Editorial version available at: https://doi.org/10.1016/j.ins.2020.08.003

Multi-aspect Renewable Energy Forecasting

Roberto Corizzo[a,b,d], Michelangelo Ceci[b,c,d], Hadi Fanaee-T[e], Joao Gama[f]

[a]*Department of Computer Science, American University - Washington D.C., US*
[b]*Department of Computer Science, University of Bari Aldo Moro - Bari, Italy*
[c]*Dept. of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, 1000, Slovenia*
[d]*National Interuniversity Consortium for Informatics (CINI) - Rome, Italy*
[e]*Center for Applied Intelligent Systems Research (CAISR), Halmstad University - Halmstad, Sweden*
[f]*Faculty of Economics, University of Porto - Portugal*

Abstract

The increasing presence of renewable energy plants has created new challenges such as grid integration, load balancing and energy trading, making it fundamental to provide effective prediction models. Recent approaches in the literature have shown that exploiting spatio-temporal autocorrelation in data coming from multiple plants can lead to better predictions. Although tensor models and techniques are suitable to deal with spatio-temporal data, they have received scarce attention in the energy domain. In this paper, we propose a new method based on Tucker tensor decomposition, capable of extracting a new feature space for the learning task. For evaluation purposes, we have investigated the performances of predictive clustering trees with the new feature space, compared to the original feature space, on three renewable energy plants. The results are favorable for the proposed method, also when compared with state-of-the-art algorithms.

1. Introduction

The presence of renewable energy sources like photovoltaic (PV) and wind parks has grown consistently during the last years, with the purpose of reducing pollution emission. However, renewable power sources are variable and

Email addresses: rcorizzo@american.edu (Roberto Corizzo), michelangelo.ceci@uniba.it (Michelangelo Ceci), hadi.fanaee@hh.se (Hadi Fanaee-T), jgama@fep.up.pt (Joao Gama)

intermittent in their energy output, because the energy produced may also depend on uncontrollable factors, such as weather conditions. This becomes an issue when learning reliable forecasting models, which are of fundamental importance in grid integration, load balancing and energy trading.

In addition, a major challenge in energy forecasting is the high data dimensionality. In fact, there are too many factors that can affect the energy outputs, and exploiting their combination effectively in the predictive modeling task is not trivial.

The rationale of this work is that modeling data as a tensor could offer a richer representation than the common "flat model" $n \times m$ (instances $\times$ features) matrix, that is beneficial in the renewable energy forecasting task. In fact, especially in the case of multi-plant datasets, a flat data model fails to consider the dependencies between data coming from different plants (spatial information), on different days and at different time (temporal information) and related to different properties (features). On the contrary, tensor-based data modeling represents such dependencies by modeling data in its natural form (in our case multivariate spatio-temporal time series).

In this work, we adopt tensor decompositions for feature extraction. Thus, the features extracted are expected to offer more accurate results when used for predictive tasks. Specifically, we adopt Tucker decomposition, since it is more flexible compared to other approaches such as PARAFAC decomposition, which assumes that all dimensions have an identical number of latent variables. Tucker decomposition, instead, allows each dimension to have a different number of components, which is relevant to our problem. For instance, the number of latent variables for energy plants can naturally be different from the number of latent variables in the time dimension.

Another more theoretical motivation for this work comes from the known issue of the possible collinearity between the several independent variables in regression models [27], [42] and [4]. Ideally, regression models are built by assuming that the independent variables $X_1, \ldots, X_n$ have high correlation with the dependent variable Y, but they are scarcely correlated with each other, offering a reliable and statistically robust model. Collinearity is a phenomenon in which two or more independent variables of a multiple regression model are highly correlated, i.e. they are partially redundant. Some of the problems induced by collinearity are the following:

- Reduced accuracy in the estimate of one variable's impact on the dependent variable Y (indeterminacy of regression coefficients);

- If observed data change unexpectedly, the coefficient estimates of the multiple regression model may change abnormally;

- Regression models could be affected by overfitting [33].

Interestingly, problems due to collinearity do not show in the models fit. The resulting model may have very small residuals, but the regression coefficients are actually poorly estimated.

This problem is strictly connected with the classical feature redundancy problem, especially in the case of linear dependency between two independent variables and it is one of the main theoretical reasons which motivate the application of feature selection/feature extraction techniques in regression tasks. A treatment suggested for data that exhibit collinearity is deleting some of the variables from a fitted model [41]. Therefore, variable subset selection is a desirable part of regression analysis. This is particularly true in our case, where features representing different aspects of weather conditions (e.g., solar irradiance and cloud cover) can hardly be considered independent (see Fig. 1).

