International Journal of Forecasting 37 (2021) 322–342

Contents lists available at ScienceDirect

International Journal of Forecasting

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

A critical overview of privacy-preserving approaches for collaborative forecasting

Carla Gonçalves [a,b], Ricardo J. Bessa [a,*], Pierre Pinson [c]

[a] INESC TEC – Institute for Systems and Computer Engineering, Technology and Science, Porto, Portugal
[b] Faculty of Sciences of the University of Porto, Portugal
[c] Technical University of Denmark, Kongens Lyngby, Denmark

ARTICLE INFO

Keywords:
Vector autoregression
Forecasting
Time series
Privacy-preserving
ADMM

ABSTRACT

Cooperation between different data owners may lead to an improvement in forecast quality—for instance, by benefiting from spatiotemporal dependencies in geographically distributed time series. Due to business competitive factors and personal data protection concerns, however, said data owners might be unwilling to share their data. Interest in collaborative privacy-preserving forecasting is thus increasing. This paper analyzes the state-of-the-art and unveils several shortcomings of existing methods in guaranteeing data privacy when employing vector autoregressive models. The methods are divided into three groups: data transformation, secure multi-party computations, and decomposition methods. The analysis shows that state-of-the-art techniques have limitations in preserving data privacy, such as (i) the necessary trade-off between privacy and forecasting accuracy, empirically evaluated through simulations and real-world experiments based on solar data; and (ii) iterative model fitting processes, which reveal data after a number of iterations.

© 2020 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

1. Introduction

The progress of the internet-of-things (IoT) and big data technologies is fostering a disruptive evolution in the development of innovative data analytics methods and algorithms. This also yields ideal conditions for data-driven services (from descriptive to prescriptive analysis), in which the accessibility to large volumes of data is a fundamental requirement. In this sense, the combination of data from different owners can provide valuable information for end-users and increase their competitiveness.

In order to combine data coming from different sources, several statistical approaches have emerged. For example, in time series collaborative forecasting, the vector autoregressive (VAR) model has been widely used to forecast variables that may have different data owners. In the energy sector, the VAR model is deemed appropriate to update very short-term forecasts (e.g., from 15 min to 6 h ahead) with recent data, thus taking advantage of geographically distributed data collected from sensors (e.g., anemometers and pyranometers) and/or wind turbines and solar power inverters (Bessa, Trindade et al., 2015; Tastu et al., 2013). The VAR model can also be used in short-term electricity price forecasting (Ziel & Weron, 2018). Furthermore, the large number of potential data owners favors the estimation of the VAR model's coefficients by applying distributed optimization algorithms. The alternating direction method of multipliers (ADMM) is a widely used convex optimization technique; see Boyd et al. (2011). The combination of the VAR model and ADMM can be used jointly for collaborative forecasting (Cavalcante et al., 2017), which consists of collecting and combining information from diverse owners. Collaborative forecasting methods require sharing data or coefficients, depending on the structure of the data, and

* Corresponding author.
E-mail addresses: carla.s.goncalves@inesctec.pt (C. Gonçalves), ricardo.j.bessa@inesctec.pt (R.J. Bessa), ppin@elektro.dtu.dk (P. Pinson).

https://doi.org/10.1016/j.ijforecast.2020.06.003
0169-2070/© 2020 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

may or may not be focused on data privacy. This process is also called federated learning (Yang et al., 2019).

Some other examples of collaborative forecasting include: (a) forecasting and inventory control in supply chains, in which the benefits of various types of information-sharing options are investigated (Aviv, 2003, 2007); (b) forecasting traffic flow (i.e., traffic speed) at different locations (Ravi & Al-Deek, 2009); and (c) forecasting retail prices of a specific product at every outlet by using historical retail prices of the product at a target outlet and at competing outlets (Ahmad et al., 2016). The VAR model is the simplest collaborative model, but conceptually, a collaborative forecasting model for time series does not need to be a VAR. Furthermore, it is possible to extend the VAR model to include exogenous information (see Nicholson et al. (2017) for more details) and to model non-linear relationships with past values (e.g., Li and Genton (2009) extend the additive model structure to a multivariate setting).

Setting aside the significant potential of the VAR model for collaborative forecasting, the concerns with the privacy of personal and commercially sensitive data constitute a critical barrier and require privacy-preserving algorithmic solutions for estimating the coefficients of the model.

A confidentiality breach occurs when third parties recover without consent any data provided in confidence. A single record leaked from a dataset is of more or less importance depending on the nature of the data. For example, in medical data, where each record represents a different patient, a single leaked record can disclose all the details about a patient. By contrast, with renewable energy generation time series, the knowledge that 30 MWh was produced in a given hour is not very relevant to a competitor. Hereafter, the term confidentiality breach designates the reconstruction of the entire dataset by another party.

These concerns with data confidentiality motivated research into methods that can handle confidential data, such as linear regression and classification problems (Du et al., 2004), ridge linear regression (Karr et al., 2009), logistic regression (Wu et al., 2012), survival analysis (Lu et al., 2015), and aggregated statistics for time series data (Jia et al., 2014). Aggregated statistics consist of aggregating a set of time series data through a specific function, such as the average (e.g., the average amount of daily exercise), sum, minimum, and maximum. However, certain approaches are vulnerable to confidentiality breaches, showing that the statistical methods developed to protect data privacy should be analyzed to confirm their robustness, and that additional research may be required to address overlooked limitations (Fienberg et al., 2009). Furthermore, the application of these methods to the VAR model needs to be carefully analyzed, since the target variables are the time series of each data owner, and the covariates are the lags of the same time series, meaning that both target and covariates share a large proportion of values.

The simplest solution would be to have the data owners agree on a commonly trusted entity (or a central node) capable of gathering private data, solving the associated model's fitting problem on behalf of the data owners, and then returning the results (Pinson, 2016). However, in many cases, the data owners are unwilling to share their data even with a trusted central node. This has motivated the development of data markets to monetize data and promote data sharing (Agarwal et al., 2019), which can be driven by blockchain and smart contracts technology (Kurtulmus & Daniel, 2018).

Another possibility would be to apply differential privacy mechanisms, which consist of adding properly calibrated noise to an algorithm (e.g., adding noise to the coefficients estimated during each iteration of the fitting procedure) or directly to the data. Differential privacy is not an algorithm, but rather a rigorous definition of privacy that is useful for quantifying and bounding privacy loss (i.e., how much original data a party can recover when receiving data protected with added noise) (Dwork & Smith, 2009). It requires computations insensitive to changes in any particular record or intermediate computations, thereby restricting data leaks through the results; see Appendix A for an elaboration. While computationally efficient and popular, these techniques invariably degrade the predictive performance of the model (Yang et al., 2019) and are not very effective, as we show in what follows.

This paper is a review of the state-of-the-art in statistical methods for collaborative forecasting with privacy-preserving approaches. This work is not restricted to a simple overview of the existing methods. It includes a critical evaluation of said methods from a mathematical and numerical point of view—namely, when applied to the VAR model. The major contribution to the literature is to show gaps and downsides to current methods and to present insights for further improvements towards fully privacy-preserving VAR forecasting methods.

In this work, we analyze existing state-of-the-art privacy-preserving techniques, dividing them into the following groups:

- *Data transformation methods*: each data owner transforms the data before the model's fitting process, by adding randomness to the original data in such a way that high accuracy and privacy can be achieved at the end of the fitting process. The statistical method is independent of the transformation function and it is applied to the transformed data.
- *Secure multi-party computation protocols*: data encryption occurs while fitting the statistical model (i.e., intermediate calculations of an iterative process) and data owners are required to conjointly compute a function over their data with protocols for secure matrix operations. A protocol consists of rules that determine how data owners must operate to determine said function. These rules establish the calculations assigned to each data owner, what information should be shared among them, and the conditions necessary for the adequate implementation of said calculations.
- *Decomposition-based methods*: the optimization problem is decomposed into sub-problems, allowing each data owner to fit model coefficients separately.

The remainder of the paper is organized as follows: Section 2 describes the state-of-the-art for collaborative privacy-preserving forecasting. Section 3 describes the VAR model, as well as coefficients estimators, and critically evaluates state-of-the-art methods when applied to the VAR model. Solar energy time series data are used in the numerical analysis. Section 4 offers a discussion and comparison of the presented approaches, and conclusions are presented in Section 5.

2. Privacy-preserving approaches

For notation purposes, vectors and matrices are denoted by bold lowercase and bold uppercase letters, e.g., $\mathbf{a}$ and $\mathbf{A}$, respectively. The vector $\mathbf{a} = [a_1, \ldots, a_k]^\top$ represents a column vector with k dimensions, where a_i denotes scalars and $i = 1, \ldots, k$. The column-wise joining of vectors and matrices is indicated by $[\mathbf{a}, \mathbf{b}]$ and $[\mathbf{A}, \mathbf{B}]$, respectively.

Furthermore, $\mathbf{Z} \in \mathbb{R}^{T\times M}$ is the covariate matrix and $\mathbf{Y} \in \mathbb{R}^{T\times N}$ is the target matrix, considering n data owners. The values T, M, and N are the number of records, covariates, and target variables, respectively. When considering collaborative forecasting models, different divisions of the data may be considered. Fig. 1 shows the two most common:

1. *Data split by records:* the data owners, represented as $A_i, i = 1, \ldots, n$, observe the same features for different groups of samples, e.g., different timestamps in the case of time series. $\mathbf{Z}$ is split into $\mathbf{Z}^r_{A_i} \in \mathbb{R}^{T_{A_i}\times M}$ and $\mathbf{Y}$ into $\mathbf{Y}^r_{A_i} \in \mathbb{R}^{T_{A_i}\times N}$, such that $\sum_{i=1}^{n} T_{A_i} = T$;
2. *Data split by features:* the data owners observe different features of the same records. $\mathbf{Z} = [\mathbf{Z}_{A_1}, \ldots, \mathbf{Z}_{A_n}]$, $\mathbf{Y} = [\mathbf{Y}_{A_1}, \ldots, \mathbf{Y}_{A_n}]$, such that $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times M_{A_i}}$, $\mathbf{Y}_{A_i} \in \mathbb{R}^{T\times N_{A_i}}$, with $\sum_{i=1}^{n} M_{A_i} = M$ and $\sum_{i=1}^{n} N_{A_i} = N$.

This section summarizes state-of-the-art approaches to deal with privacy-preserving collaborative forecasting methods. Section 2.1 describes methods that ensure confidentiality by transforming data. Section 2.2 presents and analyzes secure multi-party protocols. Section 2.3 describes decomposition-based methods.

2.1. Data transformation methods

Data transformation methods use operator $\mathcal{T}$ to transform the data matrix $\mathbf{X}$ into $\tilde{\mathbf{X}} = \mathcal{T}(\mathbf{X})$. Then, the problem is solved in the transformed domain. A common method of masking sensitive data is adding or multiplying it by perturbation matrices. In additive randomization, random noise is added to the data in order to mask the values of records. Consequently, the more masked the data becomes, the more secure it will be, as long as the differential privacy definition is respected (see Appendix A). However, the use of randomized data implies the deterioration of the estimated statistical models, and the estimated coefficients of said data should be close to the estimated coefficients after using original data (Zhou et al., 2009).

Multiplicative randomization involves changing the dimensions of the data by multiplying it by random perturbation matrices. If the perturbation matrix $\mathbf{W} \in \mathbb{R}^{k\times m}$ multiplies the original data $\mathbf{X} \in \mathbb{R}^{m\times n}$ on the left (pre-multiplication), i.e., $\mathbf{WX}$, then it is possible to change the number of records; otherwise, if $\mathbf{W} \in \mathbb{R}^{n\times s}$ multiplies $\mathbf{X} \in \mathbb{R}^{m\times n}$ on the right (post-multiplication), i.e., $\mathbf{XW}$, it is possible to modify the number of features. Hence, it is possible to change both dimensions by applying both pre- and post-multiplication by perturbation matrices.

2.1.1. Single data owner

The use of linear algebra to mask data is a common practice in recent outsourcing approaches, in which a data owner resorts to the cloud to fit model coefficients without sharing confidential data. For example, in Ma et al. (2017) the coefficients that optimize the linear regression model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{1}$$

with covariate matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, target variable $\mathbf{y} \in \mathbb{R}^m$, coefficient vector $\boldsymbol{\beta} \in \mathbb{R}^n$, and error vector $\boldsymbol{\varepsilon} \in \mathbb{R}^m$, are estimated through the regularized least squares estimate for the ridge linear regression, with penalization term $\lambda > 0$:

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y}. \tag{2}$$

In order to compute $\hat{\boldsymbol{\beta}}_{\text{ridge}}$ via a cloud server, the authors consider that

$$\hat{\boldsymbol{\beta}}_{\text{ridge}} = \mathbf{A}^{-1}\mathbf{b}, \tag{3}$$

where $\mathbf{A} = (\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}$ and $\mathbf{b} = \mathbf{X}^\top\mathbf{y}$, $\mathbf{A} \in \mathbb{R}^{n\times n}$, $\mathbf{b} \in \mathbb{R}^n$. Then, the masked matrices $\mathbf{MAN}$ and $\mathbf{M}(\mathbf{b} + \mathbf{Ar})$ are sent to the server, which computes

$$\hat{\boldsymbol{\beta}}' = (\mathbf{MAN})^{-1}(\mathbf{M}(\mathbf{b} + \mathbf{Ar})), \tag{4}$$

where $\mathbf{M}$, $\mathbf{N}$, and $\mathbf{r}$ are randomly generated matrices, $\mathbf{M}, \mathbf{N} \in \mathbb{R}^{n\times n}$, $\mathbf{r} \in \mathbb{R}^n$. Finally, the data owner receives $\hat{\boldsymbol{\beta}}'$ and recovers the original coefficients by computing $\hat{\boldsymbol{\beta}}_{\text{ridge}} = \mathbf{N}\hat{\boldsymbol{\beta}}' - \mathbf{r}$.

Data normalization is a data transformation approach that masks data by transforming the original features into a new range through the use of a mathematical function. There are many methods of data normalization, the most important ones being z-score and min–max normalization (Jain & Bhandare, 2011), which are useful when the actual minimum and maximum values of the features are unknown. However, in many applications, these values are either known or publicly available, and normalized values still encompass commercially valuable information.

For time series data, other approaches to data randomization make use of the Fourier and wavelet transforms. A Fourier transform can represent periodic time series as a linear combination of sinusoidal components (sine and cosine). In Papadimitriou et al. (2007), each data owner generates a noise time series by (i) adding Gaussian noise to relevant coefficients, or (ii) disrupting each sinusoidal component by randomly changing its magnitude and phase. Similarly, a wavelet transform

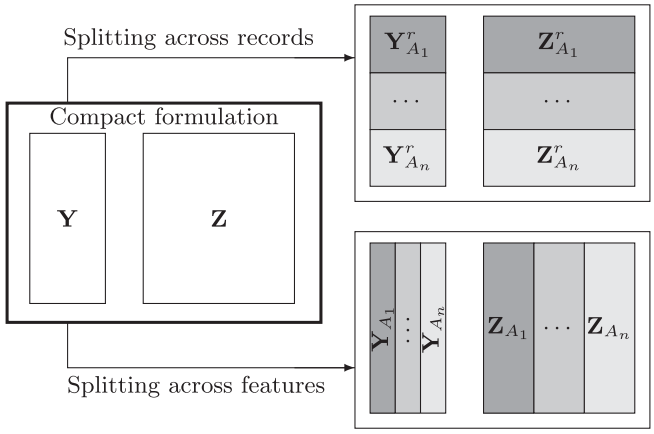


Fig. 1. Common data division structures.

can represent time series as a combination of functions (e.g., the Mexican hat or Poisson wavelets), and randomness can be introduced by adding random noise to the coefficients (Papadimitriou et al., 2007). However, there are no privacy guarantees, since noise does not respect any formal definition, unlike differential privacy.

2.1.2. Multiple data owners

The task of masking data is even more challenging when dealing with different data owners, since it is crucial to ensure that the transformations that data owners make to their data preserve the real relationship between the variables or the time series.

