the forecast. In this section, we define the analytics tasks in different forms of linear regression for both the case with only the central agent's own data and the case with additional data from the support agents.

3.1. Forecast with linear coefficients

In the case where the central agent c only considers their own features, the forecast of their target based on the coefficients from a linear regression can be written as

$$\hat{y}_{c,t}=\beta_c^0+\sum_{d=1}^{D_{c\to c}}\beta_c^d x_{c,t}^d, \tag{6}$$

In contrast, if the features of the support agents were also considered, the forecast would become

$$\begin{aligned}\hat{y}_{c,t}&=\underbrace{\beta_c^0+\sum_{d=1}^{D_{c\to c}}\beta_c^d x_{c,t}^d}_{\text{central agent}}+\underbrace{\sum_{i\in\mathcal{N}_{-c}}\left(\beta_i^0+\sum_{d=1}^{D_{i\to c}}\beta_i^d x_{i,t}^d\right)}_{\text{support agents}}\\&=\beta_{\mathcal{N}}^0+\sum_{i\in\mathcal{N}}\sum_{d=1}^{D_{i\to c}}\beta_i^d x_{i,t}^d,\end{aligned} \tag{7}$$

where $\beta_{\mathcal{N}}^0=\sum_{i\in\mathcal{N}}\beta_{i,0}$ is the sum of the intercept terms of all agents, and thus the overall intercept.

In the rest of the paper we assume that the dependencies of the target variable on the features are stationary, and thus the true linear coefficients do not vary with time. For ease of notation, we define vector $\mathbf{x}_{i,t}=[x_{i,t}^1\ x_{i,t}^2\ \dots\ x_{i,t}^{D_{i\to c}}]^\top,\forall i\in\mathcal{N}$ as the values of all of agent i's features used for the forecast of the central agent's target variable at time step t.

3.2. Baseline losses using OLS regression on own features

Without a data market, the central agent can only rely on their own data for the analytics task and forecasting. We construct the central agent's own regressor vector $\mathbf{x}_{\{c\},t}=[1\ \mathbf{x}_{c,t}^\top]^\top$, and use $\boldsymbol{\beta}_c^*=[\beta_c^{0*}\ \beta_c^{1*}\ \dots\ \beta_c^{D_{c\to c}*}]^\top$ to denote the vector of linear coefficients for the regressors. Using OLS regression, the analytics task in (2) becomes

$$\boldsymbol{\beta}_c^*=\underset{\boldsymbol{\beta}\in\mathbb{R}^{1+D_{c\to c}}}{\operatorname{argmin}}\sum_{t=1}^{T}\left(y_{c,t}-\hat{y}_{c,t}\right)^2 \tag{8}$$

$$\overset{(6)}{=}\underset{\boldsymbol{\beta}\in\mathbb{R}^{1+D_{c\to c}}}{\operatorname{argmin}}\sum_{t=1}^{T}\left(y_{c,t}-\mathbf{x}_{\{c\},t}^\top\boldsymbol{\beta}\right)^2. \tag{9}$$

Comparing (4) and (8), since the constant $\frac{1}{T}$ does not affect the "argmin" function, we can rewrite (8) as

$$\boldsymbol{\beta}_c^*=\underset{\boldsymbol{\beta}\in\mathbb{R}^{1+D_{c\to c}}}{\operatorname{argmin}}\ S_c^{\mathrm{MSE}}\left(\hat{\mathbf{y}}_c\left(\mathcal{X}_c,\boldsymbol{\beta}\right)\right). \tag{10}$$

Since the MSE measures the central agent's financial losses from the forecast (Section 2.3), we can consider $S_c^{\mathrm{MSE}}\left(\hat{\mathbf{y}}_c\left(\mathcal{X}_c,\boldsymbol{\beta}_c^*\right)\right)$ the baseline losses of the central agent without access to data from other agents.

3.3. Losses using lasso regression with support features

When data from the support agents are made available to the central agent, there is a potential for the central agent to improve their forecasting accuracy. An OLS regression can again be applied to fit the available features, but a well known drawback of OLS regression is its sensitivity to outliers and highly correlated features, which tends to result in overfitting the data and compromising the forecast accuracy [17]. Additionally, if the data collected from the support agents are of low quality (e.g., containing missing values, voluntarily flawed, etc.), OLS regression is not capable of eliminating these corrupted features. One way to mitigate such risks is to apply cross-validation of the support features, but it can be computationally expensive and complicates the market set up (i.e., how to reward the support agents for the portion of their data that are only used for cross-validation). Another way is to use feature selection methods, including L_p regularization methods that have been proposed to reduce the impact of less correlated features on the forecast performance. Among these, the lasso is a popular regularization method that yields sparse coefficient matrices, which helps prevent overfitting [18]. The lasso term is an L_1-norm penalty applied to the coefficients of a linear regression problem.