In this paper, we face this problem with a tensor data model which should, in principle, allow us to derive a new space, that better represents feature dependencies, temporal dependencies and spatial dependencies hidden in the data, along the three orthogonal dimensions, thus avoiding collinearity without mixing-up information coming from them. In energy forecasting, this aspect is particularly important since the different variables present pairwise correlations and incorporating them as-is in predictive models impacts reduced performance due to collinearity. Moreover, in energy forecasting, there are natural time dependencies due to the cyclic nature of the geophysical phenomena under investigation (temperature, irradiance, etc.) and the spatial dependencies (some plants can show similar behaviours, possibly because they are located in the same geographical area).

Methodologically, the main contribution of this paper is to propose an adaptive tensor factorization approach, based on Tucker decomposition. The adaptive solution, able to directly process data in the form of a stream, is used to improve prediction through feature extraction. In this way:

i) The new feature space is possibly smaller than the original feature space, with a reduction of the running time of any learning algorithm.

ii) The new feature space allows us to capture spatial autocorrelation, thanks to the tensor-based data modeling, which naturally represents the spatial dimension.

iii) The new feature space allows us to deal with the concept drift phenomenon, thanks to the tensor-based data modeling which naturally represents the temporal dimension.

iv) Collinearity problems are reduced, together with possible overfitting problems.

Besides the main contribution of the paper, additional practical contributions can be summarized as follows:

1) We perform extensive experiments with the proposed approach using three multi-plant energy datasets from photovoltaic and wind parks. The experiments are performed considering different values for the input parameters, different training window sizes and different learning settings. These experiments assess different practical scenarios in energy forecasting, emphasizing which specific configurations lead to the most accurate predictions.

2) We perform a comparison of the proposed approach with state-of-the-art algorithms both for feature extraction and forecasting (namely, PCA, Auto-Encoder embeddings, neural networks, predictive clustering trees and time series forecasting), considering the standard Root Mean Square Error (RMSE) measure and the Minimum Description Length Penalization (MDLP) measure, that allows us to evaluate the prediction error and the complexity of a model simultaneously. We also include statistical validation of the results.

This twofold analysis offers an important perspective for practitioners in the energy field. In fact, selecting the ideal learning setting and its appropriate configuration, including the data representation chosen, the amount of training data to be used, and the ideal configuration of parameter values, can significantly impact the ability to predict the energy produced in different experimental conditions. This ability impacts, in turn, the opportunity to mitigate losses and risks in energy trading and in the power grid integration and scheduling of renewable energy.

The paper is structured as follows. Section 2 discusses works in the literature related to the scope of our study. Section 3 presents the method. Section 4 describes the experimental setting, the datasets and the results. Finally, Section 5 concludes the paper.

2. Background

2.1. Concept drift aware energy forecasting

In the literature, several researchers addressed the energy forecasting task with solutions which range from physical to statistical. The former rely on the refinement of NWP (Numerical Weather Prediction) forecasts with physical considerations (e.g. obstacles and orography) [8] or measured data (an approach often referred to as Model Output Statistics or MOS) [43][46], whereas the latter are based on adaptive models that establish a relationship between historical values and forecast variables. Combinations of statistical and physical approaches for renewable energy power forecasting have also been recently investigated [11]. Recently proposed approaches also use machine learning and data mining techniques (independently of whether they use NWP data or not). These are based on time series [16], whereas others learn forecasting models from data, like autoregressive (AR) models [1], predictive clustering models [20], artificial neural networks (ANNs) [15], or SVM classifiers [52].

A common aspect that is typically taken into account is the phenomenon of concept drift, due to the fact that data may change characteristics and distribution over time [5]. In this respect, it has been noted that physical property behaviors (e.g. wind speed and solar irradiation) are typically subject to the concept drift phenomenon and, when this phenomenon is present, adaptive models are generally considered to produce more reliable predictions with a continuous training phase [30].

However, to the best of our knowledge, there is no study in the energy forecasting literature that tackles this problem exploiting tensor techniques. To fill this gap, in order to handle concept drift, our method represents time as an explicit dimension of analysis in the tensor data structure. Moreover, our model is flexibly retrained using window-based data snapshots, thus incorporating the latest data available, which may be subject to changes in the data distribution.