Usually, for generalized linear models (e.g., linear regression models, logistic regression models, etc.), where n data owners observe the same features—i.e., data are split by records, as illustrated in Fig. 1—each data owner A_i, $i = 1, \ldots, n$ can individually multiply the covariate matrix $\mathbf{Z}^r{}_{A_i} \in \mathbb{R}^{T_{A_i} \times M}$ and target variable $\mathbf{Y}^r{}_{A_i} \in \mathbb{R}^{T_{A_i} \times N}$ by a random matrix $\mathbf{M}_{A_i} \in \mathbb{R}^{k \times T_{A_i}}$ (with a jointly defined k value), providing $\mathbf{M}_{A_i}\mathbf{Z}^r{}_{A_i}$, $\mathbf{M}_{A_i}\mathbf{Y}^r{}_{A_i}$ to the competitors (Mangasarian, 2012; Yu et al., 2008), which allows pre-multiplying the original data,

$$\mathbf{Z}^r = \begin{bmatrix} \mathbf{Z}^r{}_{A_1} \\ \vdots \\ \mathbf{Z}^r{}_{A_n} \end{bmatrix} \text{ and } \mathbf{Y}^r = \begin{bmatrix} \mathbf{Y}^r{}_{A_1} \\ \vdots \\ \mathbf{Y}^r{}_{A_n} \end{bmatrix},$$

by $\mathbf{M} = [\mathbf{M}_{A_1}, \ldots, \mathbf{M}_{A_n}]$, since

$$\mathbf{M}\mathbf{Z}^r = \mathbf{M}_{A_1}\mathbf{Z}^r{}_{A_1} + \cdots + \mathbf{M}_{A_n}\mathbf{Z}^r{}_{A_n}. \tag{5}$$

The same holds for the multiplication $\mathbf{M}\mathbf{Y}^r$, $\mathbf{M} \in \mathbb{R}^{k \times \sum_{i=1}^{n} T_{A_i}}$, $\mathbf{Z}^r \in \mathbb{R}^{\sum_{i=1}^{n} T_{A_i} \times M}$, $\mathbf{Y}^r \in \mathbb{R}^{\sum_{i=1}^{n} T_{A_i} \times N}$. This definition of $\mathbf{M}$ is possible because, when multiplying $\mathbf{M}$ and $\mathbf{Z}^r$, the jth column of $\mathbf{M}$ only multiplies the jth row of $\mathbf{Z}^r$. For some statistical learning algorithms, a property of such a matrix is orthogonality, i.e., $\mathbf{M}^{-1} = \mathbf{M}^\top$. Model fitting is then performed with this new representation of the data, which preserves the solution to the problem. This is true of the linear regression model because the multivariate least squares estimate for the linear regression model with covariate matrix $\mathbf{M}\mathbf{Z}^r$ and target variable $\mathbf{M}\mathbf{Y}^r$ is

$$\hat{\mathbf{B}}_{\text{LS}} = \left((\mathbf{Z}^r)^\top \mathbf{Z}^r\right)^{-1} \left((\mathbf{Z}^r)^\top \mathbf{Y}^r\right), \tag{6}$$

which is also the multivariate least squares estimate for the coefficients of a linear regression considering data matrices $\mathbf{Z}^r$ and $\mathbf{Y}^r$, respectively. Despite this property, the application in least absolute shrinkage and selection operator (LASSO) regression does not guarantee that the sparsity of the coefficients is preserved, and careful analysis is needed to ensure the correct estimation of the model (Zhou et al., 2009). Liu et al. (2008) discussed attacks based on prior knowledge, in which a data owner estimates $\mathbf{M}$ by knowing a small collection of original data records. Furthermore, when considering the linear regression model for which $\mathbf{Z} = [\mathbf{Z}_{A_1}, \ldots, \mathbf{Z}_{A_n}]$ and $\mathbf{Y} = [\mathbf{Y}_{A_1}, \ldots, \mathbf{Y}_{A_n}]$, i.e., data are split by features, it is not possible to define a matrix $\mathbf{M}^* = [\mathbf{M}^*_{A_1}, \ldots, \mathbf{M}^*_{A_n}] \in \mathbb{R}^{k \times T}$ and then privately compute $\mathbf{M}^*\mathbf{Z}$ and $\mathbf{M}^*\mathbf{Y}$, because, as explained, the jth column of $\mathbf{M}^*$ multiplies the jth row of $\mathbf{Z}$, which, in this case, consists of data coming from different owners.

Similarly, if the data owners observe different features, a linear programming problem can be solved in such a way that individual data owners multiply their data $\mathbf{X}_{A_i} \in \mathbb{R}^{T \times M_{A_i}}$ by a private random matrix $\mathbf{N}_{A_i} \in \mathbb{R}^{M_{A_i} \times s}$ (with a jointly defined value s) and then share $\mathbf{X}_{A_i}\mathbf{N}_{A_i}$ (Mangasarian, 2011), $i = 1, \ldots, n$, which is equivalent to post-multiplying the original dataset $\mathbf{X} = [\mathbf{X}_{A_1}, \ldots, \mathbf{X}_{A_n}]$ by $\mathbf{N} = [\mathbf{N}^\top_{A_1}, \ldots, \mathbf{N}^\top_{A_n}]^\top$, which represents the joining of $\mathbf{N}_{A_i}$, $i = 1, \ldots, n$, through a row-wise operation. However, the obtained solution is in a different space, and it needs to be recovered by multiplying it by the corresponding $\mathbf{N}_{A_i}$, $i = 1, \ldots, n$. For linear regression, which models the relationship between the covariates $\mathbf{Z} \in \mathbb{R}^{T \times M}$ and the target $\mathbf{Y} \in \mathbb{R}^{T \times N}$, this algorithm corresponds to solving a linear regression that models the relationship between $\mathbf{Z}\mathbf{N}_\mathbf{z}$ and $\mathbf{Y}\mathbf{N}_\mathbf{y}$. That is, the solution is given by

$$\hat{\mathbf{B}}'_{\text{LS}} = \underset{\mathbf{B}}{\operatorname{argmin}} \left(\frac{1}{2} \|\mathbf{Y}\mathbf{N}_\mathbf{y} - \mathbf{Z}\mathbf{N}_\mathbf{z}\mathbf{B}\|_2^2 \right), \tag{7}$$

where $\mathbf{Z}\mathbf{N}_\mathbf{z}$ and $\mathbf{Y}\mathbf{N}_\mathbf{y}$ are shared matrices. Two private matrices $\mathbf{N}_\mathbf{z} \in \mathbb{R}^{M \times s}$, $\mathbf{N}_\mathbf{y} \in \mathbb{R}^{N \times w}$ are required to transform

the data, since the number of columns for $\mathbf{Z}$ and $\mathbf{Y}$ is different (s and w values are jointly defined). The problem is that the multivariate least squares estimate for (7) is given by

$$\begin{aligned}\hat{\mathbf{B}}'_{\text{LS}} &= \left((\mathbf{ZN_z})^\top(\mathbf{ZN_z})\right)^{-1}\left((\mathbf{ZN_z})^\top(\mathbf{YN_y})\right)\\ &= (\mathbf{N_z})^{-1}\quad \underbrace{(\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\mathbf{Y}}_{=\text{argmin}_{\mathbf{B}}\left(\frac{1}{2}\|\mathbf{Y}-\mathbf{ZB}\|_2^2\right)}\quad \mathbf{N_y},\end{aligned} \tag{8}$$

which implies that this transformation does not preserve the coefficients of the linear regression considering data matrices $\mathbf{Z}$ and $\mathbf{Y}$, respectively, and therefore $\mathbf{N_z}$ and $\mathbf{N_y}$ would have to be shared.

Generally, data transformation is performed through the generation of random matrices that pre- or post-multiply the private data. However, there are other techniques through which data are transformed with matrices defined according to that data, as with principal component analysis (PCA). PCA is a widely used statistical procedure for reducing the dimensions of data, by applying an orthogonal transformation that retains as much data variance as possible. Considering the matrix $\mathbf{W} \in \mathbb{R}^{M\times M}$ of the eigenvectors of the covariance matrix $\mathbf{Z}^\top\mathbf{Z}$, $\mathbf{Z} \in \mathbb{R}^{T\times M}$, PCA can be used to represent the data by L variables performing $\mathbf{ZN_L}$, where $\mathbf{N_L}$ denotes the first L columns of $\mathbf{W}$, $L = 1, \ldots, M$. For data split by records, Dwork et al. (2014) suggested a differentially private PCA, assuming that each data owner takes a random sample of the fitting records to form the covariate matrix. In order to protect the covariance matrix, one can add Gaussian noise to this matrix (determined without sensible data sharing), leading to the computation of the principal directions of the noisy covariance matrix. To finalize the process, the data owners multiply the sensible data by said principal directions before feeding the data into the model fitting. Nevertheless, the application to collaborative linear regression with data split by features would require sharing the data when computing the $\mathbf{Z}^\top\mathbf{Z}$ matrix, since $\mathbf{Z}^\top$ is divided by rows. Furthermore, as explained in (7) and (8), it is difficult to recover the original linear regression model by performing the estimation of the coefficients using transformed covariates and target matrices, through post-multiplication by random matrices.

Regarding the data normalization techniques mentioned above, Zhu et al. (2015) proposed that data owners mask their data by using z-score normalization, followed by the sum of random noise (from uniform or Gaussian distributions), to allow greater control over their data. The data can then be shared with a recommendation system that fits the model. However, the noise does not meet the differential privacy definition (see Appendix A).

For data collected by different sensors (e.g., smart meters or mobile users) it is common to proceed to the aggregation of data through privacy-preserving techniques—for instance, by adding carefully calibrated Laplacian noise to each time series (Fan & Xiong, 2014; Soria-Comas et al., 2017). The addition of noise to the data is an appealing technique given its easy application. However, even if this noise meets the definition of differential privacy, there is no guarantee that the resulting model will perform well.

2.2. Secure multi-party computation protocols

In secure multi-party computations, intermediate calculations required by the fitting algorithms, which require data owners to jointly compute a function over their data, are performed through protocols for secure operations, such as matrix addition or multiplication (as discussed in Section 2.2.1). In these approaches, the encryption of the data occurs while fitting the model (as discussed in Section 2.2.2), instead of as a pre-processing step, as with the data transformation methods described in the previous section.

2.2.1. Linear algebra-based protocols

The simplest secure multi-party computation protocols are based on linear algebra and address the situation where matrix operations with confidential data are necessary. Du et al. (2004) proposed secure protocols for product $\mathbf{A}.\mathbf{C}$ and the inverse of the sum $(\mathbf{A}+\mathbf{C})^{-1}$ for any two private matrices $\mathbf{A}$ and $\mathbf{C}$ with appropriate dimensions. The aim is to fit a (ridge) linear regression between two data owners who observe different covariates but share the target variable. Essentially, the $\mathbf{A}.\mathbf{C}$ protocol transforms the product of matrices, $\mathbf{A} \in \mathbb{R}^{m\times s}$, $\mathbf{C} \in \mathbb{R}^{s\times k}$, into a sum of matrices $\mathbf{V}_a + \mathbf{V}_c$ that are equally secret, $\mathbf{V}_a, \mathbf{V}_c \in \mathbb{R}^{m\times k}$. However, since the estimate of the coefficients for linear regression with covariate matrix $\mathbf{Z} \in \mathbb{R}^{T\times M}$ and target matrix $\mathbf{Y} \in \mathbb{R}^{T\times N}$ is

$$\hat{\mathbf{B}}_{\text{LS}} = (\mathbf{Z}^\top\mathbf{Z})^{-1}\mathbf{Z}^\top\mathbf{Y}, \tag{9}$$

the $\mathbf{A}.\mathbf{C}$ protocol is used to perform the computation of $\mathbf{V}_a, \mathbf{V}_c$, such that

$$\mathbf{V}_a + \mathbf{V}_c = (\mathbf{Z}^\top\mathbf{Z}), \tag{10}$$

which requires the definition of an $(\mathbf{A}+\mathbf{C})^{-1}$ protocol to compute

$$(\mathbf{Z}^\top\mathbf{Z})^{-1} = (\mathbf{V}_a + \mathbf{V}_c)^{-1}. \tag{11}$$

For the $\mathbf{A}.\mathbf{C}$ protocol, $\mathbf{A} \in \mathbb{R}^{m\times s}$, $\mathbf{C} \in \mathbb{R}^{s\times k}$, there are two different formulations according to the existence, or not, of a third entity. In cases where only two data owners perform the protocol, a random matrix $\mathbf{M} \in \mathbb{R}^{s\times s}$ is jointly generated and the $\mathbf{A}.\mathbf{C}$ protocol achieves the following results, by dividing $\mathbf{M}$ and $\mathbf{M}^{-1}$ into two matrices with the same dimensions:

$$\mathbf{AC} = \mathbf{AMM}^{-1}\mathbf{C} = \mathbf{A}[\mathbf{M}_{\text{left}}, \mathbf{M}_{\text{right}}]\begin{bmatrix}(\mathbf{M}^{-1})_{\text{top}}\\ (\mathbf{M}^{-1})_{\text{bottom}}\end{bmatrix}\mathbf{C} \tag{12}$$

$$= \mathbf{AM}_{\text{left}}(\mathbf{M}^{-1})_{\text{top}}\mathbf{C} + \mathbf{AM}_{\text{right}}(\mathbf{M}^{-1})_{\text{bottom}}\mathbf{C}, \tag{13}$$

where $\mathbf{M}_{\text{left}}$ and $\mathbf{M}_{\text{right}}$ respectively represent the left and right part of $\mathbf{M}$, and $(\mathbf{M}^{-1})_{\text{top}}$ and $(\mathbf{M}^{-1})_{\text{bottom}}$ respectively denote the top and bottom part of $\mathbf{M}^{-1}$. In this case,

$$\mathbf{V}_a = \mathbf{AM}_{\text{left}}(\mathbf{M}^{-1})_{\text{top}}\mathbf{C}, \tag{14}$$

is derived by the first data owner, and

$$\mathbf{V}_c = \mathbf{AM}_{\text{right}}(\mathbf{M}^{-1})_{\text{bottom}}\mathbf{C}, \tag{15}$$

is derived by the second data owner. Otherwise, a third entity is assumed to generate random matrices $\mathbf{R}_a, \mathbf{r}_a$ and $\mathbf{R}_c, \mathbf{r}_c$, such that

$$\mathbf{r}_a + \mathbf{r}_c = \mathbf{R}_a\mathbf{R}_c, \tag{16}$$

which are sent to the first and second data owners, respectively, $\mathbf{R}_a \in \mathbb{R}^{m\times s}$, $\mathbf{R}_c \in \mathbb{R}^{s\times k}$, $\mathbf{r}_a, \mathbf{r}_c \in \mathbb{R}^{m\times k}$. In this case, the data owners start by trading the matrices $\mathbf{A}+\mathbf{R}_a$ and $\mathbf{C}+\mathbf{R}_c$, and then the second data owner randomly generates a matrix $\mathbf{V}_c$ and sends

$$\mathbf{T} = (\mathbf{A}+\mathbf{R}_a)\mathbf{C} + (\mathbf{r}_c - \mathbf{V}_c), \tag{17}$$

to the first data owner in such a way that, at the end of the **A**.**C** protocol, the first data owner keeps the information

$$\mathbf{V}_a = \mathbf{T} + \mathbf{r}_a - \mathbf{R}_a(\mathbf{C}+\mathbf{R}_c), \tag{18}$$

and the second data owner keeps $\mathbf{V}_c$ (since the sum of $\mathbf{V}_a$ with $\mathbf{V}_c$ is $\mathbf{AC}$).

Finally, the $(\mathbf{A}+\mathbf{C})^{-1}$ protocol considers two steps, where $\mathbf{A}, \mathbf{C} \in \mathbb{R}^{m\times k}$. Initially, the matrix $(\mathbf{A}+\mathbf{C})$ is jointly converted to $\mathbf{P}(\mathbf{A}+\mathbf{C})\mathbf{Q}$ using two random matrices, **P** and **Q**, which are only known to the second data owner, preventing the first one from learning matrix **C**, $\mathbf{P} \in \mathbb{R}^{r\times m}$, $\mathbf{Q} \in \mathbb{R}^{k\times t}$. The results of $\mathbf{P}(\mathbf{A}+\mathbf{C})\mathbf{Q}$ are known only by the first data owner, who can conduct the inverse computation $\mathbf{Q}^{-1}(\mathbf{A}+\mathbf{C})^{-1}\mathbf{P}^{-1}$. In the following step, the data owners jointly remove $\mathbf{Q}^{-1}$ and $\mathbf{P}^{-1}$ and derive $(\mathbf{A}+\mathbf{C})^{-1}$. Both steps can be achieved by applying the **A**.**C** protocol. Although these protocols are efficient techniques for solving problems with a shared target variable, one cannot say the same when **Y** is private, as further elaborated in Section 3.3.2.

Another example of a secure protocol for producing private matrices can be found in Karr et al. (2009). Their protocol applies data from multiple owners who observe different covariates and target features—which are also assumed to be secret. The protocol allows two data owners, with correspondent data matrix **A** and **C**, $\mathbf{A} \in \mathbb{R}^{m\times k}$, $\mathbf{C} \in \mathbb{R}^{m\times s}$, to perform the multiplication $\mathbf{A}^\top\mathbf{C}$ as follows: (i) the first data owner generates $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_g]$, $\mathbf{W} \in \mathbb{R}^{m\times g}$, such that

$$\mathbf{w}_i^\top \mathbf{A}_j = \mathbf{0}, \tag{19}$$

where $\mathbf{A}_j$ is the jth column of **A** matrix, $i = 1, \ldots, g$ and $j = 1, \ldots, k$, and then sends **W** to the second owner; (ii) the second data owner computes $(\mathbf{I} - \mathbf{W}\mathbf{W}^\top)\mathbf{C}$ and shares it; and (iii) the first data owner performs

$$\mathbf{A}^\top(\mathbf{I} - \mathbf{W}\mathbf{W}^\top)\mathbf{C} = \mathbf{A}^\top\mathbf{C} - \underbrace{\mathbf{A}^\top\mathbf{W}\mathbf{W}^\top\mathbf{C}}_{=\mathbf{0},\ \text{since}\ \mathbf{A}^\top\mathbf{W}=\mathbf{0}} = \mathbf{A}^\top\mathbf{C}, \tag{20}$$

without the possibility of recovering **C**, since the $rank((\mathbf{I}-\mathbf{W}\mathbf{W}^\top)\mathbf{C}) = m - g$. To generate **W**, Karr et al. (2009) suggested selecting g columns from the **Q** matrix, computed by **QR** decomposition of the private matrix **C**, and excluding the first k columns. Furthermore, the authors defined the optimal value for g according to the number of linearly independent equations (represented by NLIE) on the other data owner's data. The second data owner obtains $\mathbf{A}^\top\mathbf{C}$ (providing ks values, since $\mathbf{A}^\top\mathbf{C} \in \mathbb{R}^{k\times s}$) and receives **W**, knowing that $\mathbf{A}^\top\mathbf{W} = 0$ (which contains kg values). That is,

$$\text{NLIE(Owner\#1)} = ks + kg. \tag{21}$$

Similarly, the first data owner receives $\mathbf{A}^\top\mathbf{C}$ (providing ks values) and $(\mathbf{I} - \mathbf{W}\mathbf{W}^\top)\mathbf{C}$ (providing $s(m-g)$ values since $(\mathbf{I} - \mathbf{W}\mathbf{W}^\top)\mathbf{C} \in \mathbb{R}^{m\times s}$ and $rank(\mathbf{W}) = m - g$). That is,

$$\text{NLIE(Owner\#2)} = ks + s(m-g). \tag{22}$$

Karr et al. (2009) determined the optimal value for g by assuming that both data owners equally share NLIE, so that no agent benefits from the order assumed when running the protocol:

$$|\text{NLIE(Owner\#1)} - \text{NLIE(Owner\#2)}| = 0\ , \tag{23}$$

which allows the optimal value $g^* = \frac{sm}{k+s}$ to be obtained.