To implement the lasso regression on the features of all agents, we construct the all-agents regressor vector $\mathbf{x}_{\mathcal{N},t}=[1\ \mathbf{x}_{1,t}^\top\ \dots\mathbf{x}_{N,t}^\top]^\top$. We then denote the vector of the corresponding linear coefficients using lasso regression by $\boldsymbol{\beta}_{\mathcal{N}}^{L_1}=[\beta_{\mathcal{N}}^{0\ L_1}\ \boldsymbol{\beta}_{\mathcal{N},1}^{L_1\ \top}\ \dots\ \boldsymbol{\beta}_{\mathcal{N},N}^{L_1\ \top}]^\top$, where $\boldsymbol{\beta}_{\mathcal{N},i}^{L_1}=[\beta_i^{1\ L_1}\ \beta_i^{2\ L_1}\ \dots\ \beta_i^{D_{i\to c}\ L_1}]^\top,\forall i\in\mathcal{N}$. Applying the generic lasso estimator, the analytics task in (1) becomes

$$\boldsymbol{\beta}_{\mathcal{N}}^{L_1}=\underset{\boldsymbol{\beta}\in\mathbb{R}^{1+\sum_{i\in\mathcal{N}}D_{i\to c}}}{\operatorname{argmin}}\left[\frac{1}{2}\sum_{t=1}^{T}\left(y_{c,t}-\mathbf{x}_{\mathcal{N},t}^\top\boldsymbol{\beta}\right)^2+\lambda\left\|\boldsymbol{\beta}\right\|_1\right], \tag{11}$$

where λ is the lasso regularization parameter. The lasso term $\lambda\left\|\boldsymbol{\beta}\right\|_1$ shrinks some coefficients and sets some of them to zero. As a result, it reduces or even eliminates the influence of the features that are less likely to contribute to the improvement of the forecast. The greater λ is, the more shrinkage is applied to the regression coefficients.

We apply a constant factor of $\frac{2}{T}$ within the "argmin" function in (11) and rewrite it based on (4) and (7) as

$$\boldsymbol{\beta}_{\mathcal{N}}^{L_1}=\underset{\boldsymbol{\beta}\in\mathbb{R}^{1+\sum_{i\in\mathcal{N}}D_{i\to c}}}{\operatorname{argmin}}\left[S_c^{\mathrm{MSE}}\left(\hat{\mathbf{y}}_c\left(\mathcal{X}_{\mathcal{N}},\boldsymbol{\beta}\right)\right)+\frac{2\lambda}{T}\left\|\boldsymbol{\beta}\right\|_1\right]. \tag{12}$$

Similarly, $S_c^{\mathrm{MSE}}\left(\hat{\mathbf{y}}_c\left(\mathcal{X}_{\mathcal{N}},\boldsymbol{\beta}_{\mathcal{N}}^{L_1}\right)\right)$ represents the financial losses of the central agent if the forecast is based on a lasso regression given all support agents' features.

4. Regression market with the lasso

In this section, we introduce the concept of the *lag* in time series data as a way to define the number of relevant features from recent time steps from each agent. For wind power forecasting, the lag not only captures the temporal correlations of the wind generation at a specific site, it also indirectly captures the spatial correlations between neighboring sites as a result of the natural development of wind [14].

Using the lasso term to define the payment, we then construct a regression market for wind agents, which is proved to meet the payment requirement of support agents while guaranteeing profit for the central agent. We note that only in-sample market outcomes are analyzed in this paper, meaning T can also be interpreted as the total number of time steps for training the model.

4.1. Linear regression on features with a fixed maximum lag

Recall from Sections 2 and 3 that $D_{i\to c}$ represents the number of relevant features from each agent. Here, since we are dealing with time series data, we equate $D_{i\to c}$ to agent i's maximum lag, which is the number of recent time steps that are considered relevant for the target variable. This means that each feature from a support agent represents their data of a certain lag to the target variable. Assuming features from all agents are available up to one time step before the target variable $y_{c,t}$ is revealed, and that a fixed maximum lag D applies to all agents, i.e., $D_{i\to c} = D, \forall i \in \mathcal{N}$, the relevant features for $y_{c,t}$ from any agent i can be gathered in the vector $\mathbf{x}_{i,t} \in \mathbb{R}^D = [x_{i,t-D}\ x_{i,t-D+1}\ \dots\ x_{i,t-1}]^\top, \forall i \in \mathcal{N}$. Applying the fixed maximum lag D to the regressions in Sections 3.2 and 3.3, we have $|\boldsymbol{\beta}_c^*| = 1 + D$, and $|\boldsymbol{\beta}_{\mathcal{N}}^*| = 1 + ND$.

The features of any agent i can then be expressed in $\mathbf{X}_{i,T} \in \mathbb{R}^{T\times D} := \left[\mathbf{x}_{i,1}\ \mathbf{x}_{i,2}\ \dots\ \mathbf{x}_{i,T}\right]^\top$. To prepare for the regression, we gather the data from the central agent alone in $\mathbf{X}_{\{c\},T} \in \mathbb{R}^{T\times(1+D)} := \left[\mathbf{1}_T\ \mathbf{X}_{c,T}\right]$ and data from all agents in $\mathbf{X}_{\mathcal{N},T} \in \mathbb{R}^{T\times(1+ND)} := \left[\mathbf{1}_T\ \mathbf{X}_{1,T}\ \dots\ \mathbf{X}_{N,T}\right]$, where $\mathbf{1}_T = [\underbrace{1\ \dots\ 1}_{T}]^\top$. Therefore, (9) and (11) can be rewritten as