2.2. Spatial autocorrelation in multi-plant energy datasets

An additional aspect that should be considered is the spatial proximity of plants. While most of the works in the literature consider forecasting solutions for single plants and ignore the information collected from/at other plants/sites in the vicinity, different studies in the literature [15] [14] [6] [18] have shown that considering multiple plants and the spatial autocorrelation induced by their proximity can lead to better predictions. This has been proven also in other domains, such as ecological applications [53].

On this respect, some methods in the literature, exploit the information deriving from the distance between plants. For instance, in [31], geo-distributed weather observations in the spatial proximity of a wind farm are exploited as off-site predictors. The approach in [14] extracts features that model the spatio-temporal autocorrelation between plants for each weather feature [14]. The extracted feature space can be used by off-the-shelf machine learning methods to perform forecasting.

Focusing on auto-regressive models such as Vector Auto-Regression (VAR), in [7], the authors propose a spatio-temporal framework that combines Recursive Least Squares fitting and Gradient Boosting. In a similar fashion, in [23], spatio-temporal dependencies are considered by means of a sparse parametrization of VAR models, which leads to the extraction of coefficients that link sites with a positive spatial co-dependence, and to the disregard sites with weak dependencies. In [54], the authors propose a parametric model for tracking conditional spatio-temporal dependencies, under the assumption that the local forecasting error made at time t at the target plant depends on the errors previously observed at a set of neighboring plants. In [12], multiple sparse structures for the VAR model are taken into account by exploiting the least absolute shrinkage and selection operator (LASSO) framework.

Another related subclass of approaches are based on Spatial Auto-Regressive (SAR) model, defined as:

$$\hat{e}_i = \lambda \sum_{j=1}^{K} w_{ij} e_j + \epsilon_i \; i = 1, \; \ldots, \; K, \tag{1}$$

where K is the number of training examples, $e_j = Y_j - \overline{Y}$ is the prediction error for the average, w_{ij} represents the spatial proximity or similarity between i and j, λ represents the spatial dependence, and ϵ_i is the error, which is normally distributed.

Some researchers exploited this formulation to customize traditional predictive modeling approaches, in order to consider spatial autocorrelation. In particular, [56] propose a decision tree learning algorithm that substitutes the traditional entropy-based measure with the "spatial entropy" [39]. This measure allows to catch how the entropy is dispersed over the spatial neighborhoods. Similar decision trees approaches are also presented in [51], where the spatial entropy is computed for each example as the weighted information gain of examples that overlap.

Specifically for regression, another typical way to take spatial autocorrelation into account in the learning task is Geographically Weighted Regression (GWR) [28]. In GWR, a linear regression model is associated to each point $(a,\ b)$. In this way, the weighting of an example is not a constant, but depends on $(a,\ b)$. Formally:

$$y(a,\ b) = \alpha_0(a,\ b) + \sum_k \alpha_k(a,\ b) x_k(a,\ b) + \epsilon_{(a,b)}, \tag{2}$$

where $\alpha_k(a,\ b)$ is estimated using observations close to $(a,\ b)$.

The approach of using autocorrelation aware local models is also used in Kriging [9], where an optimal linear interpolation method is exploited to estimate the response values $y(a,\ b)$ at each plant $(a,\ b)$. Such linear interpolation step takes into account a structural component, which defines a constant trend (average), a random spatially correlated component, and noise. A spatial associative classifier that simultaneously learns spatial association rules is proposed in [13]. [40] presents a regression method that captures both global and local spatial autocorrelation for the predictive attributes in the learning phase.

On the contrary of the aforementioned approaches, which train a model exclusively based on the target space, our method exploits latent dependencies between variables in the feature space, i.e. weather conditions, to optimize the forecasting of energy production. In this perspective, in our method we follow the intuition that, by exploiting one-day-ahead weather forecasts and their latent interactions, latent interactions of the variables over time and latent interactions of the variables on the geographical dimension, it is possible to provide valuable information for the forecasting task, especially when weather conditions are changing over time. The aim of this approach is to obtain an increased predictive accuracy of the model.

2.3. Complex feature space in energy datasets

One important aspect related to energy datasets is represented by the complexity of time series data, which often constitutes a limit for predictive tools when applied directly to the original data representation. Popular time series analysis techniques in the literature, such as Empirical Mode Decomposition (EMD) [50], variational mode decomposition [24], and singular spectrum analysis [25], exhibit the common behavior to simplify complex time series into a finite number of intrinsic mode functions. Such techniques have been also exploited in the energy domain, with the intention to simplify the signal and carry out the forecasting task exploiting the information from the dimensions in a combined manner [3].