An advantage to this approach, when compared to the one proposed by Du et al. (2004), is that **W** is simply generated by the first data owner, while the invertible matrix **M** proposed by Du et al. (2004) needs to be agreed upon by both parties, which entails substantial communication costs when the number of records is high.

2.2.2. Homomorphic cryptography-based protocols

The use of homomorphic encryption was successfully introduced in model fitting and works by encrypting the original values in such a way that the application of arithmetic operations in the public space does not compromise the encryption. Homomorphic encryption ensures that, after the decryption stage (in the private space), the resulting values correspond to the ones obtained by operating on the original data. Consequently, homomorphic encryption is especially responsive and engaging to privacy-preserving applications. As an example, the Paillier homomorphic encryption scheme stipulates that (i) two integer values encrypted with the same public key may be multiplied together to give an encryption of the sum of the values, and (ii) an encrypted value may be taken to some power, yielding encryption of the product of the values. Hall et al. (2011) proposed a secure protocol for summing and multiplying real numbers by extending Paillier encryption, aiming to perform the matrix products required to solve linear regression for data divided by features or records.

Equally based in Paillier encryption, Nikolaenko et al. (2013) proposed a scheme whereby two parties can correctly perform their tasks without teaming up to discover private data: a crypto-service provider (i.e., a party that provides software- or hardware-based encryption and decryption services) and an evaluator (i.e., a party who runs the learning algorithm). With this scheme, secure linear regression can be performed for data split by records. Similarly, Chen et al. (2018) used Paillier and ElGamal encryption to fit the coefficients of ridge linear regression while including these entities. In both works, the use of the crypto-service provider is prompted by assuming that the evaluator does not corrupt its computation by producing an incorrect result. Two conditions are required to prevent confidentiality breaches: the crypto-service provider must publish the system keys correctly, and there can be no collusion between the evaluator and the crypto-service provider. The data can be reconstructed if the crypto-service provider supplies correct keys to a curious evaluator. For data divided by features, Gascón et al. (2017) extended the approach of Nikolaenko et al. (2013) by designing a secure multi/two-party inner product.

Jia et al. (2018) explored a privacy-preserving data classification scheme with a support vector machine, to ensure that the data owners can successfully conduct data classification without exposing their learned models to

a "tester", while the "testers" keep their data private. For example, a hospital (owner) can create a model to learn the relation between a set of features and the existence of a disease, and another hospital (tester) can use this model to obtain forecasting values, without any knowledge about the model. The method is supported by cryptography-based protocols for secure computation of multivariate polynomial functions, but unfortunately, this only works for data split by records.

Li and Cao (2012) addresses the privacy-preserving computation of the sum and the minimum of multiple time series collected by different data owners, by combining homomorphic encryption with a novel key management technique to support large data dimensions. These statistics with a privacy-preserving solution for individual user data are quite useful for exploring mobile sensing in different applications such as environmental monitoring (e.g., the average level of air pollution in an area), traffic monitoring (e.g., the highest moving speed during rush hour), healthcare (e.g., the number of users infected by a flu), etc. Liu et al. (2018) and Li et al. (2018) explored similar approaches based on Paillier or ElGamal encryption concerning their application to smart grids. However, the estimation of models such as the linear regression model also requires protocols for the secure product of matrices.

Homomorphic cryptography was further explored to solve secure linear programming problems through intermediate steps of the simplex method, which optimizes the problem by using slack variables, tableaus, and pivot variables (Hoogh, 2012). However, the author observed that the proposed protocols are not viable when solving linear programming problems with numerous variables and constraints, which are common in practice.

Aono et al. (2017) combined homomorphic cryptography with differential privacy in order to deal with data split by records. In summary, if data are split by records, as illustrated in Fig. 1, each ith data owner observes the covariates $\mathbf{Z}^r{}_{A_i}$ and target variable $\mathbf{Y}^r{}_{A_i}$, $\mathbf{Z}^r{}_{A_i} \in \mathbb{R}^{T_{A_i}\times M}$, $\mathbf{Y}^r{}_{A_i} \in \mathbb{R}^{T_{A_i}\times N}$, $i = 1, \ldots, n$. Then, $(\mathbf{Z}^r{}_{A_i})^\top \mathbf{Z}^r{}_{A_i}$ and $(\mathbf{Z}^r{}_{A_i})^\top \mathbf{Y}^r{}_{A_i}$ are computed and Laplacian noise is added to them. This information is encrypted and sent to the cloud server, which works on the encrypted domain, summing all the matrices received. Finally, the server provides the result of this sum to a client who decrypts it and obtains relevant information to perform the linear regression, i.e., $\sum_{i=1}^n (\mathbf{Z}^r{}_{A_i})^\top \mathbf{Z}^r{}_{A_i}$, $\sum_{i=1}^n (\mathbf{Z}^r{}_{A_i})^\top \mathbf{Y}^r{}_{A_i}$, etc. However, the addition of noise can result in a poor estimation of the coefficients, limiting the performance of the model. Furthermore, this approach is not valid when data are divided by features, because $\mathbf{Z}^\top\mathbf{Z} \neq \sum_{i=1}^n \mathbf{Z}_{A_i}^\top \mathbf{Z}_{A_i}$ and $\mathbf{Z}^\top\mathbf{Y} \neq \sum_{i=1}^n \mathbf{Z}_{A_i}^\top \mathbf{Y}_{A_i}$.

In summary, cryptography-based methods are usually robust to confidentiality breaches but may require a third party to generate keys, as well as external entities to perform the computations in the encrypted domain. Furthermore, the high computational complexity is a challenge when dealing with real applications (Hoogh, 2012; Tran & Hu, 2019; Zhao et al., 2019).

2.3. Decomposition-based methods

In decomposition-based methods, problems are solved by breaking them up into smaller sub-problems and solving each separately, either in parallel or in sequence. Consequently, private data are naturally distributed between the data owners. However, this natural division requires sharing intermediate information. For that reason, some approaches combine decomposition-based methods with data transformation or homomorphic cryptography-based methods; here, we focus on these methods separately.

2.3.1. ADMM

The ADMM is a powerful algorithm that circumvents problems without a closed-form solution, such as the LASSO regression. The algorithm is efficient and well suited for distributed convex optimization, in particular for large-scale statistical problems (Boyd et al., 2011). Let E be a convex forecast error function between the true values $\mathbf{Y}$ and the forecasted values given by the model $\hat{\mathbf{Y}} = f(\mathbf{B}, \mathbf{Z})$ using a set of covariates $\mathbf{Z}$ and coefficients $\mathbf{B}$, and let R be a convex regularization function. The ADMM (Boyd et al., 2011) solves the optimization problem

$$\min_{\mathbf{B}} E(\mathbf{B}) + R(\mathbf{B}), \tag{24}$$

by splitting $\mathbf{B}$ into two variables ($\mathbf{B}$ and $\mathbf{H}$),

$$\min_{\mathbf{B},\mathbf{H}} E(\mathbf{B}) + R(\mathbf{H}) \text{ subject to } \mathbf{AB} + \mathbf{CH} = \mathbf{D}, \tag{25}$$

and using the corresponding augmented Lagrangian function formulated with the dual variable $\mathbf{U}$,

$$L(\mathbf{B}, \mathbf{H}, \mathbf{U}) = E(\mathbf{B}) + R(\mathbf{H}) + \mathbf{U}^\top(\mathbf{AB}+\mathbf{CH} - \mathbf{D}) + \frac{\rho}{2}\|\mathbf{AB} + \mathbf{CH} - \mathbf{D}\|_2^2. \tag{26}$$

The quadratic term $\frac{\rho}{2}\|\mathbf{AB}+\mathbf{CH}-\mathbf{D}\|_2^2$ provides theoretical convergence guarantees because it is strongly convex. This implies mild assumptions about the objective function. Even if the original objective function is convex, the augmented Lagrangian is strictly convex (and in some cases strongly convex) (Boyd et al., 2011).

The ADMM solution is estimated by the following iterative system:

$$\begin{cases} \mathbf{B}^{k+1} := \underset{\mathbf{B}}{\operatorname{argmin}}\, L(\mathbf{B}, \mathbf{H}^k, \mathbf{U}^k) \\ \mathbf{H}^{k+1} := \underset{\mathbf{H}}{\operatorname{argmin}}\, L(\mathbf{B}^{k+1}, \mathbf{H}, \mathbf{U}^k) \\ \mathbf{U}^{k+1} := \mathbf{U}^k + \rho(\mathbf{AB}^{k+1} + \mathbf{CH}^{k+1} - \mathbf{D}). \end{cases} \tag{27}$$

For data split by records, the consensus problem splits primal variables $\mathbf{B}$ and separately optimizes the decomposable cost function $E(\mathbf{B}) = \sum_{i=1}^n E_i(\mathbf{B}_{A_i})$ for all data owners under global consensus constraints. Considering that the sub-matrix $\mathbf{Z}^r_{A_i} \in \mathbb{R}^{T_{A_i}\times M}$ of $\mathbf{Z} \in \mathbb{R}^{T\times M}$ corresponds to the local data of the ith data owner, the coefficients $\mathbf{B}_{A_i} \in \mathbb{R}^{M\times N}$ are given by

$$\begin{aligned} &\underset{\Gamma}{\operatorname{argmin}} \sum_i E_i(\mathbf{B}_{A_i}) + R(\mathbf{H}) \\ &\text{s.t. } \mathbf{B}_{A_1} - \mathbf{H} = \mathbf{0},\ \mathbf{B}_{A_2} - \mathbf{H} = \mathbf{0}, \ldots, \mathbf{B}_{A_n} - \mathbf{H} = \mathbf{0}, \end{aligned} \tag{28}$$

where $\boldsymbol{\Gamma} = \{\mathbf{B}_{A_1}, \ldots, \mathbf{B}_{A_n}, \mathbf{H}\}$. In this case, $E_i(\mathbf{B}_{A_i})$ measures the error between the true values $\mathbf{Y}^r_{A_i}$ and the forecasted values given by the model $\hat{\mathbf{Y}}_{A_i} = f(\mathbf{B}_{A_i}, \mathbf{Z}^r_{A_i})$.

For data split by features, the sharing problem splits $\mathbf{Z}$ into $\mathbf{Z}_{A_i} \in \mathbb{R}^{T \times M_{A_i}}$, and $\mathbf{B}$ into $\mathbf{B}_{A_i} \in \mathbb{R}^{M_{A_i} \times N}$. Auxiliary $\mathbf{H}_{A_i} \in \mathbb{R}^{T \times N}$ are introduced for the ith data owner based on $\mathbf{Z}_{A_i}$ and $\mathbf{B}_{A_i}$. In this case, the sharing problem is formulated based on the decomposable cost function $E(\mathbf{B}) = E(\sum_{i=1}^{n} \mathbf{B}_{A_i})$ and $R(\mathbf{B}) = \sum_{i=1}^{n} R(\mathbf{B}_{A_i})$. Then, $\mathbf{B}_{A_i}$ is given by

$$\begin{aligned} &\underset{\boldsymbol{\Gamma}'}{\operatorname{argmin}} \; E(\sum_i \mathbf{H}_{A_i}) + \sum_i R(\mathbf{B}_{A_i}) \\ &\text{s.t. } \mathbf{Z}_{A_1}\mathbf{B}_{A_1} - \mathbf{H}_{A_1} = \mathbf{0}, \; \mathbf{Z}_{A_2}\mathbf{B}_{A_2} - \mathbf{H}_{A_2} \\ &\quad = \mathbf{0}, \ldots, \mathbf{Z}_{A_n}\mathbf{B}_{A_n} - \mathbf{H}_{A_n} = \mathbf{0}, \end{aligned} \tag{29}$$

where $\boldsymbol{\Gamma}' = \{\mathbf{B}_{A_1}, \ldots, \mathbf{B}_{A_n}, \mathbf{H}_{A_1}, \ldots, \mathbf{H}_{A_n}\}$. In this case, $E(\sum_{i=1}^{n} \mathbf{H}_{A_i})$ is related to the error between the true values $\mathbf{Y}$ and the forecasted values given by the model $\hat{\mathbf{Y}} = \sum_{i=1}^{n} f(\mathbf{B}_{A_i}, \mathbf{Z}_{A_i})$.

Undeniably, the ADMM provides a desirable formulation for parallel computing (Dai et al., 2018). However, it is not possible to ensure continuous privacy, since the ADMM requires intermediate calculations, allowing the most curious competitors to recover the data after enough iterations by solving non-linear equation systems (Bessa et al., 2018). An ADMM-based distributed LASSO algorithm, in which each data owner only communicates with its neighbor to protect data privacy, is described by Mateos et al. (2010), with applications in signal processing and wireless communications. Unfortunately, this approach is only valid in cases where data are distributed by records.

The concept of differential privacy was also explored in the ADMM by introducing randomization when computing the primal variables. That is, during the iterative process, each data owner estimates the corresponding coefficients and perturbs them by adding random noise (Zhang & Zhu, 2017). However, these local randomization mechanisms can result in a non-convergent algorithm with poor performance even under moderate privacy guarantees. To address these concerns, Huang et al. (2019) used an approximate augmented Lagrangian function and Gaussian mechanisms with time-varying variance. Nevertheless, the addition of noise is insufficient to guarantee privacy, as a competitor can potentially use the results from all iterations to infer information (Zhang et al., 2018).

Zhang et al. (2019) recently combined a variant of the ADMM with homomorphic encryption for cases where data are divided by records. As explained by the authors, however, the incorporation of their mechanism in decentralized optimization under data divided by features is quite difficult. Whereas for data split by records, the algorithm only requires sharing the coefficients, the exchange of coefficients in data split by features is insufficient, since each data owner observes different features. Division by features requires a local estimation of $\mathbf{B}^{k+1}_{A_i} \in \mathbb{R}^{M_{A_i} \times N}$ by using information related to $\mathbf{Z}_{A_j}\mathbf{B}^{k}_{A_j}$ and $\mathbf{Y}$, meaning that, for each new iteration, an ith data owner shares TN new values, instead of $M_{A_i}N$ (from $\mathbf{B}^{k}_{A_i}$), $i, j = 1, \ldots, n$.

For data split by features, Zhang and Wang (2018) proposed a probabilistic forecasting method that combines ridge linear quantile regression with the ADMM. The output is a set of quantiles instead of a unique value (usually the expected value). In this case, the ADMM is applied to split the corresponding optimization problem into sub-problems, which are solved by each data owner, assuming that all the data owners communicate with a central node in an iterative process. Consequently, intermediate results are provided, rather than private data. In fact, the authors claimed that their method achieves wind power probabilistic forecasting with off-site information in a privacy-preserving and distributed fashion. However, the authors did not conduct an in-depth analysis of the method, as shown in Section 3. Furthermore, their method assumes that the central node knows the target matrix.

2.3.2. Newton–Raphson method

The ADMM is now a standard technique used in research on distributed computing in statistical learning, but it is not the only one. For generalized linear models, distributed optimization for model fitting has been efficiently achieved through the Newton–Raphson method, which minimizes a twice differentiable forecast error function E between the true values $\mathbf{Y}$ and the forecasted values given by the model $\hat{\mathbf{Y}} = f(\mathbf{B}, \mathbf{Z})$ using a set of covariates $\mathbf{Z}$, including lags of $\mathbf{Y}$. $\mathbf{B}$ is the coefficient matrix, which is updated iteratively. The estimate for $\mathbf{B}$ at iteration k, represented by $\mathbf{B}^k$, is given by

$$\mathbf{B}^{k+1} = \mathbf{B}^k - (\nabla^2 E(\mathbf{B}^k))^{-1} \nabla E(\mathbf{B}^k), \tag{30}$$

where ∇E and $\nabla^2 E$ are the gradient and Hessian of E, respectively. With certain properties, convergence to a certain global minima can be guaranteed (Nocedal & Wright, 2006).

In order to enable distributed optimization, ∇E and $\nabla^2 E$ must be decomposable over multiple data owners. That is, these functions can be rewritten as the sum of functions that depend exclusively on local data from each data owner. Slavkovic et al. (2007) proposed a secure logistic regression approach for data split by records and features by using secure multi-party computation protocols during iterations of the Newton–Raphson method. Although distributed computing is feasible, there is no sufficient guarantee of data privacy, because it is an iterative process. While a single iteration cannot reveal private information, sufficient iterations can: in a logistic regression with data split by features, for each iteration k, the data owners exchange the matrix $\mathbf{Z}_{A_i}\mathbf{B}^k_{A_i}$, making it possible to recover the local data $\mathbf{Z}_{A_i}$ after enough iterations (Fienberg et al., 2009).