$$\boldsymbol{\beta}_c^* = \underset{\boldsymbol{\beta}\in\mathbb{R}^{1+D}}{\operatorname{argmin}} \left\|\mathbf{y}_c - \mathbf{X}_{\{c\},T}\boldsymbol{\beta}\right\|_2^2, \tag{13}$$

and

$$\boldsymbol{\beta}_{\mathcal{N}}^{L_1} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{1+ND}}{\operatorname{argmin}} \left\{\frac{1}{2}\left\|\mathbf{y}_c - \mathbf{X}_{\mathcal{N},T}\boldsymbol{\beta}\right\|_2^2 + \lambda\left\|\boldsymbol{\beta}\right\|_1\right\}. \tag{14}$$

For clarity, we expand the following matrix operation from (14) as

$$\mathbf{y}_c - \mathbf{X}_{\mathcal{N},T}\boldsymbol{\beta} = \begin{bmatrix} y_{c,1} \\ y_{c,2} \\ \vdots \\ y_{c,T} \end{bmatrix} - \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_{1,1-D} & x_{1,2-D} & \cdots & x_{1,T-D} \\ x_{1,1-D+1} & x_{1,2-D+1} & \cdots & x_{1,T-D+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1,0} & x_{1,1} & \cdots & x_{1,T-2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1-D} & x_{N,2-D} & \cdots & x_{N,T-D} \\ x_{N,1-D+1} & x_{N,2-D+1} & \cdots & x_{N,T-D+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,0} & x_{N,1} & \cdots & x_{N,T-2} \end{bmatrix}^\top \begin{bmatrix} \beta_{\mathcal{N}}^0 \\ \beta_1^1 \\ \beta_1^2 \\ \vdots \\ \beta_1^D \\ \vdots \\ \beta_N^1 \\ \beta_N^2 \\ \vdots \\ \beta_N^D \end{bmatrix}.$$

4.2. Regression market with the lasso term as payment

Considering all T time steps, we rewrite the average financial losses of the central agent as a function of the regressor matrix $\mathbf{X}$ and a given vector of linear coefficients $\boldsymbol{\beta}$ as

$$S_c^{\mathrm{MSE}}\left(\mathbf{X}\in\mathbb{R}^{T\times|\beta|}, \boldsymbol{\beta}\right) = \frac{1}{T}\left\|\mathbf{y}_c - \mathbf{X}\boldsymbol{\beta}\right\|_2^2, \tag{15}$$

where $|\boldsymbol{\beta}|$ equals the number of features available for the regression including the intercept.

In the generic lasso estimator in (14), a single λ is applied to all the coefficients, whereas in practice we can assign different λ's to different coefficients without compromising the computation efficiency. Therefore we can construct a lasso regularization scalar matrix $\lambda_{\mathcal{N}}^D \in \mathbb{R}_{\geq 0}^{1+ND} = \mathrm{diag}(\lambda_0\ \lambda_1^1\ \dots\ \lambda_1^D\ \lambda_2^1\ \dots\ \lambda_2^D\ \dots\ \lambda_N^1\ \dots\ \lambda_N^D)$, where we set $\lambda_0 = 0$ to ensure no shrinkage is applied to the intercept term. We then define the *lasso loss* function as

$$\begin{aligned} S_c^{L_1}\left(\mathbf{X}_{\mathcal{N},T}, \lambda_{\mathcal{N}}^D, \boldsymbol{\beta}\right) &= \frac{1}{T}\left\|\mathbf{y}_c - \mathbf{X}_{\mathcal{N},T}\boldsymbol{\beta}\right\|_2^2 + \left\|\frac{2}{T}\lambda_{\mathcal{N}}^D\boldsymbol{\beta}\right\|_1 \\ &= S_c^{\mathrm{MSE}}\left(\mathbf{X}_{\mathcal{N},T}, \boldsymbol{\beta}\right) + \left\|\frac{2}{T}\lambda_{\mathcal{N}}^D\boldsymbol{\beta}\right\|_1. \end{aligned} \tag{16}$$

So (14) can be modified as

$$\boldsymbol{\beta}_{\mathcal{N}}^{L_1}\left(\mathbf{X}_{\mathcal{N},T}, \lambda_{\mathcal{N}}^D\right) = \underset{\boldsymbol{\beta}\in\mathbb{R}^{1+ND}}{\operatorname{argmin}} S_c^{L_1}\left(\mathbf{X}_{\mathcal{N},T}, \lambda_{\mathcal{N}}^D, \boldsymbol{\beta}\right). \tag{17}$$