However, these techniques operate on univariate time series. As a consequence, the application of such techniques on all the univariate time series leads to a consistent increase in the number of features, which negatively impacts the prediction effectiveness due to the curse of dimensionality problem.

Focusing on multi-plant energy forecasting, it is worth noting that data usually has a multi-dimensional structure with dimensions such as plants, time and features, which model spatio-temporal information and weather information. The different features present positive pairwise correlation (see Fig. 1). These conditions represent an additional challenge for forecasting tools, which translates to a reduced accuracy in forecasts. One possibility to address these challenges is resorting to feature reduction and feature extraction techniques.

A recent group of approaches for feature reduction are tensor factorizations (TFs) [45, 26], which are successfully applied in a wide range of domains from Psychometrics and Chemometrics to Environmental Monitoring and Signal Processing. Today, they are known as state-of-the-art tools for feature reduction, when learning better models from multi-dimensional data sets. Some recent reports [47, 17] demonstrate the extraordinary performance of TFs in feature extraction for the classification of high-dimensional data sets, in particular, in brain data analysis and computer vision. TF is also used in various other prediction problems. Examples include audio signals for music genre classification [44], hyperspectral images classification [49], multi-sensor vibration signals for damage detection in engineering structures [48] and telecommunication usage data for anomaly detection [55], among others. Although the effectiveness of TFs has been generally recognized in feature extraction, their application in the predictive modeling of energy production has, to the best of our knowledge, been investigated only for clustering [29].

In our method, we address the challenge of modeling the complex feature space of energy datasets by leveraging TFs as feature extraction techniques, with the aim of improving the accuracy of forecasting models.

3. Method

The task we consider in this work is to predict the power generated by multiple plants considering i) historical data on power production, ii) weather forecast data provided by NWP systems, iii) weather information collected by sensors and iv) geographic coordinates of the plants. The output is a per-hour prediction for the next day. The learning algorithms update the prediction models every day. We use historical weather information collected by sensors as features in the training phase, whereas we use weather forecast data provided by NWP systems as features for predictions.

In the next subsection we provide the basic concepts of tensors and Tucker decomposition. The proposed method is presented in subsections 3.2, 3.3 and 3.4.

3.1. Preliminary concepts: Tensors and Tucker decomposition

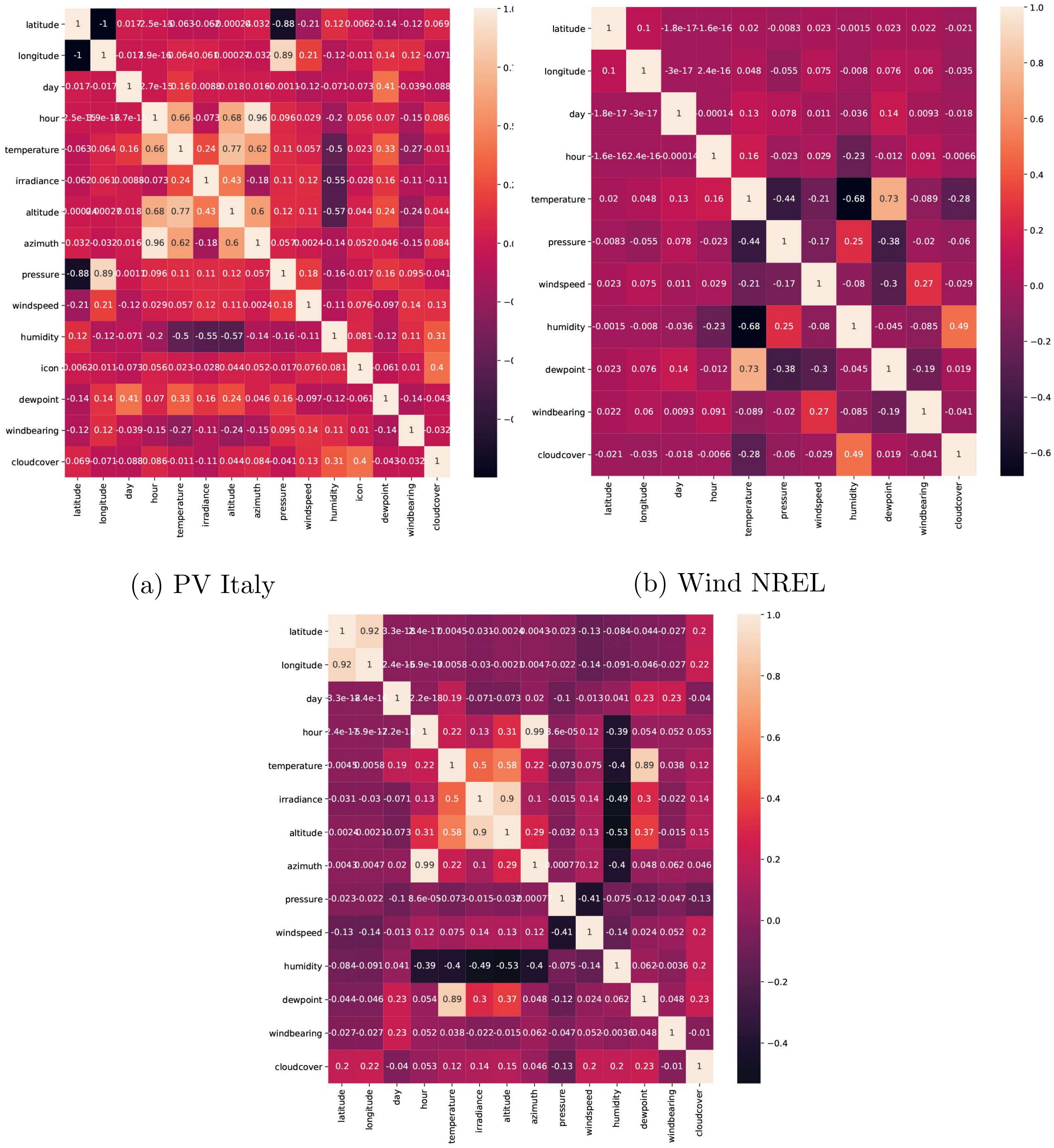
A tensor is a mathematical object for extension of scalars, vectors and matrices to higher dimensions. More formally, an N-way or Nth-order tensor is an element of the tensor product of N vector spaces, each with its own coordinate system [38].