An example of an earlier promising work that combined logistic regression with the Newton–Raphson method for data distributed by records was the Grid binary LOgistic REgression (GLORE) framework (Wu et al., 2012). The GLORE model is based on model sharing rather than patient-level data, and it has motivated subsequent improvements. Some of these continue to suffer from confidentiality breaches on intermediate results, and others

resort to protocols for matrix addition and multiplication. Later, Li et al. (2015) explored the issue concerning the Newton–Raphson method over data distributed by features by considering a server that receives the transformed data and computes the intermediate results, returning them to each data owner. In order to avoid disclosing local data while obtaining an accurate global solution, the authors applied the kernel trick to obtain the global linear matrix, computed using dot products of local records ($\mathbf{Z}_{A_i}\mathbf{Z}_{A_i}{}^\top$), which can be used to solve the dual problem for logistic regression. However, they identified a technical challenge from scaling up the model with a large sample size, since each record requires a parameter.

2.3.3. Gradient-descent methods

Different gradient-descent methods have also been explored, aiming to minimize a forecast error function E between the true values $\mathbf{Y}$ and the forecasted values given by the model $\hat{\mathbf{Y}} = f(\mathbf{B}, \mathbf{Z})$ using a set of covariates $\mathbf{Z}$, including lags of $\mathbf{Y}$. The coefficient matrix $\mathbf{B}$ is updated iteratively such that the estimate at iteration k, $\mathbf{B}^k$, is given by

$$\mathbf{B}^k = \mathbf{B}^{k-1} + \eta \nabla E(\mathbf{B}^{k-1}), \tag{31}$$

where η is the learning rate. This allows for parallel computations when the optimization function E is decomposable. A common error function is the multivariate least squared error:

$$E(\mathbf{B}) = \frac{1}{2}\|\mathbf{Y} - f(\mathbf{B}, \mathbf{Z})\|^2. \tag{32}$$

With certain properties, convergence to a certain global minima can be guaranteed (Nesterov, 1998): (i) E is convex, (ii) ∇E is Lipschitz-continuous with constant L, i.e., for any $\mathbf{F}$, $\mathbf{G}$,

$$\|\nabla E(\mathbf{F}) - \nabla E(\mathbf{G})\|^2 \leq L\|\mathbf{F} - \mathbf{G}\|^2, \tag{33}$$

and (iii) $\eta \leq 1/L$.

Han et al. (2010) proposed a privacy-preserving linear regression technique for data distributed over features (with shared $\mathbf{Y}$) by combining distributed gradient descent with secure protocols, based on pre- or post-multiplication of the data by random private matrices. Song et al. (2013) introduced differential privacy by adding random noise $\mathbf{W}$ in the $\mathbf{B}$ updates:

$$\mathbf{B}^k = \mathbf{B}^{k-1} + \eta\Big(\nabla E(\mathbf{B}^{k-1}) + \mathbf{W}\Big). \tag{34}$$

When this iterative process uses a few randomly selected samples (or even a single sample), rather than the entire data, the process is known as *stochastic* gradient descent (SGD). The authors argued that the trade-off between performance and privacy is most pronounced when smaller batches are used.

3. Collaborative forecasting with VAR: Privacy analysis

This section presents a privacy analysis of collaborative forecasting with the VAR model, a model for the analysis of multivariate time series. The VAR model is not only used for forecasting tasks in different domains (and with significant improvements over univariate autoregressive models), but also for structural inference, where the main objective is to explore certain assumptions about the causal structure of the data (Toda & Phillips, 1993). A variant with LASSO regularization is also covered. We critically evaluate the methods described in Section 2 from a mathematical and numerical point of view in Section 3.3. The solar energy time series dataset and R scripts are published in an online supplement to this paper (see Appendix B).

3.1. VAR model formulation

Let $\{\mathbf{y}_t\}_{t=1}^T$ be an n-dimensional multivariate time series, where n is the number of data owners. Then, $\{\mathbf{y}_t\}_{t=1}^T$ follows a VAR model with p lags, represented as VAR$_n(p)$, when the following relationship holds:

$$\mathbf{y}_t = \boldsymbol{\eta} + \sum_{\ell=1}^{p} \mathbf{y}_{t-\ell}\mathbf{B}^{(\ell)} + \boldsymbol{\varepsilon}_t, \tag{35}$$

for $t = 1, \ldots, T$, where $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_n]$ is the constant intercept (row) vector, $\boldsymbol{\eta} \in \mathbb{R}^n$; $\mathbf{B}^{(\ell)}$ represents the coefficient matrix at lag $\ell = 1, \ldots, p$, $\mathbf{B}^{(\ell)} \in \mathbb{R}^{n\times n}$, and the coefficient associated with lag ℓ of time series i (to estimate time series j) is positioned at (i, j) of $\mathbf{B}^{(\ell)}$, for $i, j = 1, \ldots, n$; $\boldsymbol{\varepsilon}_t = [\varepsilon_{1,t}, \ldots, \varepsilon_{n,t}]$; and $\boldsymbol{\varepsilon}_t \in \mathbb{R}^n$ indicates a white noise vector that is independent and identically distributed with mean zero and nonsingular covariance matrix. By simplification, $\mathbf{y}_t$ is assumed to follow a centered process, $\boldsymbol{\eta} = \mathbf{0}$, i.e., as a vector of zeros of appropriate dimensions. A compact representation of a VAR$_n(p)$ model reads as follows:

$$\mathbf{Y} = \mathbf{Z}\mathbf{B} + \mathbf{E}, \tag{36}$$

where

$$\mathbf{Y} = \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_T \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}^{(1)} \\ \vdots \\ \mathbf{B}^{(p)} \end{bmatrix},$$

$$\mathbf{Z} = \begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_T \end{bmatrix}, \text{ and } \mathbf{E} = \begin{bmatrix} \boldsymbol{\varepsilon}_1 \\ \vdots \\ \boldsymbol{\varepsilon}_T \end{bmatrix},$$

are obtained by joining the vectors row-wise, and respectively define the $T \times n$ response matrix, the $np \times n$ coefficient matrix, the $T \times np$ covariate matrix, and the $T \times n$ error matrix, with $\mathbf{z}_t = [\mathbf{y}_{t-1}, \ldots, \mathbf{y}_{t-p}]$.

Notice that the VAR formulation adopted in this paper is not the usual $\mathbf{Y}^\top = \mathbf{B}^\top\mathbf{Z}^\top + \mathbf{E}^\top$, because a large proportion of the literature on privacy-preserving techniques derives from the standard linear regression problem, in which each row is a record and each column is a feature.

Notwithstanding the high potential of the VAR model for collaborative forecasting, namely by linearly combining time series from different data owners, data privacy or confidentiality issues might hinder this approach. For instance, renewable energy companies, competing in the same electricity market, will never share their electrical energy production data, even if this leads to a forecast error improvement in all individual forecasts.

For classical linear regression models, there are several techniques for estimating coefficients without sharing private information. However, in the VAR model, the data are divided by features (Fig. 2) and the variables to be forecasted are also covariates. This is challenging for privacy-preserving techniques (especially because it is also necessary to protect the data matrix $\mathbf{Y}$, as illustrated in Fig. 3). In what follows, when defining a VAR model, $\mathbf{Y}_{A_i} \in \mathbb{R}^{T\times 1}$ and $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times p}$ respectively denote the target and covariate matrix for the ith data owner. Therefore, the covariates and target matrices are obtained by joining the individual matrices column-wise, i.e., $\mathbf{Z} = [\mathbf{Z}_{A_1}, \ldots, \mathbf{Z}_{A_n}]$ and $\mathbf{Y} = [\mathbf{Y}_{A_1}, \ldots, \mathbf{Y}_{A_n}]$. For distributed computation, the coefficient matrix of data owner i is denoted by $\mathbf{B}_{A_i} \in \mathbb{R}^{p\times n}$, $i = 1, \ldots, n$.

3.2. Estimation in VAR models

Commonly, when the number of covariates included, np, is substantially smaller than the length of the time series, T, the VAR model can be fitted using a multivariate least squares solution, given by

$$\hat{\mathbf{B}}_{\text{LS}} = \underset{\mathbf{B}}{\operatorname{argmin}} \left(\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2 \right), \tag{37}$$

where $\|.\|_r$ represents both vector and matrix L_r norms. However, in collaborative forecasting, as the number of data owners increases, as well as the number of lags, it becomes crucial to use regularization techniques such as LASSO to introduce sparsity into the coefficient matrix estimated by the model. In the standard LASSO-VAR approach (see Nicholson et al. (2017) for different variants of LASSO regularization in the VAR model), the coefficients are given by

$$\hat{\mathbf{B}} = \underset{\mathbf{B}}{\operatorname{argmin}} \left(\frac{1}{2}\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2 + \lambda\|\mathbf{B}\|_1 \right), \tag{38}$$

where $\lambda > 0$ is a scalar penalty parameter.

With the addition of the LASSO regularization term, the convex objective function in (38) becomes non-differentiable, limiting the variety of optimization techniques that can be employed. In this domain, the ADMM (which was described in Section 2.3.1) is a widespread and computationally efficient technique that enables parallel estimations for data divided by features. The ADMM formulation of the non-differentiable cost function associated with the LASSO-VAR model in (38) solves the following optimization problem:

$$\min_{\mathbf{B},\mathbf{H}} \left(\frac{1}{2}\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2 + \lambda\|\mathbf{H}\|_1 \right) \text{ subject to } \mathbf{H} = \mathbf{B}, \tag{39}$$

which differs from (38) by splitting $\mathbf{B}$ into two parts ($\mathbf{B}$ and $\mathbf{H}$). Thus, the objective function can be split in two distinct objective functions, $f(\mathbf{B}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2$ and $g(\mathbf{H}) = \lambda\|\mathbf{H}\|_1$. The augmented Lagrangian (Boyd et al., 2011) of this problem is

$$L_\rho(\mathbf{B}, \mathbf{H}, \mathbf{W}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2 + \lambda\|\mathbf{H}\|_1 + \mathbf{W}^\top(\mathbf{B} - \mathbf{H}) + \frac{\rho}{2}\|\mathbf{B} - \mathbf{H}\|_2^2, \tag{40}$$

where $\mathbf{W}$ is the dual variable and $\rho > 0$ is the penalty parameter. The scaled form of this Lagrangian is

$$L_\rho(\mathbf{B}, \mathbf{H}, \mathbf{U}) = \frac{1}{2}\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2 + \lambda\|\mathbf{H}\|_1 + \frac{\rho}{2}\|\mathbf{B} - \mathbf{H} + \mathbf{U}\|^2 - \frac{\rho}{2}\|\mathbf{U}\|^2, \tag{41}$$

where $\mathbf{U} = (1/\rho)\mathbf{W}$ is the scaled dual variable associated with the constraint $\mathbf{B} = \mathbf{H}$. Hence, according to (27), the ADMM formulation for LASSO-VAR consists in the following iterations (Cavalcante et al., 2017):

$$\begin{cases} \mathbf{B}^{k+1} := \underset{\mathbf{B}}{\operatorname{argmin}} \left(\frac{1}{2}\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2 + \frac{\rho}{2}\|\mathbf{B} - \mathbf{H}^k + \mathbf{U}^k\|_2^2 \right) \\ \mathbf{H}^{k+1} := \underset{\mathbf{H}}{\operatorname{argmin}} \left(\lambda\|\mathbf{H}\|_1 + \frac{\rho}{2}\|\mathbf{B}^{k+1} - \mathbf{H} + \mathbf{U}^k\|_2^2 \right) \\ \mathbf{U}^{k+1} := \mathbf{U}^k + \mathbf{B}^{k+1} - \mathbf{H}^{k+1}. \end{cases} \tag{42}$$

Concerning the LASSO-VAR model, and since data are naturally divided by features (i.e., $\mathbf{Y} = [\mathbf{Y}_{A_1}, \ldots, \mathbf{Y}_{A_n}]$, $\mathbf{Z} = [\mathbf{Z}_{A_1}, \ldots, \mathbf{Z}_{A_n}]$ and $\mathbf{B} = [\mathbf{B}_{A_1}^\top, \ldots, \mathbf{B}_{A_n}^\top]^\top$) and the functions $\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2$ and $\|\mathbf{B}\|_1$ are decomposable (i.e., $\|\mathbf{Y} - \mathbf{Z}\mathbf{B}\|_2^2 = \|\mathbf{Y} - \sum_{i=1}^n \mathbf{Z}_{A_i}\mathbf{B}_{A_i}\|_2^2$ and $\|\mathbf{B}\|_1 = \sum_{i=1}^n \|\mathbf{B}_{A_i}\|_1$), the model fitting problem (38) becomes the following:

$$\underset{\Gamma}{\operatorname{argmin}} \left(\frac{1}{2}\|\mathbf{Y} - \sum_{i=1}^n \mathbf{Z}_{A_i}\mathbf{B}_{A_i}\|_2^2 + \lambda \sum_{i=1}^n \|\mathbf{B}_{A_i}\|_1 \right), \tag{43}$$

$\Gamma = \{\mathbf{B}_{A_1}, \ldots, \mathbf{B}_{A_n}\}$, which is rewritten as

$$\underset{\Gamma'}{\operatorname{argmin}} \left(\frac{1}{2}\|\mathbf{Y} - \sum_{i=1}^n \mathbf{H}_{A_i}\|_2^2 + \lambda \sum_{i=1}^n \|\mathbf{B}_{A_i}\|_1 \right) \quad \text{s.t. } \mathbf{B}_{A_1}\mathbf{Z}_{A_1} = \mathbf{H}_{A_1}, \ \ldots, \ \mathbf{B}_{A_n}\mathbf{Z}_{A_n} = \mathbf{H}_{A_n}, \tag{44}$$

$\Gamma' = \{\mathbf{B}_{A_1}, \ldots, \mathbf{B}_{A_n}, \mathbf{H}_{A_1}, \ldots, \mathbf{H}_{A_n}\}$, while the corresponding distributed ADMM formulation (Boyd et al., 2011; Cavalcante et al., 2017) is the one presented in the system of Eqs. (45),

$$\mathbf{B}_{A_i}^{k+1} = \underset{\mathbf{B}_{A_i}}{\operatorname{argmin}} \left(\frac{\rho}{2}\|\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^k + \overline{\mathbf{H}}^k - \overline{\mathbf{Z}\mathbf{B}}^k - \mathbf{U}^k - \mathbf{Z}_{A_i}\mathbf{B}_{A_i}\|_2^2 + \lambda\|\mathbf{B}_{A_i}\|_1 \right), \tag{45a}$$

$$\overline{\mathbf{H}}^{k+1} = \frac{1}{N+\rho}\left(\mathbf{Y} + \rho\overline{\mathbf{Z}\mathbf{B}}^{k+1} + \rho\mathbf{U}^k \right), \tag{45b}$$

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \overline{\mathbf{Z}\mathbf{B}}^{k+1} - \overline{\mathbf{H}}^{k+1}, \tag{45c}$$

where $\overline{\mathbf{Z}\mathbf{B}}^{k+1} = \frac{1}{n}\sum_{j=1}^n \mathbf{Z}_{A_j}\mathbf{B}_{A_j}^{k+1}$ and $\mathbf{B}_{A_i}^{k+1} \in \mathbb{R}^{p\times n}$, $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times p}$, $\mathbf{Y} \in \mathbb{R}^{T\times n}$, $\overline{\mathbf{H}}^k, \mathbf{U} \in \mathbb{R}^{T\times n}$, $i = 1, \ldots, n$.

Although parallel computation is an appealing property for the design of a privacy-preserving approach, the ADMM is an iterative optimization process that requires intermediate calculations. Thus, careful analysis is needed to determine whether a confidentiality breach will occur after enough iterations.

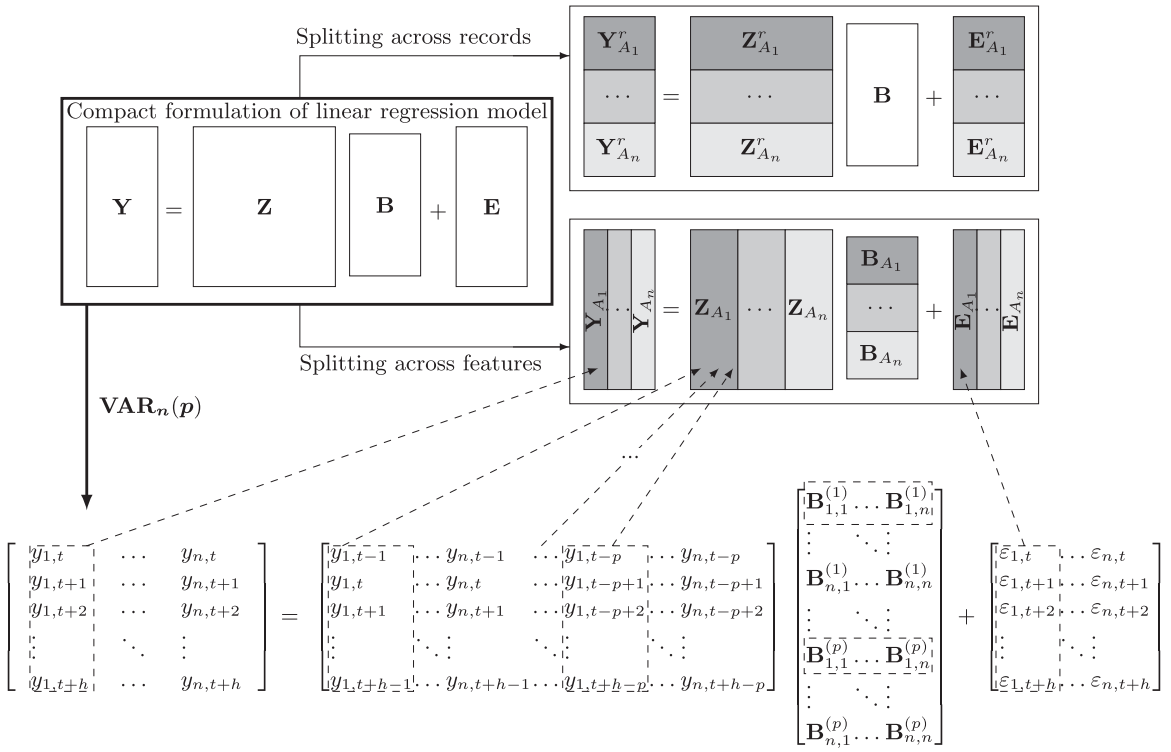


Fig. 2. Common data division structures and VAR model.