Here, we propose to use (17) as the basis for the regression market for wind agents. Relying on their own data, the central agent has the baseline financial losses of $S_c^{\mathrm{MSE}}\left(\mathbf{X}_{\{c\},T}, \boldsymbol{\beta}_c^*\right)$. When support agent $i \in \mathcal{N}_{-c}$ offers feature $x_{i,t}^d$ into the market, they also have the freedom to set λ_i^d based on u_i^d, their reservation to sell (Section 2.2). After the market operator conducts the market using (17), $\beta_i^{dL_1}$ is computed. If $\beta_i^{dL_1} = 0$, feature $x_{i,t}^d$ is not selected to be used for the central agent's forecast, and no payment is needed. Otherwise the central agent has to pay agent i in the amount of $\left|\frac{2}{T}\lambda_i^d\beta_i^{dL_1}\right|$ for using feature $x_{i,t}^d$ for the forecast. Therefore, the lasso term $\left\|\frac{2}{T}\lambda_{\mathcal{N}}^D\boldsymbol{\beta}\right\|_1$ in (16) represents the central agent's total payment, and $S_c^{L_1}\left(\mathbf{X}_{\mathcal{N},T}, \lambda_{\mathcal{N}}^D, \boldsymbol{\beta}\right)$ represents the central agent's sum of financial losses and payments in the regression market.

Proposition 1. *If no shrinkage is applied to the central agent's own features ($\lambda_c^d = 0, \forall d$), and each support agent sets $\lambda_i^d = \frac{T}{2}u_i^d$, then the central agent's sum of financial losses and data payments in the regression market is no greater than the central agent's financial losses without the regression market.*

Proof. In order to avoid shrinkage of the central agent's own features, we construct scalar matrix $\lambda_{\mathcal{N}}^D = \mathrm{diag}(0\ \lambda_1^1\ \dots\ \lambda_1^D\ \lambda_2^1\ \dots\ \lambda_2^D\ \dots\ \lambda_N^1\ \dots\ \lambda_N^D)$, while

$$\lambda_c^d = 0,\ d = 1, 2, \dots, D. \tag{18}$$

Let us construct a vector of regression parameters $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{1+ND} := [\hat{\beta}^0\ \hat{\boldsymbol{\beta}}_1^\top\ \dots\ \hat{\boldsymbol{\beta}}_N^\top]^\top$, where $\hat{\boldsymbol{\beta}}_i = [\hat{\beta}_i^1\ \hat{\beta}_i^2\ \dots\ \hat{\beta}_i^D]^\top, \forall i \in \mathcal{N}$. We set

$$\hat{\boldsymbol{\beta}}_i = \mathbf{0}^D = [\underbrace{0\ \dots\ 0}_{D}]^\top, \forall i \neq c. \tag{19}$$

We further set

$$[\hat{\beta}^0\ \hat{\boldsymbol{\beta}}_c^\top]^\top = \boldsymbol{\beta}_c^* \overset{(13)}{=} \underset{\boldsymbol{\beta}\in\mathbb{R}^{1+D}}{\operatorname{argmin}} \left\|\mathbf{y}_c - \mathbf{X}_{\{c\},T}\boldsymbol{\beta}\right\|_2^2. \tag{20}$$

Therefore, the central agent's average financial losses per time step using their own features are given by

$$S_c^{\mathrm{MSE}}\left(\mathbf{X}_{\{c\},T}, \boldsymbol{\beta}_c^*\right) = \frac{1}{T}\left\|\mathbf{y}_c - \left[\mathbf{1}_T\ \mathbf{X}_{c,T}\right][\hat{\beta}^0\ \hat{\boldsymbol{\beta}}_c^\top]^\top\right\|_2^2. \tag{21}$$

Applying $\hat{\boldsymbol{\beta}}$ to the lasso loss function of the central agent using all agents' features, we have

$$S_c^{L_1}\left(\mathbf{X}_{\mathcal{N},T}, \lambda_{\mathcal{N}}^D, \hat{\boldsymbol{\beta}}\right) \tag{22}$$

$$\overset{(16)}{=} \frac{1}{T}\left\|\mathbf{y}_c - \mathbf{X}_{\mathcal{N},T}\hat{\boldsymbol{\beta}}\right\|_2^2 + \left\|\frac{2}{T}\lambda_{\mathcal{N}}^D\hat{\boldsymbol{\beta}}\right\|_1 \tag{23}$$

$$\overset{(19)}{=} \frac{1}{T}\left\|\mathbf{y}_c - \left[\mathbf{1}_T\ \mathbf{X}_{c,T}\right][\hat{\beta}^0\ \hat{\boldsymbol{\beta}}_c^\top]^\top\right\|_2^2 + \left\|\frac{2}{T}\mathrm{diag}(0\ \lambda_c^1\ \dots\ \lambda_c^D)[\hat{\beta}^0\ \hat{\boldsymbol{\beta}}_c^\top]^\top\right\|_1 \tag{24}$$

$$\overset{(18)}{=} \frac{1}{T}\left\|\mathbf{y}_c - \left[\mathbf{1}_T\ \mathbf{X}_{c,T}\right][\hat{\beta}^0\ \hat{\boldsymbol{\beta}}_c^\top]^\top\right\|_2^2 + 0 \tag{25}$$

$$\overset{(21)}{=} S_c^{\mathrm{MSE}}\left(\mathbf{X}_{\{c\},T}, \boldsymbol{\beta}_c^*\right) \tag{26}$$

$$\overset{(22)}{\geq} \min_{\boldsymbol{\beta}\in\mathbb{R}^{1+ND}} S_c^{L_1}\left(\mathbf{X}_{\mathcal{N},T}, \lambda_{\mathcal{N}}^D, \boldsymbol{\beta}\right) \tag{27}$$

$$\overset{(17)}{=} S_c^{L_1}\left(\mathbf{X}_{\mathcal{N},T}, \lambda_{\mathcal{N}}^D, \boldsymbol{\beta}_{\mathcal{N}}^{L_1}\right) \tag{28}$$

$$\overset{(16)}{=} S_c^{\mathrm{MSE}}\left(\mathbf{X}_{\mathcal{N},T}, \boldsymbol{\beta}_{\mathcal{N}}^{L_1}\right) + \left\|\frac{2}{T}\lambda_{\mathcal{N}}^D\boldsymbol{\beta}_{\mathcal{N}}^{L_1}\right\|_1 \tag{29}$$

$$= S_c^{\mathrm{MSE}}\left(\mathbf{X}_{\mathcal{N},T}, \boldsymbol{\beta}_{\mathcal{N}}^{L_1}\right) + \sum_{i\in\mathcal{N}_{-c}}\sum_{d=1}^{D}\left|u_i^d\beta_i^{dL_1}\right|, \tag{30}$$

where we obtain (30) from (29) by setting $\lambda_i^d = \frac{T}{2}u_i^d$. Since (30) is the central agent's sum of financial losses and data payments in the regression market, and (26) is the central agent's financial losses without the regression market, we have thus proved Proposition 1. □

Note that with (26) ≥ (30), we also prove that (5) is satisfied, which means the central agent is guaranteed to financially benefit from the regression market.

To further explain this, let us consider a support feature $x_{i,t}^d$ for estimating $y_{c,t}$. In order for the market operator (17) not to set the corresponding coefficient β_i^d to zero, it has to contribute to a reduction in financial losses that is greater than or equal to the payment it incurs. Using the assumption $\lambda_i^d = \frac{T}{2}u_i^d$ in Proposition 1, the payment to agent i for feature $x_{i,t}^d$ is

$$\left|\frac{2}{T}\lambda_i^d \beta_i^{d\,L_1}\right| = \left|u_i^d \beta_i^{d\,L_1}\right|, \tag{31}$$

meeting the payment requirement of each support agent i. In summary, the proposed regression market guarantees financial viability for both the support agents and the central agent.

Observe that a direct relationship is drawn between the linear coefficients of support features and the payment, without the support features being standardized in the lasso regression. This means that a support feature's mean and variance can influence the linear coefficient, thus the payment as well. Therefore, each support agent, given a desirable revenue threshold, needs to set their reservation to sell accordingly to account for the means and variances of their support features. The reservation to sell can be considered an independent choice of each support agent with the aim to achieve a certain revenue threshold rather than a metric to compare laterally support agents' willingness to disclose their data. An alternative way of designing the market is to require each support agent to submit standardized data and the corresponding reservation to sell. We choose the former to provide the possibility for support agents to encrypt both their data and their true reservation to sell before submitting features to the market [21].

The linear relationship between the payment and the absolute value of the coefficient also raises the question of whether other forms of the payment term (e.g., a quadratic term in β) could apply to the regression market. Albeit it not being the focus of the paper, the lasso regularizer can also enable the regression with support agents' data to be conducted in a distributed manner [20], thus providing another level of privacy protection. Therefore, we apply the lasso payment in our proposed framework, with the aim to extend it to distributed learning in the future.

5. Case studies

To verify the performance of the proposed regression market, we design two case studies in this section. In the first case study, we construct synthetic datasets for all the agents with fixed linear correlations. This way we have the ground truth of the correlations between the agents' data to verify the model results. In the second case study, we use the hourly wind generation zonal data from Nord Pool to demonstrate the impact of the support agents' reservation to sell on the profit of the market participants.

5.1. Regression market implemented on synthetic data

In this case study, we use an autoregressive process to simulate the data of 4 independent market players (P2–P5) with first order autocorrelations, meaning that their data only have a linear correlation with their own data from the previous time step. Then we use a vector autoregressive process to simulate one market player (P1) that holds data with first order correlations with the other agents. Assuming each time step is one hour, the synthetic data are generated by

$$\mathbf{z}_t = \begin{bmatrix} 0.08 & 0.18 & 0.16 & 0.14 & 0.12 \\ 0 & 0.9 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0.7 & 0 \\ 0 & 0 & 0 & 0 & 0.6 \end{bmatrix} \mathbf{z}_{t-1} + \boldsymbol{\epsilon}_t,$$

where $\mathbf{z}_t = [z_{1,t}\ z_{2,t}\ \cdots\ z_{5,t}]^\top$ denote both the target and feature values of players (P1–P5), and $\boldsymbol{\epsilon}_t = [\epsilon_{1,t}\ \epsilon_{2,t}\ \cdots\ \epsilon_{5,t}]^\top$ represent the error terms: $\epsilon_{i,t} \sim \mathcal{N}(0,1)$.