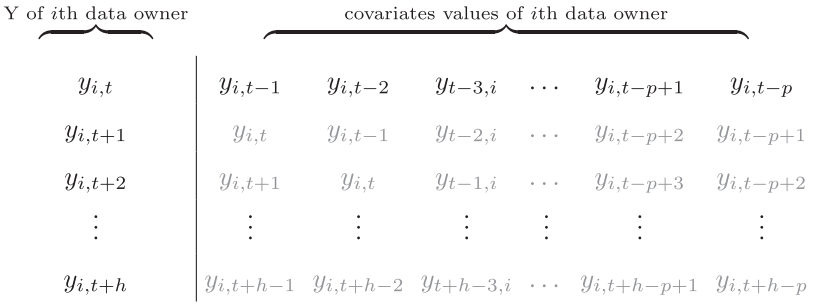


Fig. 3. Illustration of the data used by the ith data owner when fitting a VAR model.

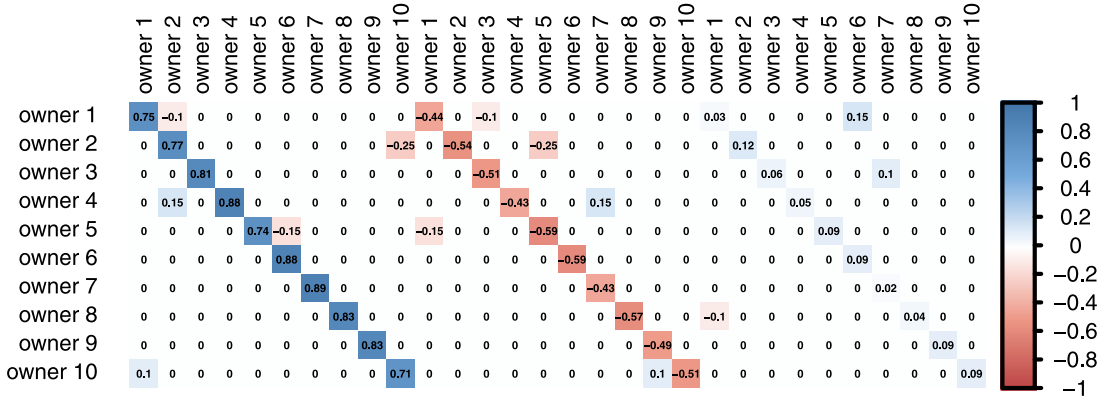


Fig. 4. Transpose of the coefficient matrix used to generate the VAR with ten data owners and three lags.

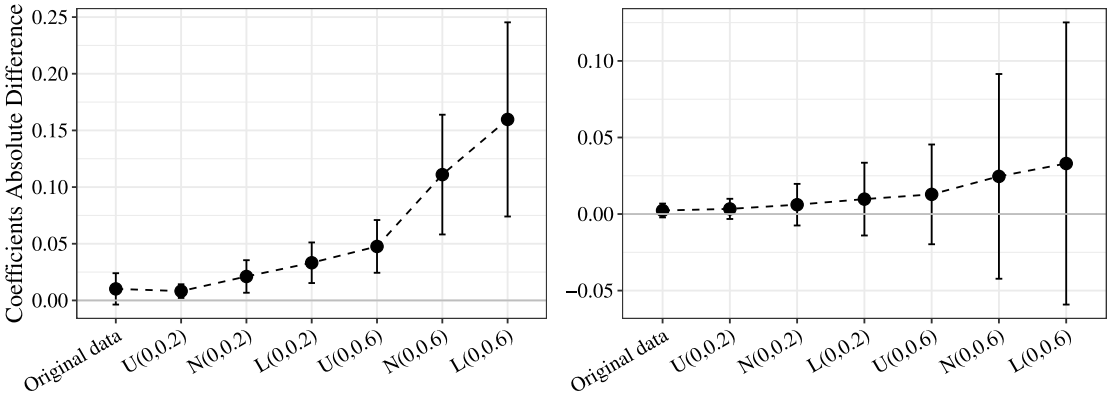


Fig. 5. Mean ± standard deviation for the absolute difference between the real and estimated coefficients (left: VAR with two data owners, right: VAR with ten data owners).

3.3. Privacy analysis

3.3.1. Data transformation with noise addition

This section presents experiments with simulated data and solar energy data collected from a smart grid pilot in Portugal. The objective was to quantify the impact of data distortion (through noise addition) on the model forecasting skill.

a) Synthetic Data: An experiment was performed to add random noise from a Gaussian distribution with zero mean and variance b^2, a Laplace distribution with zero mean and scale parameter b, and a uniform distribution with support $[-b, b]$—represented by $\mathcal{N}(0, b^2)$, $\mathcal{L}(0, b)$, and $\mathcal{U}(-b, b)$, respectively. Synthetic data generated by VAR processes were used to measure the differences between the coefficients' values when adding noise to the data. The simplest case considered a VAR with two data owners and two lags, described by

$$\begin{pmatrix} y_{1,t} & y_{2,t} \end{pmatrix} = \begin{pmatrix} y_{1,t-1} & y_{2,t-1} & y_{1,t-2} & y_{2,t-2} \end{pmatrix} \times \begin{pmatrix} 0.5 & 0.3 \\ 0.3 & 0.75 \\ -0.3 & -0.05 \\ -0.1 & -0.4 \end{pmatrix} + \begin{pmatrix} \varepsilon_{1,t} & \varepsilon_{2,t} \end{pmatrix}.$$

The second case included ten data owners and three lags and introduced a high percentage of null coefficients ($\approx$ 86%). Fig. 4 illustrates the considered coefficients. Since a specific configuration can generate various distinct trajectories, 100 simulations were performed for each specified VAR model, with 20,000 timestamps each. For both simulated datasets, the errors $\boldsymbol{\varepsilon}_t$ were assumed to follow a multivariate normal distribution with a zero mean vector and a covariance matrix equal to the identity matrix of appropriate dimensions. A distributed ADMM (detailed in Section 2.3.1) was used to estimate the LASSO-VAR coefficients, considering two different noise characterizations, $b \in \{0.2, 0.6\}$.

Fig. 5 summarizes the mean and the standard deviation of the absolute difference between the real and estimated coefficients for both VAR processes from the 100 simulations. The greater the noise b, the greater the distortion of the estimated coefficients. Moreover, the Laplace distribution, which has desirable properties to make data private according to a differential privacy framework, registered the greater distortion in the estimated model.

Using the original data, the ADMM solution tended to stabilize after 50 iterations, and the value of the coefficients was correctly estimated (the difference was approximately zero). The distorted time series converged faster, but the coefficients deviated from the real ones. In fact, adding noise contributed to decreasing the absolute value of the coefficients. That is, the relationships between the time series weakened.

These experiments allow us to draw conclusions about the use of differential privacy. The Laplace distribution has advantageous properties, since it ensures ε-differential privacy when random noise follows $\mathcal{L}(0, \frac{\Delta f_1}{\varepsilon})$. For the VAR with two data owners, $\Delta f_1 \approx 12$, since the observed values were in the interval $[-6, 6]$. Therefore, $\varepsilon = 20$ when $\mathcal{L}(0, 0.6)$ and $\varepsilon = 15$ when $\mathcal{L}(0, 0.8)$, meaning that the data still encompassed much relevant information. Finally, we verified the impact of noise addition on forecasting performance. Fig. 6 illustrates the improvement of each estimated $\text{VAR}_2(2)$ model (with and without noise addition) over the autoregressive (AR) model estimated with the original time series, in which collaboration was not used. This improvement was measured in terms of the mean absolute error (MAE) and root mean squared error (RMSE). In the case of ten data owners and when using data without noise, seven data owners improved their forecasting performance, which was expected from the coefficient matrix in Fig. 4. When Laplacian noise was applied to the data, only one data owner (the first one) improved its forecasting skill (when compared to the AR model) by using the estimated VAR model. Even though the masked data continued to provide relevant information, the model obtained for the Laplacian noise performed worse than the AR model for the second data owner, making the VAR useless for the majority of the data owners.

However, these results cannot be generalized for all VAR models, especially regarding the illustrated $\text{VAR}_{10}(3)$, which is very close to the AR(3) model. Given that, we conducted a third experiment, in which 200 random coefficient matrices were generated for a stationary $\text{VAR}_2(2)$

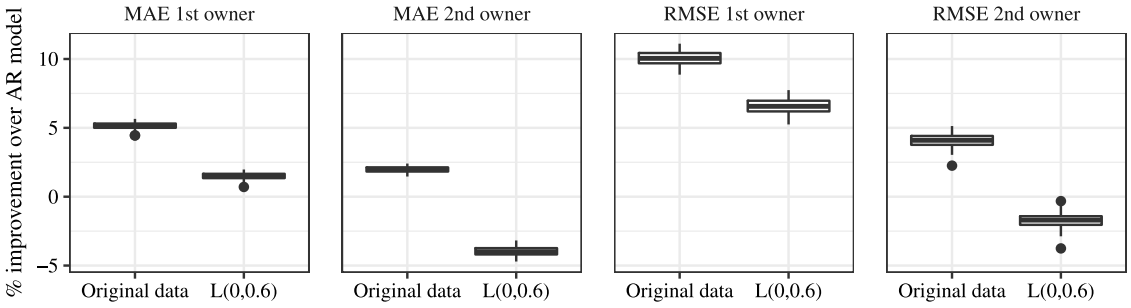
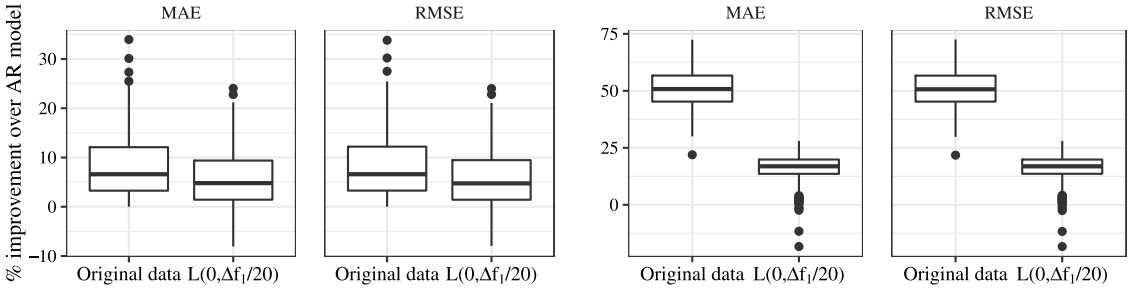


Fig. 6. Improvement (%) of $VAR_2(2)$ model over AR(2) model in terms of MAE and RMSE for synthetic data.



(a) $VAR_2(2)$ model. (b) $VAR_{10}(3)$ model.

Fig. 7. Improvement (%) of VAR model over AR model in terms of MAE and RMSE for synthetic data.

and $VAR_{10}(3)$ following the algorithm proposed by Ansley and Kohn (1986). Usually, the generated coefficient matrix has no null entries and the higher values are not necessarily found on diagonals. Fig. 7 illustrates the improvement for each data owner when using a VAR model (with and without noise addition) over the AR model. In this case, the percentage of times the AR model performed better than the VAR model with distorted data was smaller, but the degradation of the models was still noticeable, especially in the case with ten data owners.

b) Real Data: We also used a real dataset comprising hourly time series of solar power generation from 44 micro-generation units located in Évora city (Portugal), covering the period from February 1, 2011 to March 6, 2013. As in Cavalcante and Bessa (2017), records corresponding to a solar zenith angle higher than 90° were removed, in order to exclude nighttime hours (i.e., hours without any generation). To make the time series stationary, a normalization of the solar power was applied by using a clear-sky model (see Bacher et al. (2009)) that gives an estimate of solar power under clear sky conditions at any given time. The power generation for the next hour was modeled through the VAR model, which combined data from the 44 data owners and considered three non-consecutive lags (1 h, 2 h, and 24 h). Fig. 8(a) summarizes the improvement for the 44 solar power plants over the autoregressive model, in terms of the MAE and RMSE. The quartile 25% shows that the MAE improved by at least 10% for 33 of the 44 solar power plants, when the data owners shared their observed data. The improvement to the RMSE was not as significant, but was still greater than zero. Although the data obtained after adding Laplacian noise retained its temporal dependency, as illustrated in Fig. 8(b), the corresponding VAR model was useless for 4 of the 44 data owners. When considering the RMSE, 2 of the 44 data owners obtained better results by using an autoregressive model. Once again, the resulting model suffered a significant reduction in forecasting capability.

3.3.2. Linear algebra-based protocols

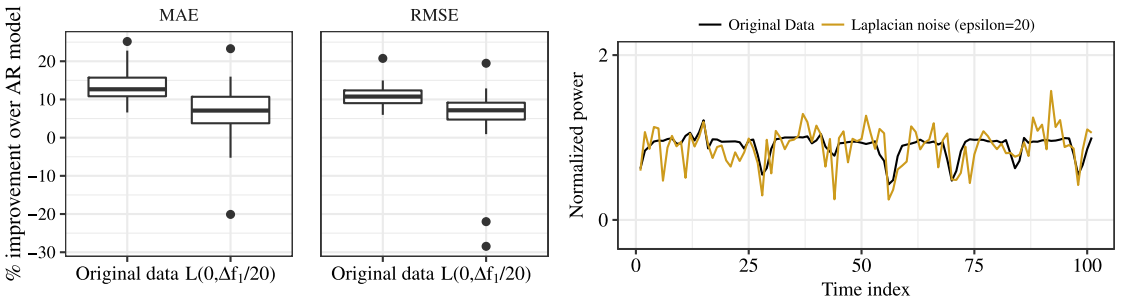
Let us consider a case with two data owners. Since the multivariate least squares estimate for the VAR model with covariates $\mathbf{Z} = [\mathbf{Z}_{A_1}, \mathbf{Z}_{A_2}]$ and target $\mathbf{Y} = [\mathbf{Y}_{A_1}, \mathbf{Y}_{A_2}]$ is

$$\hat{\mathbf{B}}_{LS} = \left(\begin{bmatrix} \mathbf{Z}_{A_1}^\top \\ \mathbf{Z}_{A_2}^\top \end{bmatrix} [\mathbf{Z}_{A_1}, \mathbf{Z}_{A_2}] \right)^{-1} \left(\begin{bmatrix} \mathbf{Z}_{A_1}^\top \\ \mathbf{Z}_{A_2}^\top \end{bmatrix} [\mathbf{Y}_{A_1}, \mathbf{Y}_{A_2}] \right) \quad (46)$$

$$= \begin{pmatrix} \mathbf{Z}_{A_1}^\top \mathbf{Z}_{A_1} & \mathbf{Z}_{A_1}^\top \mathbf{Z}_{A_2} \\ \mathbf{Z}_{A_2}^\top \mathbf{Z}_{A_1} & \mathbf{Z}_{A_2}^\top \mathbf{Z}_{A_2} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{Z}_{A_1}^\top \mathbf{Y}_{A_1} & \mathbf{Z}_{A_1}^\top \mathbf{Y}_{A_2} \\ \mathbf{Z}_{A_2}^\top \mathbf{Y}_{A_1} & \mathbf{Z}_{A_2}^\top \mathbf{Y}_{A_2} \end{pmatrix}, \quad (47)$$

the data owners need to jointly compute $\mathbf{Z}_{A_1}^\top \mathbf{Z}_{A_2}$, $\mathbf{Z}_{A_1}^\top \mathbf{Y}_{A_2}$ and $\mathbf{Z}_{A_2}^\top \mathbf{Y}_{A_1}$.

As mentioned in the introduction to Section 2.2.1, Du et al. (2004) proposed protocols for secure matrix multiplication for situations where two data owners observe the same common target matrix and different confidential covariates. Unfortunately, without assuming a trusted third entity for generating random matrices, the proposed protocol fails when applied to the VAR model. This is because $2(T-1)p$ values of the covariate matrix $\mathbf{Z} \in \mathbb{R}^{T \times 2p}$ are included in the target matrix $\mathbf{Y} \in \mathbb{R}^{T \times 2}$, which is also



(a) Improvement (%) of VAR_{44} model over AR model, in terms of MAE and RMSE.

(b) Example of the normalized time series.

Fig. 8. Results of a real case-study with solar power time series.

undisclosed. Additionally, $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times p}$ has $T+p-1$ unique values instead of Tp—regarding which, see Fig. 3.

Proposition 1. *Consider a case in which two data owners with private data $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times p}$ and $\mathbf{Y}_{A_i} \in \mathbb{R}^{T\times 1}$ want to estimate a VAR model without trusting a third entity, $i = 1, 2$. Assume that the T records are consecutive, as well as the p lags. The multivariate least squares estimate for the VAR model with covariates $\mathbf{Z} = [\mathbf{Z}_{A_1}, \mathbf{Z}_{A_2}]$ and target $\mathbf{Y} = [\mathbf{Y}_{A_1}, \mathbf{Y}_{A_2}]$ requires the computation of $\mathbf{Z}_{A_1}^\top \mathbf{Z}_{A_2}$, $\mathbf{Z}_{A_1}^\top \mathbf{Y}_{A_2}$ and $\mathbf{Z}_{A_2}^\top \mathbf{Y}_{A_1}$.*

If the data owners use the protocol proposed by Du et al. (2004) for computing such matrices, then the information exchanged can be used to recover the data matrices.