Focusing on P1 and P2 as central agents, we let the market operator conduct their analytics tasks using different regression methods and obtain the estimated coefficients to compare with the known real coefficients. With a fixed maximum lag of 3 h and a total training time of 10 days, the market operator returns the results shown in Fig. 1 for P1 and Fig. 2 for P2. Three regression methods are compared: (a) OLS regression on the central agent's own data, (b) OLS regression with all agents' data, and (c) lasso regression with all agents' data. For P1, method (a) cannot capture the correlations with the support agents, and method (b) overfits the data by estimating non-zero coefficients on features that are not correlated with the central agent's data. Method (c) most successfully identifies the non-zero coefficients, albeit shrinking their values due to the lasso regularization. For P2, method (b) overfits the data again, while method (c) yields similar results as method (a), well capturing the independence as well as the first-order autocorrelation of P2's data.

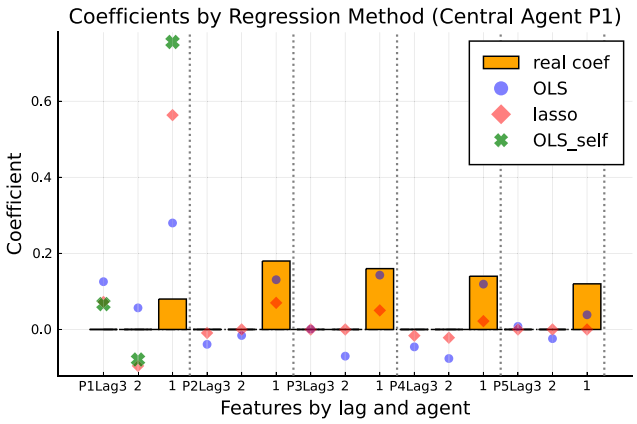

Fig. 1. Comparison of regression coefficients by regression method of P1 that has a target variable highly correlated with support agents' data.

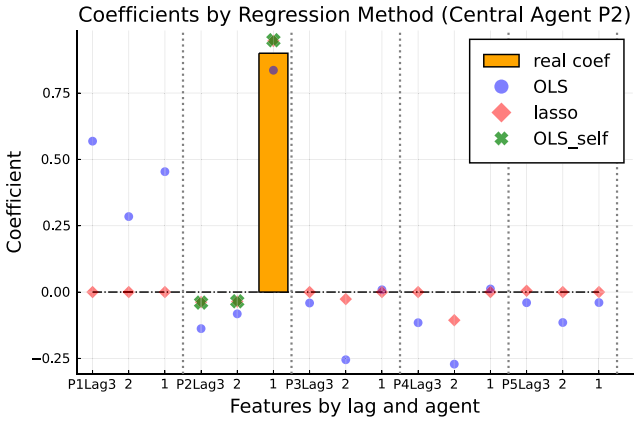

Fig. 2. Comparison of regression coefficients by regression method of P2 that has a target variable uncorrelated with support agents' data.

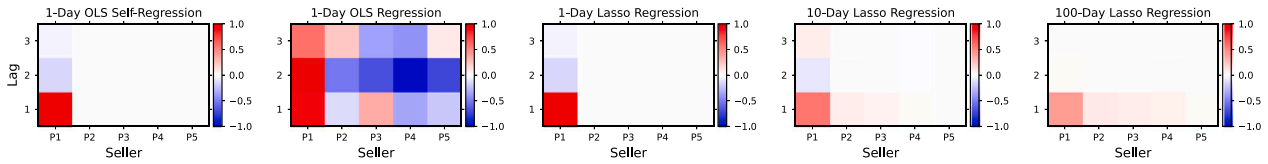

Fig. 3. Regression coefficients of P1 that has a target variable correlated with support agents' data, varying by the time scale of regression.

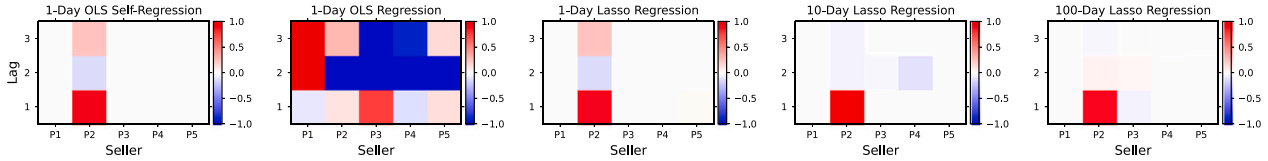

Fig. 4. Regression coefficients of P2 that has a target variable uncorrelated with support agents' data, varying by the time scale of regression.

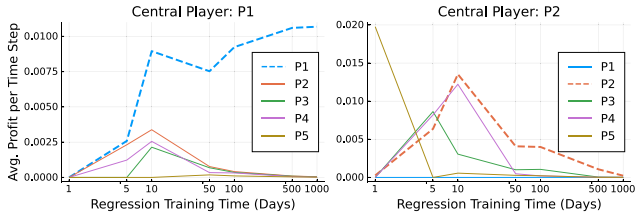

Fig. 5. Average profit per time step of all players with P1 and P2 as the central agents, respectively.

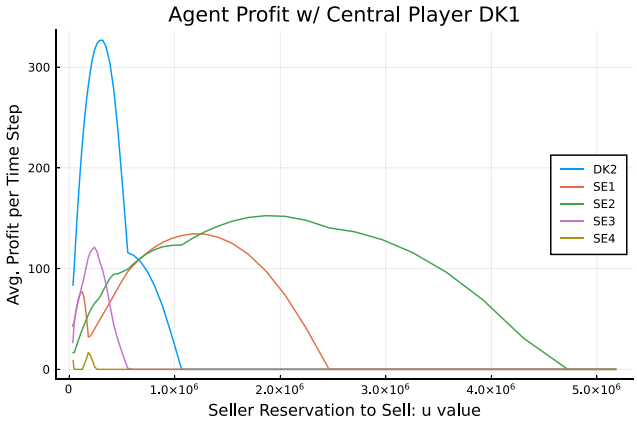

Fig. 6. Support agent profits with varying u values (central agent: DK1, training time: 10 days).

To see how the results from the regression change as the training time increases, Figs. 3 and 4 provide a visual representation of the estimated coefficients. For P1, lasso regression demonstrates a clear advantage over OLS regression for reducing overfitting, and over OLS self-regression for being able to capture correlations with features from support agents. For P2, even though the lasso regression cannot provide much additional benefit to the central agent, it is still effective in identifying the agent's independence and reducing overfitting.

Next, we evaluate the impact of the training time on the profits of the agents. Here, we reiterate that the analyses done in this paper are in-sample, meaning the central agent's profit only reflects the forecast improvement during the training period. As Fig. 5 shows, increasing the training time T eventually reduces the support agents' average profit per time step to zero regardless of the correlations between the support agents and the central agent. This is because the payment term $\left|u_i^d \beta_i^{d\,L_1}\right|$ from (31) does not scale with T. In practice, in order to fulfill a certain revenue threshold per time step, the support agents can take T into account when setting their reservation to sell (e.g., setting the u value in proportion to the length of their offered support features). Meanwhile, P1 and P2 as central agents have very different profit trajectories as T increases. For P1, as their data are highly correlated with the support agents, more training time improves the estimates of the coefficients, hence increasing their profit. For P2, as their data are independent of others, the improvement of their analytics task is limited to having the lasso shrink the untrue coefficients of their own data, and as the training time increases, this minor improvement also reduces to zero. Note that in reality, an agent with data that are completely independent from others would not have the motivation to participate in a data market.

5.2. Regression market implemented on real data

To test our proposed regression market on real-world data, we obtain zonal wind power data for Denmark and Sweden from the open source Nord Pool data repository. As an illustrative example, these data are aggregated in six zones[2] (DK1, DK2, SE1, SE2, SE3, and SE4) to represent six players in the regression market. In practice, since the clearing of the regression market is decoupled from the energy market, a central agent with assets in multiple energy market zones can participate in the regression market by adjusting their loss function to truthfully reflect their financial situation in the energy market. The main purpose of this case study is to examine the impact of the support agents' reservation to sell on the final market outcomes.

Assigning DK1 as the central agent, and the others as support agents for the regression market, we use 1 h as the time step, 1 h ahead as the forecast horizon, 5 h as the maximum lag, and 10 days (240 h) as the training time. First, we assume all the support agents' reservation to sell (u_i^d) for all the features to be the same value u. Varying its value, we plot the payments for all the support agents in Fig. 6. In general, every agents' profit first increases with the u value, then peaks, and then reduces to zero. To explain this, we recall the payment term $\left|u_i^d \beta_i^{d\,L_1}\right|$. When $u_i^d = u = 0$, the payment is zero, but β_i^d is at its peak due to a lack of shrinkage. As u increases, the profit increases, but meanwhile more shrinkage is applied to β_i^d. A peak appears when the trade-off between the two opposite forces reaches a balance, but afterwards the lasso gradually shrinks β_i^d to zero, and the profit becomes zero again. The position of the peak may have to do with the correlation and the magnitude of the support agent's data. This is an interesting topic for future work.

Lastly, we select from Fig. 6 the two agents whose profits peak the last to analyze the mutual impact of their reservation to sell on each

[2] https://www.nordpoolgroup.com/the-power-market/Bidding-areas/.

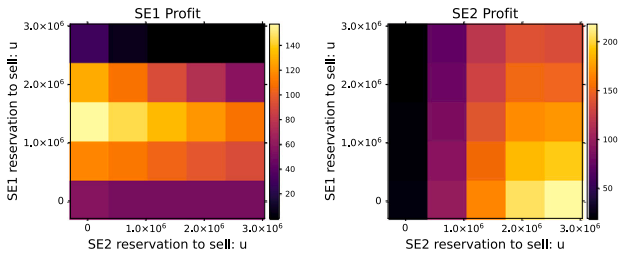

Fig. 7. SE1 and SE2 profits with varying u values of SE1 and SE2 (central agent: DK1, training time: 10 days).

other's profit. Here, we allow the reservation to sell for SE1 and SE2 to be different while fixing the other agents' u value, and show their resulting profits in Fig. 7. It is observed that within the demonstrated range, SE2 benefits from higher u values, while SE1 achieves the highest profit in the middle range. Meanwhile, each player's profit decreases as the other player's reservation to sell increases. So far, the reservation to sell has been used as a customized parameter by the support agents to ensure a revenue threshold, but the results from Fig. 7 raise the question of whether the support agents could instead strategically set the reservation to sell to maximize their gain in the regression market. Future research can examine the interplay of the reservation to sell of more than two players and the corresponding market equilibrium.