Proof. As in Du et al. (2004), let us consider a case with two data owners without a third entity generating random matrices.

In order to compute $\mathbf{Z}_{A_1}^\top \mathbf{Z}_{A_2}$ both data owners define a matrix $\mathbf{M} \in \mathbb{R}^{T\times T}$ and compute its inverse $\mathbf{M}^{-1}$. Then, the protocol stipulates that

$$\begin{aligned}\mathbf{Z}_{A_1}^\top \mathbf{Z}_{A_2} &= \mathbf{Z}_{A_1}^\top \mathbf{M}\mathbf{M}^{-1}\mathbf{Z}_{A_2} \\ &= \mathbf{A}[\mathbf{M}_{\text{left}}, \mathbf{M}_{\text{right}}] \begin{bmatrix} (\mathbf{M}^{-1})_{\text{top}} \\ (\mathbf{M}^{-1})_{\text{bottom}} \end{bmatrix} \mathbf{Z}_{A_2} \\ &= \underbrace{\mathbf{Z}_{A_1}^\top \mathbf{M}_{\text{left}} (\mathbf{M}^{-1})_{\text{top}} \mathbf{Z}_{A_2}}_{\text{derived by Owner \#1}} \\ &\quad + \underbrace{\mathbf{Z}_{A_1}^\top \mathbf{M}_{\text{right}} (\mathbf{M}^{-1})_{\text{bottom}} \mathbf{Z}_{A_2}}_{\text{derived by Owner \#2}},\end{aligned}$$

requiring the data owners to share $\mathbf{Z}_{A_1}^\top \mathbf{M}_{\text{right}} \in \mathbb{R}^{p\times T/2}$ and $(\mathbf{M}^{-1})_{\text{top}}\mathbf{Z}_{A_2} \in \mathbb{R}^{T/2\times p}$, respectively. This implies that each data owner shares $pT/2$ values.

Similarly, the computation of $\mathbf{Z}_{A_1}^\top \mathbf{Y}_{A_2}$ implies that the data owners define a matrix $\mathbf{M}^*$, and share $\mathbf{Z}_{A_1}^\top \mathbf{M}^*_{\text{right}} \in \mathbb{R}^{p\times T/2}$ and $(\mathbf{M}^{*-1})_{\text{top}}\mathbf{Y}_{A_2} \in \mathbb{R}^{T/2\times p}$, respectively, providing new $pT/2$ values. This means that Owner #2 receives $\mathbf{Z}_{A_1}^\top \mathbf{M}_{\text{right}}$ and $\mathbf{Z}_{A_1}^\top \mathbf{M}^*_{\text{right}}$, i.e., Tp values, and may recover $\mathbf{Z}_{A_1}$, which consists of Tp values and represents a confidentiality breach. Furthermore, when considering a VAR model with p lags, $\mathbf{Z}_{A_1}$ has $T+p-1$ unique values, meaning there are fewer values to recover. Analogously, Owner #1 may recover $\mathbf{Z}_{A_2}$ through the matrices shared for the computation of $\mathbf{Z}_{A_1}^\top \mathbf{Z}_{A_2}$ and $\mathbf{Z}_{A_2}^\top \mathbf{Y}_{A_1}$.

Finally, when considering a VAR with p lags, $\mathbf{Y}_{A_i}$ only has p values that are not in $\mathbf{Z}_{A_i}$. While computing $\mathbf{Z}_{A_1}^\top \mathbf{Y}_{A_2}$, Owner #1 receives $T/2$ values from $(\mathbf{M}^{*-1})_{\text{top}}\mathbf{Y}_{A_2} \in \mathbb{R}^{T/2\times 1}$, such that a confidentiality breach can occur (in general $T/2 > p$). In the same way, Owner #2 recovers $\mathbf{Y}_{A_1}$ when computing $\mathbf{Z}_{A_2}^\top \mathbf{Y}_{A_1}$. □

The main disadvantage of linear algebra-based methods is that they do not take into account that, in the VAR model, both target variables and covariates are private, and that a large proportion of the covariates matrix is determined by knowing the target variables. This means that the data shared between data owners may be enough for competitors to be able to reconstruct the original data. For the method proposed by Karr et al. (2009), a consequence of such data is that the assumption $rank\left((\mathbf{I}-\mathbf{W}\mathbf{W}^\top)\mathbf{C}\right) = m-g$ may still provide a sufficient number of linearly independent equations on the other data owner's data to recovering the latter's data.

3.3.3. ADMM and central node

Zhang and Wang (2018) offered a promising approach to dealing with the problem of private data during the ADMM iterative process described by (45). According to their approach, for each iteration k, each data owner i communicates local results, $\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^{k+1}$, to the central node, $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times p}$, $\mathbf{B}_{A_i}^{k+1} \in \mathbb{R}^{p\times n}$, $i = 1, \ldots, n$. Then, the central node computes the intermediate matrices in (45b)–(45c) and returns the matrix $\overline{\mathbf{H}}^k - \overline{\mathbf{Z}\mathbf{B}}^k - \mathbf{U}^k$ to each data owner, in order to update $\mathbf{B}_{A_i}$ in the next iteration, as seen in (45a). Fig. 9 illustrates this method for the LASSO-VAR with three data owners. In this solution, there is no direct exchange of private data. However, as we explain next, not only can the central node recover the original data, but also the individual data owners can obtain a good estimation of the data used by their competitors.

Proposition 2. *In the most optimistic scenario, without repeated values in $\mathbf{Y}_{A_i} \in \mathbb{R}^{T\times 1}$ and $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times p}$, when applying the algorithm from Zhang et al. (2019) to solve the LASSO-VAR model in (45), the central agent can recover*

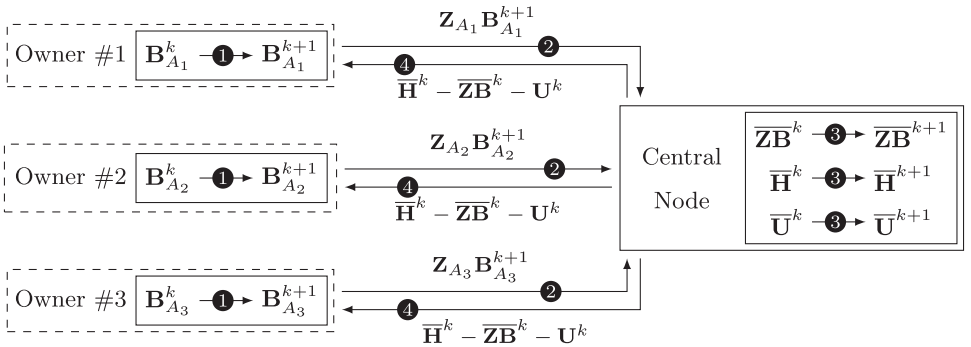


Fig. 9. Distributed ADMM LASSO-VAR with a central node and three data owners (related to the algorithm in (45)).

sensible data after

$$k = \left\lceil \frac{Tp}{Tn - pn} \right\rceil \tag{48}$$

iterations, where $\lceil x \rceil$ *denotes the ceiling function.*

Proof. Using the notation in Section 3.1, each of the n data owners is assumed to use the same number of lags p to fit a LASSO-VAR model with a total number of T records. (Importantly, $T > np$; otherwise more coefficients must be determined than system equations.) After k iterations, the central node receives a total of Tnk values from each data owner i, corresponding to $\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^1, \mathbf{Z}_{A_i}\mathbf{B}_{A_i}^2, \ldots, \mathbf{Z}_{A_i}\mathbf{B}_{A_i}^k \in \mathbb{R}^{T\times n}$, and does not know $pnk + Tp$, corresponding to $\mathbf{B}_{A_i}^1, \ldots, \mathbf{B}_{A_i}^k \in \mathbb{R}^{p\times n}$ and $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times p}$, respectively, $i = 1, \ldots, n$. Given that, the solution of the inequality

$$Tnk \geq pnk + Tp, \tag{49}$$

in k suggests that a confidentiality breach can occur after

$$k = \left\lceil \frac{Tp}{Tn - pn} \right\rceil \tag{50}$$

iterations, where $\lceil x \rceil$ denotes the ceiling function. Since T tends to be large, k tends to $\lceil p/n \rceil$, which may represent a confidentiality breach if the number of iterations required for the algorithm to converge is greater than $\lceil p/n \rceil$. □

Proposition 3. *In the most optimistic scenario, without repeated values in* $\mathbf{Y}_{A_i} \in \mathbb{R}^{T\times 1}$ *and* $\mathbf{Z}_{A_i} \in \mathbb{R}^{T\times p}$*, when applying the algorithm from Zhang et al. (2019) to solve the LASSO-VAR model in (45), the data owners can recover sensible data from competitors after*

$$k = \left\lceil \frac{Tn + (n-1)(Tp+T)}{Tn - (n-1)pn} \right\rceil \tag{51}$$

iterations.

Proof. Without loss of generality, Owner #1 is considered a semi-trusted data owner. (A semi-trusted data owner completes and shares his/her computations faithfully, but tries to learn additional information while or after the algorithm runs.) For each iteration k, this data owner receives the intermediate matrix $\overline{\mathbf{H}}^k - \underbrace{\overline{\mathbf{ZB}}^k}_{=\frac{1}{n}\sum_{i=1}^n \mathbf{Z}_{A_i}\mathbf{B}_{A_i}^k} - \mathbf{U}^k \in \mathbb{R}^{T\times n}$, which provides Tn values. However, Owner #1 does not know

$$\underbrace{-\mathbf{U}^k + \overline{\mathbf{H}}^k}_{\in\mathbb{R}^{T\times n}}, \ \underbrace{\mathbf{B}_{A_2}^k, \ldots, \mathbf{B}_{A_n}^k}_{n-1 \text{ matrices } \in\mathbb{R}^{p\times n}}, \ \underbrace{\mathbf{Z}_{A_2}, \ldots, \mathbf{Z}_{A_n}}_{n-1 \text{ matrices } \in\mathbb{R}^{T\times p}}, \ \underbrace{\mathbf{Y}_{A_2}, \ldots, \mathbf{Y}_{A_n}}_{n-1 \text{ matrices } \in\mathbb{R}^{T\times 1}},$$

which corresponds to $Tn + (n-1)pn + (n-1)Tp + (n-1)T$ values. Nevertheless, since all the data owners know that $\overline{\mathbf{H}}^k$ and $\mathbf{U}^k$ are defined by the expressions in (45b) and (45c), it is possible to perform some simplifications in which $\mathbf{U}^k$ and $\overline{\mathbf{H}}^k - \overline{\mathbf{ZB}}^k - \mathbf{U}^k$ become (52) and (53), respectively:

$$\begin{aligned}\mathbf{U}^k &\overset{(45c)}{=} \mathbf{U}^{k-1} + \overline{\mathbf{ZB}}^k - \overline{\mathbf{H}}^k = \mathbf{U}^{k-1} + \overline{\mathbf{ZB}}^k \\ &\quad - \underbrace{\frac{1}{N+\rho}\left(\mathbf{Y} + \rho\overline{\mathbf{ZB}}^k + \rho\mathbf{U}^{k-1}\right)}_{=\overline{\mathbf{H}}^k, \text{ according to (45b)}} \\ &= \left[1 - \frac{\rho}{N+\rho}\right]\mathbf{U}^{k-1} + \left[1 - \frac{\rho}{N+\rho}\right]\overline{\mathbf{ZB}}^k \\ &\quad - \frac{1}{N+\rho}\mathbf{Y},\end{aligned} \tag{52}$$

$$\overline{\mathbf{H}}^k - \overline{\mathbf{ZB}}^k - \mathbf{U}^k = \underbrace{\frac{1}{N+\rho}\left(\mathbf{Y} + \rho\overline{\mathbf{ZB}}^k + \rho\mathbf{U}^{k-1}\right)}_{=\overline{\mathbf{H}}^k, \text{ according to (45b)}} - \overline{\mathbf{ZB}}^k - \mathbf{U}^k. \tag{53}$$

Therefore, the iterative process of finding the competitors' data proceeds as follows:

1.) *Initialization:* The central node generates $\mathbf{U}^0 \in \mathbb{R}^{T\times n}$, and the ith data owner generates $\mathbf{B}_{A_i}^1 \in \mathbb{R}^{p\times n}$, $i = 1,\ldots,n$.
2.) *Iteration #1:* The central node receives $\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^1$ and computes $\mathbf{U}^1$, returning $\overline{\mathbf{H}}^1 - \overline{\mathbf{ZB}}^1 - \mathbf{U}^1 \in \mathbb{R}^{T\times n}$, which is returned for all n data owners. At this point, Owner #1 receives Tn values and does not know

$$\underbrace{\mathbf{U}^0}_{\in\mathbb{R}^{T\times n}},\ \underbrace{\mathbf{B}_{A_2}^1,\ldots,\mathbf{B}_{A_n}^1}_{n-1 \text{ matrices } \in\mathbb{R}^{p\times n}},\ \underbrace{\mathbf{Z}_{A_2},\ldots,\mathbf{Z}_{A_n}}_{n-1 \text{ matrices } \in\mathbb{R}^{T\times p}},$$

and $n-1$ columns of $\mathbf{Y} \in \mathbb{R}^{T\times n}$, corresponding to $Tn + (n-1)[pn + Tp + T]$ values.
3.) *Iteration #2:* The central node receives $\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^2$ and computes $\mathbf{U}^2$, returning $\overline{\mathbf{H}}^2 - \overline{\mathbf{ZB}}^2 - \mathbf{U}^2$ for the n data owners. At this point, only new estimations for the vectors $\mathbf{B}_{A_2},\ldots,\mathbf{B}_{A_n}$ were introduced to the system, which means that more $(n-1)pn$ values must be estimated.

As a result, after k iterations, Owner #1 has received $\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^1,\ldots,\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^k \in \mathbb{R}^{T\times n}$ corresponding to Tnk values and needs to estimate

$$\underbrace{\mathbf{U}^0}_{\in\mathbb{R}^{T\times n}},\underbrace{\mathbf{B}_{A_2}^1,\ldots,\mathbf{B}_{A_n}^1,\mathbf{B}_{A_2}^2,\ldots,\mathbf{B}_{A_n}^2,\ldots,\mathbf{B}_{A_2}^k,\ldots,\mathbf{B}_{A_n}^k}_{(n-1)k \text{ matrices } \in\mathbb{R}^{p\times n}},$$
$$\underbrace{\mathbf{Z}_{A_2},\ldots,\mathbf{Z}_{A_n}}_{n-1 \text{ matrices } \in\mathbb{R}^{T\times p}},$$

and $n-1$ columns of $\mathbf{Y} \in \mathbb{R}^{T\times n}$, corresponding to $Tn + (n-1)[kpn + Tp + T]$. Then, the solution for the inequality

$$Tnk \geq Tn + (n-1)[kpn + Tp + T], \tag{54}$$

suggests that a confidentiality breach may occur after

$$k = \left\lceil \frac{Tn + (n-1)(Tp+T)}{Tn - (n-1)pn} \right\rceil \tag{55}$$

iterations. □

Fig. 10 illustrates the k value for different combinations of T, n, and p. In general, the greater the number of records T, the smaller the number of iterations necessary for a confidentiality breach. This is because more information is shared during each iteration of the ADMM algorithm. By contrast, the number of iterations before a possible confidentiality breach increases with the number of data owners (n). The same is true for the number of lags (p).

3.3.4. ADMM and noise mechanisms

The target matrix $\mathbf{Y} = [\mathbf{Y}_{A_1},\ldots,\mathbf{Y}_{A_n}]$ corresponds to the sum of private matrices $\mathbf{I}_{\mathbf{Y}_{A_i}} \in \mathbb{R}^{T\times n}$. That is,

$$\underbrace{\begin{bmatrix} y_{1,t} & y_{2,t} & \cdots & y_{n,t} \\ y_{1,t+1} & y_{2,t+1} & \cdots & y_{n,t+1} \\ y_{1,t+2} & y_{2,t+2} & \cdots & y_{n,t+2} \\ \vdots & \ddots & \vdots & \\ y_{1,t+h} & y_{2,t+h} & \cdots & y_{n,t+h} \end{bmatrix}}_{\mathbf{Y}}$$
$$= \underbrace{\begin{bmatrix} y_{1,t} & 0 & \cdots & 0 \\ y_{1,t+1} & 0 & \cdots & 0 \\ y_{1,t+2} & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \\ y_{1,t+h} & 0 & \cdots & 0 \end{bmatrix}}_{\mathbf{I}_{Y_{A_1}}} + \underbrace{\begin{bmatrix} 0 & y_{2,t} & \cdots & 0 \\ 0 & y_{1,t+1} & \cdots & 0 \\ 0 & y_{1,t+2} & \cdots & 0 \\ \vdots & \ddots & \vdots & \\ 0 & y_{1,t+h} & \cdots & 0 \end{bmatrix}}_{\mathbf{I}_{Y_{A_2}}}$$
$$+ \cdots + \underbrace{\begin{bmatrix} 0 & 0 & \cdots & y_{n,t} \\ 0 & 0 & \cdots & y_{n,t+1} \\ 0 & 0 & \cdots & y_{n,t+2} \\ \vdots & \ddots & \vdots & \\ 0 & 0 & \cdots & y_{n,t+h} \end{bmatrix}}_{\mathbf{I}_{Y_{A_n}}}, \tag{56}$$

where $[\mathbf{I}_{\mathbf{Y}_{A_i}}]_{i,j}=[\mathbf{Y}]_{i,j}$ in cases where the entry (i, j) of $\mathbf{Y}$ is from ith data owner and $[\mathbf{I}_{\mathbf{Y}_{A_i}}]_{i,j}=0$ otherwise.