6. Conclusion

Adopting the lasso regularizer, we construct a regression market for wind agents to trade wind power data to improve forecasting. Each support agent as a data seller has the freedom to determine their reservation to sell each feature they own, which is incorporated in the lasso term of the central agent's analytics task under linear regression. The product of a support agent's reservation to sell a feature and the absolute value of the corresponding estimated coefficient is directly computed as the payment from the central agent for the sold feature. This market framework is proved to meet the support agents' profit requirements while guaranteeing financial benefits for the central agent.

Some immediate future work includes the incorporation of out-of-sample analyses, the strategies to set the reservation to sell from the data sellers' perspective for maximizing individual gains, and the extension to an online market. This regression market can be applied to other use cases, where the data sellers have individual revenue requirements on the data sold to the data buyer. It can also be readily extended to a multi-buyer framework since the analytics tasks of multiple agents can be conducted simultaneously and the outcome of each agent's task does not affect the task of another.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Smart4RES project (European Union's Horizon 2020, No. 864337). The open source and easily accessible market data from Nord Pool have provided the basis for case studies in this work. In addition, the authors would like to acknowledge the fruitful discussions on data markets with Ricardo J. Bessa and Carla Gonçalves at INESC TEC, as well as the useful tips on autoregressive models provided by Amandine Pierrot at DTU.

References

[1] L. Veldkamp, Data and the aggregate economy, in: Annual Meeting Plenary, Society for Economic Dynamics, 2019.
[2] J.M. Morales, A.J. Conejo, H. Madsen, P. Pinson, M. Zugno, Integrating Renewables in Electricity Markets: Operational Problems, Springer, 2014.
[3] J. Jung, R.P. Broadwater, Current status and future advances for wind speed and power forecasting, Renew. Sustain. Energy Rev. 31 (2014) 762–777.
[4] J. Tastu, P. Pinson, H. Madsen, Multivariate conditional parametric models for a spatio-temporal analysis of short-term wind power forecast errors, in: Proceedings of the European Wind Energy Conference, 2010.
[5] J.R. Andrade, R.J. Bessa, Improving renewable energy forecasting with a grid of numerical weather predictions, IEEE Trans. Sustain. Energy 8 (4) (2017) 1571–1580.
[6] F. Farokhi, Review of results on smart-meter privacy by data manipulation, demand shaping, and load scheduling, IET Smart Grid 3 (5) (2020) 605–613.
[7] C. Goncalves, P. Pinson, R.J. Bessa, Towards data markets in renewable energy forecasting, IEEE Trans. Sustain. Energy 12 (1) (2021) 533–542.
[8] D. Bergemann, A. Bonatti, Markets for information: An introduction, Annu. Rev. Econ. 11 (2019) 85–107.
[9] P. Pinson, L. Han, J. Kazempour, Regression markets and application to energy forecasting, TOP (2022).
[10] R. Montes, W. Sand-Zantman, T. Valletti, The value of personal information in online markets with endogenous privacy, Manage. Sci. 65 (3) (2019) 1342–1362.
[11] D. Acemoglu, A. Makhdoumi, A. Malekian, A. Ozdaglar, Too much data: Prices and inefficiencies in data markets, Am. Econ. J.: Microecon.: Micro (2021) forthcoming.
[12] K. Bimpikis, D. Crapis, A. Tahbaz-Salehi, Information sale and competition, Manage. Sci. 65 (6) (2019) 2646–2664.
[13] A. Agarwal, M. Dahleh, T. Sarkar, A marketplace for data: An algorithmic solution, in: Proceedings of the 2019 ACM Conference on Economics and Computation, 2019, pp. 701–726.
[14] R. Girard, D. Allard, Spatio-temporal propagation of wind power prediction errors, Wind Energy 16 (7) (2013) 999–1012.
[15] M. He, L. Yang, J. Zhang, V. Vittal, A spatio-temporal analysis approach for short-term forecast of wind farm generation, IEEE Trans. Power Syst. 29 (4) (2014) 1611–1622.
[16] P. Pinson, Introducing distributed learning approaches in wind power forecasting, in: 2016 International Conference on Probabilistic Methods Applied to Power Systems, IEEE, 2016.
[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2010) 1–122.
[18] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. 58 (1) (1996) 267–288.
[19] L. Cavalcante, R.J. Bessa, M. Reis, J. Browell, LASSO vector autoregression structures for very short-term wind power forecasting, Wind Energy 20 (4) (2017) 657–675.
[20] J.W. Messner, P. Pinson, Online adaptive lasso estimation in vector autoregressive models for high dimensional wind power forecasting, Int. J. Forecast. 35 (4) (2019) 1485–1498.
[21] D. Obst, P. Pinson, Distributed learning for high-dimensional wind energy production forecasting, Tech. Rep 1–61, Technical University of Denmark, 2017, [Online]. Available: https://davidobst.github.io/Rapport_PRe_Obst_David.pdf.