Since the LASSO-VAR ADMM formulation is provided by (45), at iteration k, the data owners receive the intermediate matrix $\overline{\mathbf{H}}^k - \overline{\mathbf{ZB}}^k - \mathbf{U}^k$ and then update their local solution through (45a). The combination of (52) with (56) can be used to rewrite $\mathbf{U}^k$ as

$$\mathbf{U}^k = \left[1 - \frac{\rho}{N+\rho}\right]\mathbf{U}^{k-1} + \sum_{i=1}^{n} \underbrace{\left[1 - \frac{\rho}{N+\rho}\right]\frac{1}{n}\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^k - \frac{1}{N+\rho}\mathbf{I}_{\mathbf{Y}_{A_i}}}_{\text{information from owner } i}, \tag{57}$$

and, similarly, $\overline{\mathbf{H}}^k - \overline{\mathbf{ZB}}^k$ can be rewritten as

$$\begin{aligned} \overline{\mathbf{H}}^k - \overline{\mathbf{ZB}}^k &= \frac{1}{N+\rho}\mathbf{Y} + \left[\frac{\rho}{N+\rho} - 1\right]\overline{\mathbf{ZB}}^k + \frac{\rho}{N+\rho}\mathbf{U}^{k-1} - \mathbf{U}^k \\ &= \sum_{i=1}^{n} \underbrace{\left(\frac{1}{N+\rho}\mathbf{I}_{\mathbf{Y}_{A_i}} + \left[\frac{\rho}{N+\rho} - 1\right]\frac{1}{n}\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^k\right)}_{\text{information from owner } i} \\ &\quad + \frac{\rho}{N+\rho}\mathbf{U}^{k-1} - \mathbf{U}^k, \end{aligned} \tag{58}$$

where

$$\mathbf{Y} = \sum_{i=1}^{n} \mathbf{I}_{\mathbf{Y}_{A_i}}, \tag{59}$$

$$\overline{\mathbf{ZB}}^{k+1} = \sum_{i=1}^{n} \frac{\rho}{n}\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^{k+1}. \tag{60}$$

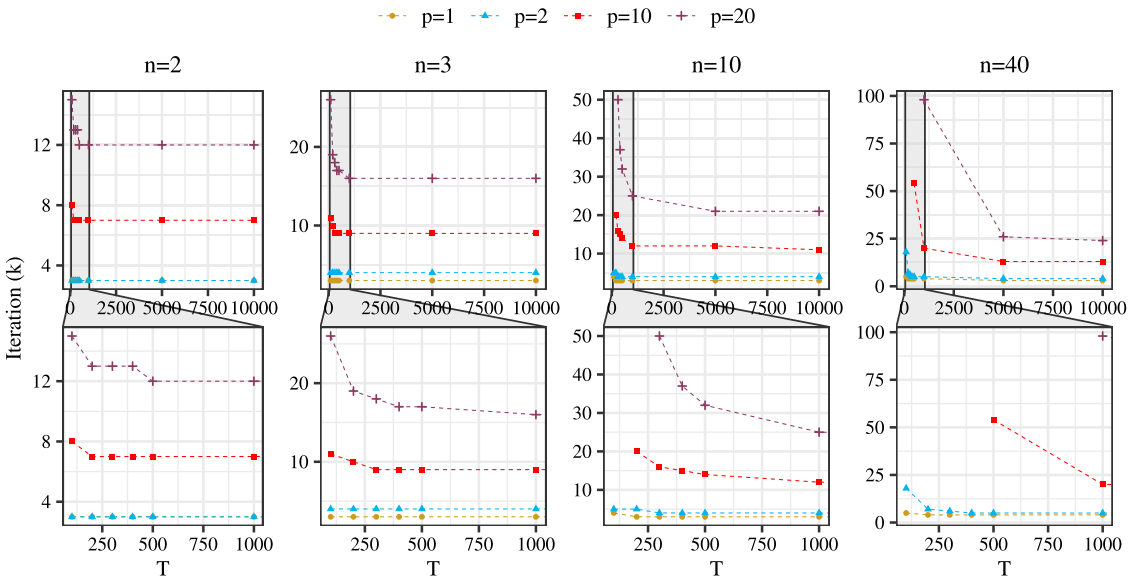


Fig. 10. Number of iterations until a possible confidentiality breach, considering the centralized ADMM-based algorithm in Zhang et al. (2019).

By analyzing (57) and (58), it is possible to verify that data owner i only needs to share

$$\frac{1}{N+\rho}\mathbf{I}_{\mathbf{Y}_{A_i}} + \left[\frac{\rho}{N+\rho} - 1\right]\frac{1}{n}\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^k, \tag{61}$$

for the computation of $\overline{\mathbf{H}}^k - \overline{\mathbf{Z}\mathbf{B}}^k - \mathbf{U}^k$.

Let $\mathbf{W}_{1,A_i} \in \mathbb{R}^{T\times n}$, $\mathbf{W}_{2,A_i} \in \mathbb{R}^{T\times p}$, $\mathbf{W}_{3,A_i} \in \mathbb{R}^{p\times n}$, $\mathbf{W}_{4,A_i} \in \mathbb{R}^{T\times n}$ represent noise matrices generated according to the differential privacy framework. The noise mechanism can be introduced by

(i) adding noise to the data itself, i.e., replacing $\mathbf{I}_{\mathbf{Y}_{A_i}}$ and $\mathbf{Z}_{A_i}$ by

$$\mathbf{I}_{\mathbf{Y}_{A_i}} + \mathbf{W}_{1,A_i} \text{ and } \mathbf{Z}_{A_i} + \mathbf{W}_{2,A_i}, \tag{62}$$

(ii) adding noise to the estimated coefficients, i.e., replacing $\mathbf{B}_{A_i}^k$ by

$$\mathbf{B}_{A_i}^k + \mathbf{W}_{3,A_i}, \tag{63}$$

(iii) adding noise to the intermediate matrix (61),

$$\frac{1}{N+\rho}\mathbf{I}_{\mathbf{Y}_{A_i}} + \left[\frac{\rho}{N+\rho} - 1\right]\frac{1}{n}\mathbf{Z}_{A_i}\mathbf{B}_{A_i}^k + \mathbf{W}_{4,A_i}. \tag{64}$$

The addition of noise directly to the data with (62) was empirically analyzed in Section 3.3.1. As we showed, confidentiality comes at the cost of deteriorating model accuracy. The question is whether adding noise to the coefficients or intermediate matrix can ensure that data are not recovered after a number of iterations.

Proposition 4. *Consider noise addition in an ADMM-based framework by*

(i) adding noise to the coefficients, as described in (63);
(ii) adding noise to the exchanged intermediate matrix, as described in (64).

In both cases, a semi-trusted data owner can recover the data after

$$k = \left\lceil \frac{Tn + (n-1)(Tp+T)}{Tn - (n-1)pn} \right\rceil \tag{65}$$

iterations.

Proof. These statements are promptly deduced from the Proof presented for Proposition 3. Without loss of generality, Owner #1 is considered a semi-trusted data owner.

(i) Owner #1 can estimate $\mathbf{B}_{A_i}$, without distinguishing between $\mathbf{B}_{A_i}$ and $\mathbf{W}_{3,A_i}$ in (63), by recovering $\mathbf{I}_{\mathbf{Y}_{A_i}}$ and $\mathbf{Z}_{A_i}$. Let $\mathbf{B}'_{A_i} = \mathbf{B}_{A_i} + \mathbf{W}_{3,A_i}$ and $\overline{\mathbf{H}'}^k$, $\mathbf{U}'^k$ be the matrices $\overline{\mathbf{H}}^k$, $\mathbf{U}^k$ replacing $\mathbf{B}_{A_i}$ by $\mathbf{B}'_{A_i}$. Then, at iteration k, Owner #1 receives $\overline{\mathbf{H}'}^k - \overline{\mathbf{Z}\mathbf{B}'}^k - \mathbf{U}'^k \in \mathbb{R}^{T\times n}$ (Tn values) and does not know

$$\underbrace{\overline{\mathbf{H}'}^k - \mathbf{U}'^k}_{\in\mathbb{R}^{T\times n}}, \ \underbrace{\mathbf{B}'^k_{A_2}, \ldots, \mathbf{B}'^k_{A_n}}_{n-1 \text{ matrices } \in\mathbb{R}^{p\times n}}, \ \underbrace{\mathbf{Z}_{A_2}, \ldots, \mathbf{Z}_{A_n}}_{n-1 \text{ matrices } \in\mathbb{R}^{T\times p}}, \ \underbrace{\mathbf{Y}_{A_2}, \ldots, \mathbf{Y}_{A_n}}_{n-1 \text{ matrices } \in\mathbb{R}^{T\times 1}},$$

which corresponds to $Tn + (n-1)pn + (n-1)Tp + (n-1)T$ values. As in Proposition 3, this means that, after k iterations, Owner #1 has received Tnk values and needs to estimate

$$\underbrace{\mathbf{U}'^0}_{\in\mathbb{R}^{T\times n}}, \underbrace{\mathbf{B}'^1_{A_2}, \ldots, \mathbf{B}'^1_{A_n}, \mathbf{B}'^2_{A_2}, \ldots, \mathbf{B}'^2_{A_n}, \ldots, \mathbf{B}'^k_{A_2}, \ldots, \mathbf{B}'^k_{A_n}}_{(n-1)k \text{ matrices } \in\mathbb{R}^{p\times n}}, \underbrace{\mathbf{Z}_{A_2}, \ldots, \mathbf{Z}_{A_n}}_{n-1 \text{ matrices } \in\mathbb{R}^{T\times p}},$$

and $n-1$ columns of $\mathbf{Y} \in \mathbb{R}^{T\times n}$, corresponding to $Tn+(n-1)[kpn+Tp+T]$. Then, the solution for the inequality $Tnk \geq Tn+(n-1)[kpn+Tp+T]$ suggests that a confidentiality breach may occur after

$$k = \left\lceil \frac{Tn+(n-1)(Tp+T)}{Tn-(n-1)pn} \right\rceil$$

iterations.

(ii) Since Owner #1 can estimate $\mathbf{B}_{A_i}$ by recovering data, adding noise to the intermediate matrix reduces to the case of adding noise to the coefficients, in (i), because Owner #1 can rewrite (64) as

$$\frac{1}{N+\rho}\mathbf{I}_{\mathbf{Y}_{A_i}} + \left[\frac{\rho}{N+\rho} - 1\right] \times \frac{1}{n}\mathbf{Z}_{A_i}\underbrace{\left[\mathbf{B}_{A_i}^{k} + \left[\frac{\rho}{N+\rho} - 1\right]^{-1}\mathbf{Z}_{A_i}^{-1}\mathbf{W}_{4,A_i}\right]}_{=\mathbf{B}'_{A_i}}. \quad \square \tag{66}$$

4. Discussion

Table 1 summarizes the methods from the literature. These privacy-preserving algorithms ought to be carefully constructed, and two key components should be considered: (i) how data are distributed between data owners, and (ii) the statistical model used. Decomposition-based methods are very sensitive to data partitioning, while data transformation and cryptography-based methods are very sensitive to the problem structure. Differential privacy methods are notable exceptions, as they simply add random noise, from specific probability distributions, directly to the data. This property makes these methods appealing, but differential privacy usually involves a trade-off between accuracy and privacy.

Cryptography-based methods are usually more robust to confidentiality breaches, but they have some disadvantages: (i) some of them require a third-party to generate keys, as well as external entities to perform the computations in the encrypted domain; and (ii) there are challenges to the scalability and implementation efficiency, mostly due to the high computational complexity and overhead of existing homomorphic encryption schemes (Hoogh, 2012; Tran & Hu, 2019; Zhao et al., 2019). Regarding some protocols, such as secure multi-party computation through homomorphic cryptography, communication complexity grows exponentially with the number of records (Rathore et al., 2015).

Data transformation methods do not affect the computational time for training the model, since data owners transform their data before the model fitting process. The same is true of decomposition-based methods, in which data are split by data owners. Secure multi-party protocols have the disadvantage of transforming the information while fitting the statistical model, which implies a higher computational cost.

As mentioned above, the main challenge to the application of existing privacy-preserving algorithms in the VAR model is the fact that $\mathbf{Y}$ and $\mathbf{Z}$ share a high percentage of values, not only during the fitting of the statistical model but also when using it to perform forecasts. A confidentiality breach can occur during the forecasting process if, after the model is estimated, the algorithm to maintain privacy provides the coefficient matrix $\mathbf{B}$ for all data owners. When using the estimated model to perform forecasts, we assume that each ith data owner sends its own contribution for time series forecasting to every other jth data owner:

1. In LASSO-VAR models with one lag, since the ith data owner sends $y_{i,t}[\mathbf{B}^{(1)}]_{i,j}$ for the jth data owner, the value $y_{i,t}$ may be directly recovered when the coefficient $[\mathbf{B}^{(1)}]_{i,j}$ is known by all data owners, being $[\mathbf{B}^{(1)}]_{i,j}$ the coefficient associated with lag 1 of time series i, to estimate j.
2. In LASSO-VAR models with p consecutive lags, forecasting a new timestamp only requires the introduction of one new value in the covariate matrix of the ith data owner. In other words, after h timestamps, the jth data owner receives h values. However, there are $h+p$ values that the data owner does not know about. This may represent a confidentiality breach, since a semi-trusted data owner can assume different possibilities for the initial p values and then generate possible trajectories.
3. In LASSO-VAR models with p non-consecutive lags, $p_1, \ldots, p_p$, after $p_p - p_{p-1}$ timestamps, only one new value is introduced in the covariate matrix, meaning that the model is also subject to a confidentiality breach.

Therefore, and considering the issue of data naturally split by features, it would be more advantageous to apply decomposition-based methods, since the time required for model fitting is unaffected by data transformations and data owners only have access to their own coefficients. However, with state-of-the-art approaches, it is difficult to guarantee that these techniques can indeed offer a robust solution to data privacy when addressing data split by features.

Finally, we offer a remark on specific business applications of VAR, where data owners know some exact past values of competitors. For example, consider a VAR model with lags $\Delta t = 1$, 2 and 24, which predicts the production of solar plants. When forecasting the first sunlight hour of a day, all data owners will know that the previous lags 1 and 2 have zero production (no sunlight). Irrespective of whether the coefficients are shared, a confidentiality breach may occur. In these special cases, the estimated coefficients cannot be used for a long time horizon, and online learning may represent an efficient alternative.

The privacy issues analyzed in this paper are not restricted to the VAR model, nor to point forecasting tasks. Probabilistic forecasts, using data from different data owners (or geographical locations), can be generated with splines quantile regression (Tastu et al., 2013), component-wise gradient boosting (Bessa, Trindade, Silva et al., 2015), a VAR that estimates the location parameter (mean) of data transformed by a logit-normal distribution (Dowell & Pinson, 2015), linear quantile regression with LASSO regularization (Agoua et al., 2018), and others.

Table 1
Summary of state-of-the-art privacy-preserving approaches.

		Split by features	Split by records
Data transformation		Mangasarian (2011)	Dwork et al. (2014), Mangasarian (2012), Yu et al. (2008)
Secure multi-party computation	Linear algebra	Du et al. (2004), Fan and Xiong (2014)[a], Karr et al. (2009), Zhu et al. (2015), Soria-Comas et al. (2017)	Aono et al. (2017), Zhu et al. (2015)
	Homomorphic-cryptography	Gascón et al. (2017), Hall et al. (2011), Slavkovic et al. (2007), Yang et al. (2019)	Chen et al. (2018), Hall et al. (2011), Jia et al. (2018), Nikolaenko et al. (2013), Slavkovic et al. (2007), Yang et al. (2019)
Decomposition-based methods	Pure	Pinson (2016), Zhang and Wang (2018)	Ahmadi et al. (2010), Lu et al. (2015), Mateos et al. (2010), Wu et al. (2012)
	Linear algebra	Han et al. (2010), Li et al. (2015)	Huang et al. (2019), Zhang et al. (2018), Zhang and Zhu (2017)
	Homomorphic-cryptography	Li and Cao (2012)[a], Yang et al. (2019), Liu et al. (2018)[a], Li et al. (2018)[a], Fienberg et al. (2009), Mohassel and Zhang (2017)	Fienberg et al. (2009), Mohassel and Zhang (2017), Yang et al. (2019), Zhang et al. (2019)

[a]Secure data aggregation.

These are some examples of collaborative probabilistic forecasting methods. However, none of them considers the confidentiality of data. Moreover, the method proposed by Dowell and Pinson (2015) can be influenced by the confidentiality breaches discussed throughout this paper, since the VAR model is directly used to estimate the mean of transformed data from different data owners. By contrast, when performing non-parametric models such as quantile regression, each quantile is estimated by solving an independent optimization problem, which means that the risk of a confidentiality breach increases with the number of quantiles being estimated. (Note that quantile regression-based models may be solved through the ADMM (Zhang et al., 2019).) However, as discussed in Section 2.3, a semi-trusted agent can collect enough information to infer confidential data. The quantile regression method may also be estimated by applying linear programming algorithms (Agoua et al., 2018), which may be solved through homomorphic encryption, despite being computationally demanding for high-dimensional multivariate time series.

5. Conclusion

This paper presented a critical overview of techniques used to handle privacy issues in collaborative forecasting methods. In addition, we analyzed their application to the VAR model. The techniques were divided into three groups of approaches: data transformation, secure multi-party computation, and decomposition of the optimization problem into sub-problems.

For each group, several points can be concluded. Starting with *data transformation techniques*, two remarks were made. The first concerns the addition of random noise to the data. While the algorithm is simple to apply, this technique demands a trade-off between privacy and the correct estimation of the model's parameters (Yang et al., 2019). In our experiments, there was clear model degradation even though the data continued to provide relevant information (see Section 3.3.1). The second relates to the multiplication by an undisclosed random matrix. Ideally, and in what concerns data where different data owners observe different variables, this secret matrix would post-multiply data, thus enabling each data owner to generate a few lines of this matrix. However, as demonstrated in Eq. (8) in Section 2.1.2, this transformation does not preserve the estimated coefficients, and the reconstruction of the original model may require sharing the matrices used to encrypt the data, thus exposing the original data.

The second group of techniques, *secure multi-party computation*, introduce privacy to the intermediate computations by defining the protocols for addition and multiplication of the private datasets. Confidentiality breaches are avoided by using either linear algebra or homomorphic encryption methods. For independent records, data confidentiality is guaranteed for (ridge) linear regression through linear algebra-based protocols; not only do records need to be independent, but some also require that the target variable is known by all data owners. These assumptions might prevent their application when covariates and target matrices share a large proportion of values—in the case of the VAR model, for instance. This means that data shared between agents might be enough for competitors to be able to reconstruct the data. Homomorphic cryptography methods can result in computationally demanding techniques, since each dataset value must be encrypted. The protocols we discussed preserve privacy while using (ridge) linear regression, provided that there are two entities that correctly perform the protocol without agent collusion. These entities are an external server (e.g., a cloud server) and an entity that generates the encryption keys. In some approaches, all data owners know the coefficient matrix $\mathbf{B}$ after model estimation. This is a disadvantage when applying models in which covariates include the lags of the target variable, because confidentiality breaches can occur during the forecasting phase.

Finally, *decomposition of the optimization problem* into sub-problems (which can be solved in parallel) have all

the desired properties of a collaborative forecasting problem, since data owners only estimate their own coefficients. A common assumption of such methods is that the objective function is decomposable. However, these approaches consist of iterative processes that require sharing intermediate results for the next update, meaning that each new iteration conveys more information about the secret datasets to the data owners, with the possibly of breaching data confidentiality.

Acknowledgments

The research leading to this work is being carried out as part of the Smart4RES project (European Union's Horizon 2020, No. 864337). Carla Gonçalves was supported by the Portuguese funding agency, FCT (Fundação para a Ciência e a Tecnologia), within the Ph.D. grant PD/BD/128189/2016 with financing from POCH (Operational Program of Human Capital) and the EU. The sole responsibility for the content lies with the authors. It does not necessarily reflect the opinion of the Innovation and Networks Executive Agency (INEA) or the European Commission (EC), which are not responsible for any use that may be made of the information it contains.

Appendix A. Differential privacy

Mathematically, a randomized mechanism $\mathcal{A}$ satisfies (ε, δ)-differential privacy (Dwork & Smith, 2009) if, for every possible output t of $\mathcal{A}$ and for every pair of datasets $\mathbf{D}$ and $\mathbf{D}'$ (differing in at most one record),

$$\Pr(\mathcal{A}(\mathbf{D}) = t) \leq \delta + \exp(\varepsilon)\Pr(\mathcal{A}(\mathbf{D}') = t). \tag{A.1}$$

In practice, differential privacy can be achieved by adding random noise W to some desirable function f of the data $\mathbf{D}$. That is,

$$\mathcal{A}(\mathbf{D}) = f(\mathbf{D}) + W. \tag{A.2}$$

The $(\varepsilon,0)$-differential privacy is achieved by applying noise from a Laplace distribution with scale parameter $\frac{\Delta f_1}{\varepsilon}$, with $\Delta f_k = \max\{\|f(\mathbf{D}) - f(\mathbf{D}')\|_k\}$. A common alternative is the Gaussian distribution, but in this case, $\delta > 0$ and the scale parameter that allows (ε, δ)-differential privacy is $\sigma \geq \sqrt{2\log\left(\frac{1.25}{\delta}\right)}\frac{\Delta_2 f}{\varepsilon}$. Dwork and Smith (2009) showed that the data can be masked by considering

$$\mathcal{A}(\mathbf{D}) = \mathbf{D} + \mathbf{W}. \tag{A.3}$$

Appendix B. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ijforecast.2020.06.003.

References

Agarwal, A., Dahleh, M., & Sarkar, T. (2019). A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM conference on economics and computation* (pp. 701–726).

Agoua, X. G., Girard, R., & Kariniotakis, G. (2018). Probabilistic models for spatio-temporal photovoltaic power forecasting. *IEEE Transactions on Sustainable Energy*, *10*(2), 780–789.

Ahmad, H. W., Zilles, S., Hamilton, H. J., & Dosselmann, R. (2016). Prediction of retail prices of products using local competitors. *International Journal of Business Intelligence and Data Mining*, *11*(1), 19–30.

Ahmadi, H., Pham, N., Ganti, R., Abdelzaher, T., Nath, S., & Han, J. (2010). Privacy-aware regression modeling of participatory sensing data. In *Proceedings of the 8th acm conference on embedded networked sensor systems* (pp. 99–112). ACM.

Ansley, C. F., & Kohn, R. (1986). A note on reparameterizing a vector autoregressive moving average model to enforce stationarity. *Journal of Statistical Computation and Simulation*, *24*(2), 99–106.

Aono, Y., Hayashi, T., Phong, L. T., & Wang, L. (2017). Input and output privacy-preserving linear regression. *IEICE Transactions on Information and Systems*, *100*(10), 2339–2347.

Aviv, Y. (2003). A time-series framework for supply-chain inventory management. *Operational Research*, *51*(2), 175–342.

Aviv, Y. (2007). On the benefits of collaborative forecasting partnerships between retailers and manufacturers. *Management Science*, *53*(5), 777–794.

Bacher, P., Madsen, H., & Nielsen, H. A. (2009). Online short-term solar power forecasting. *Solar Energy*, *83*(10), 1772–1783.

Bessa, R. J., Rua, D., Abreu, C., Machado, P., Andrade, J. R., Pinto, R., Gonçalves, C., & Reis, M. (2018). Data economy for prosumers in a smart grid ecosystem. In *Proceedings of the 9th international conference on future energy systems* (pp. 622–630). ACM.

Bessa, R., Trindade, A., & Miranda, V. (2015). Spatial-temporal solar power forecasting for smart grids. *IEEE Transactions on Industrial Informatics*, *11*(1), 232–241.

Bessa, R. J., Trindade, A., Silva, C. S., & Miranda, V. (2015). Probabilistic solar power forecasting in smart grids using distributed information. *International Journal of Electrical Power & Energy Systems*, *72*, 16–23.

Boyd, S., Parikh, N., Chu, E., Peleato, B., & Eckstein, J. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, *3*(1), 1–122.

Cavalcante, L., & Bessa, R. J. (2017). Solar power forecasting with sparse vector autoregression structures. In *PowerTech, 2017 IEEE manchester* (pp. 1–6). IEEE.

Cavalcante, L., Bessa, R. J., Reis, M., & Dowell, J. (2017). LASSO vector autoregression structures for very short-term wind power forecasting. *Wind Energy*, *20*(4), 657–675.

Chen, Y.-R., Rezapour, A., & Tzeng, W.-G. (2018). Privacy-preserving ridge regression on distributed data. *Information Sciences*, *451*, 34–49.

Dai, W., Wang, S., Xiong, H., & Jiang, X. (2018). Privacy preserving federated big data analysis. In *Guide to big data applications* (pp. 49–82). Springer.

Dowell, J., & Pinson, P. (2015). Very-short-term probabilistic wind power forecasts by sparse vector autoregression. *IEEE Transactions on Smart Grid*, *7*(2), 763–770.

Du, W., Han, Y. S., & Chen, S. (2004). Privacy-preserving multivariate statistical analysis: Linear regression and classification. In *SIAM international conference on data mining* (pp. 222–233). SIAM.

Dwork, C., & Smith, A. (2009). Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, *1*(2), 135–154.

Dwork, C., Talwar, K., Thakurta, A., & Zhang, L. (2014). Analyze gauss: optimal bounds for privacy-preserving principal component analysis. In *Proceedings of the 46th annual ACM symposium on theory of computing* (pp. 11–20). ACM.

Fan, L., & Xiong, L. (2014). An adaptive approach to real-time aggregate monitoring with differential privacy. *IEEE Transactions on knowledge and data engineering*, *26*(9), 2094–2106.

Fienberg, S. E., Nardi, Y., & Slavković, A. B. (2009). Valid statistical analysis for logistic regression with multiple sources. In *Protecting persons while protecting the people* (pp. 82–94). Springer.

Gascón, A., Schoppmann, P., Balle, B., Raykova, M., Doerner, J., Zahur, S., & Evans, D. (2017). Privacy-preserving distributed linear regression on high-dimensional data. *Proceedings on Privacy Enhancing Technologies*, *2017*(4), 345–364.

Hall, R., Fienberg, S. E., & Nardi, Y. (2011). Secure multiple linear regression based on homomorphic encryption. *Journal of Official Statistics*, *27*(4), 669.

Han, S., Ng, W. K., Wan, L., & Lee, V. C. (2010). Privacy-preserving gradient-descent methods. *IEEE Transactions on Knowledge and Data Engineering*, *22*(6), 884–899.

Hoogh, S. (2012). Design of large scale applications of secure multiparty computation: secure linear programming. Technische Universiteit Eindhoven.

Huang, Z., Hu, R., Guo, Y., Chan-Tin, E., & Gong, Y. (2019). DP-ADMM: ADMM-based distributed learning with differential privacy. *IEEE Transactions on Information Forensics and Security*, *15*, 1002–1012.

Jain, Y., & Bhandare, S. K. (2011). Min max normalization based data perturbation method for privacy protection. *International Journal of Computer & Communication Technology*, *2*(8), 45–50.

Jia, Q., Guo, L., Jin, Z., & Fang, Y. (2018). Preserving model privacy for machine learning in distributed systems. *IEEE Transactions on Parallel and Distributed Systems*, *29*(8), 1808–1822.

Jia, W., Zhu, H., Cao, Z., Dong, X., & Xiao, C. (2014). Human-factor-aware privacy-preserving aggregation in smart grid. *IEEE Systems Journal*, *8*(2), 598–607.

Karr, A. F., Lin, X., Sanil, A. P., & Reiter, J. P. (2009). Privacy-preserving analysis of vertically partitioned data using secure matrix products. *Journal of Official Statistics*, *25*(1), 125.

Kurtulmus, A., & Daniel, K. (2018). Trustless machine learning contracts; evaluating and exchanging machine learning models on the ethereum blockchain. (pp. 1–11). arXiv:1802.10185.

Li, Q., & Cao, G. (2012). Efficient and privacy-preserving data aggregation in mobile sensing. In *2012 20th ieee international conference on network protocols* (pp. 1–10). IEEE.

Li, Y., & Genton, M. G. (2009). Single-index additive vector autoregressive time series models. *Scandinavian Journal of Statistic*, *36*(3), 369–388.

Li, Y., Jiang, X., Wang, S., Xiong, H., & Ohno-Machado, L. (2015). Vertical grid logistic regression (vertigo). *Journal of the American Medical Informatics Association*, *23*(3), 570–579.

Li, S., Xue, K., Yang, Q., & Hong, P. (2018). PPMA: privacy-preserving multisubset data aggregation in smart grid. *IEEE Transactions on Industrial Informatics*, *14*(2), 462–471.

Liu, K., Giannella, C., & Kargupta, H. (2008). A survey of attack techniques on privacy-preserving data perturbation methods. In *Privacy-preserving data mining* (pp. 359–381). Springer.

Liu, Y., Guo, W., Fan, C.-I., Chang, L., & Cheng, C. (2018). A practical privacy-preserving data aggregation (3PDA) scheme for smart grid. *IEEE Transactions on Industrial Informatics*, *15*(3), 1767–1774.

Lu, C.-L., Wang, S., Ji, Z., Wu, Y., Xiong, L., Jiang, X., & Ohno-Machado, L. (2015). Webdisco: a web service for distributed cox model learning without patient-level data sharing. *Journal of the American Medical Informatics Association*, *22*(6), 1212–1219.

Ma, X., Zhu, Y., & Li, X. (2017). An efficient and secure ridge regression outsourcing scheme in wearable devices. *Computers and Electrical Engineering*, *63*, 246–256.

Mangasarian, O. L. (2011). Privacy-preserving linear programming. *Optimization Letters*, *5*(1), 165–172.

Mangasarian, O. L. (2012). Privacy-preserving horizontally partitioned linear programs. *Optimization Letters*, *6*(3), 431–436.

Mateos, G., Bazerque, J. A., & Giannakis, G. B. (2010). Distributed sparse linear regression. *IEEE Transactions on Signal Processing*, *58*(10), 5262–5276.

Mohassel, P., & Zhang, Y. (2017). Secureml: A system for scalable privacy-preserving machine learning. In *2017 ieee symposium on security and privacy* (pp. 19–38). IEEE.

Nesterov, Y. (1998). Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, *3*(4), 5.

Nicholson, W. B., Matteson, D. S., & Bien, J. (2017). VARX-L: Structured regularization for large vector autoregressions with exogenous variables. *International Journal of Forecasting*, *33*(3), 627–651.

Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D., & Taft, N. (2013). Privacy-preserving ridge regression on hundreds of millions of records. In *2013 IEEE symposium on security and privacy* (pp. 334–348). IEEE.

Nocedal, J., & Wright, S. (2006). Numerical optimization. Springer Science & Business Media.

Papadimitriou, S., Li, F., Kollios, G., & Yu, P. S. (2007). Time series compressibility and privacy. In *Proceedings of the 33rd international conference on very large data bases* (pp. 459–470). VLDB Endowment.

Pinson, P. (2016). Introducing distributed learning approaches in wind power forecasting. In *2016 international conference on probabilistic methods applied to power systems* (pp. 1–6). IEEE.

Rathore, B. S., Singh, A., & Singh, D. (2015). A survey of cryptographic and non-cryptographic techniques for privacy preservation. *International Journal of Computer Applications*, *975*, 8887.

Ravi, S., & Al-Deek, H. (2009). Predictions of freeway traffic speeds and volumes using vector autoregressive models. *Journal of Intelligent Transportation Systems*, *13*(2), 53–72.

Slavkovic, A. B., Nardi, Y., & Tibbits, M. M. (2007). Secure logistic regression of horizontally and vertically partitioned distributed databases. In *Icdmw* (pp. 723–728). IEEE.

Song, S., Chaudhuri, K., & Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. In *Global conference on signal and information processing (GlobalSIP), 2013 IEEE* (pp. 245–248). IEEE.

Soria-Comas, J., Domingo-Ferrer, J., Sánchez, D., & Megías, D. (2017). Individual differential privacy: A utility-preserving formulation of differential privacy guarantees. *IEEE Transactions on Information Forensics and Security*, *12*(6), 1418–1429.

Tastu, J., Pinson, P., Trombe, P.-J., & Madsen, H. (2013). Probabilistic forecasts of wind power generation accounting for geographically dispersed information. *IEEE Transactions on Smart Grid*, *5*(1), 480–489.

Toda, H. Y., & Phillips, P. C. (1993). Vector autoregressions and causality. *Econometrica*, 1367–1393.

Tran, H.-Y., & Hu, J. (2019). Privacy-preserving big data analytics a comprehensive survey. *Journal of Parallel and Distributed Computing*, *134*, 207–218.

Wu, Y., Jiang, X., Kim, J., & Ohno-Machado, L. (2012). Grid binary LOgistic REgression (GLORE): building shared models without sharing data. *Journal of the American Medical Informatics Association*, *19*(5), 758–764.

Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *10*(2), 12.

Yu, S., Fung, G., Rosales, R., Krishnan, S., Rao, R. B., Dehing-Oberije, C., & Lambin, P. (2008). Privacy-preserving cox regression for survival analysis. In *Proceedings of the 14th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1034–1042). ACM.

Zhang, C., Ahmad, M., & Wang, Y. (2019). ADMM based privacy-preserving decentralized optimization. *IEEE Transactions on Information Forensics and Security*, *14*(3), 565–580.

Zhang, X., Khalili, M. M., & Liu, M. (2018). Recycled ADMM: Improve privacy and accuracy with less computation in distributed algorithms. In *2018 56th annual allerton conference on communication, control, and computing (Allerton)* (pp. 959–965). IEEE.

Zhang, Y., & Wang, J. (2018). A distributed approach for wind power probabilistic forecasting considering spatio-temporal correlation without direct access to off-site information. *IEEE Transactions on Power Systems*, *33*(5), 5714–5726.

Zhang, T., & Zhu, Q. (2017). Dynamic differential privacy for ADMM-based distributed classification learning. *IEEE Transactions on Information Forensics and Security*, *12*(1), 172–187.

Zhao, C., Zhao, S., Zhao, M., Chen, Z., Gao, C.-Z., Li, H., & Tan, Y.-a. (2019). Secure multi-party computation: Theory, practice and applications. *Information Sciences*, *476*, 357–372.

Zhou, S., Lafferty, J., & Wasserman, L. (2009). Compressed and privacy-sensitive sparse regression. *IEEE Transactions on Information Theory*, *55*(2), 846–866.

Zhu, J., He, P., Zheng, Z., & Lyu, M. R. (2015). A privacy-preserving qos prediction framework for web service recommendation. In *2015 ieee international conference on web services* (pp. 241–248). IEEE.

Ziel, F., & Weron, R. (2018). Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks. *Energy Economics*, *70*, 396–420.