Tucker decomposition is a form of higher-order PCA [36], which decomposes a tensor into a core tensor multiplied (or transformed) by a matrix along each way (or mode).

Given a three-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, we have

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{A}^{\mathbf{(1)}} \times_2 \mathbf{A}^{\mathbf{(2)}} \times_3 \mathbf{A}^{\mathbf{(3)}} = \sum_{p=1}^{R_1} \sum_{q=1}^{R_2} \sum_{r=1}^{R_3} g_{pqr} \cdot \mathbf{a}^{\mathbf{1}}_{\mathbf{p}} \circ \mathbf{a}^{\mathbf{2}}_{\mathbf{q}} \circ \mathbf{a}^{\mathbf{3}}_{\mathbf{r}} \qquad (3)$$

where, $\mathbf{A}^{\mathbf{(1)}} \in \mathbb{R}^{I1 \times R_1}$, $\mathbf{A}^{\mathbf{(2)}} \in \mathbb{R}^{I_2 \times R_2}$ and $\mathbf{A}^{\mathbf{(3)}} \in \mathbb{R}^{I_3 \times R_3}$ are the factor matrices and can be thought of as the principal components in each mode, whereas $\mathcal{G} \times_1 \mathbf{A}^{\mathbf{(1)}} \times_2 \mathbf{A}^{\mathbf{(2)}} \times_3 \mathbf{A}^{\mathbf{(3)}}$ is the *n-mode (matrix) product* of a tensor and a matrix. The tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is called the core tensor and its entries show the level of interaction between the different components. Finally, R_1, R_2, and R_3 are the number of components (i.e. columns) in the



(a) PV Italy

(b) Wind NREL

(c) LightSource

Figure 1: Correlation matrices of the different datasets analyzed in this study.

factor matrices $\mathbf{A^{(1)}}$, $\mathbf{A^{(2)}}$ and $\mathbf{A^{(3)}}$, respectively. A graphical representation of a Tucker decomposition of a three-way array is shown in Fig. 2.

There are several methods for fitting the Tucker model. The most popular one is the Alternating Least Square (ALS), which is also used in this work (see Algorithm 1 for more details).

Algorithm 1: The Tucker decomposition algorithm: Alternating least squares algorithm to compute a rank-$(R_1, R_2, ..., R_N)$ Tucker decomposition for an Nth order tensor $\mathcal{X}$ of size $I_1 \times I_2 \times \cdots \times I_N$. $\mathbf{Y}^{(i)}$ is the i-th mode of the tensor $\mathcal{Y}$.

```
Data: $\mathcal{X}, R_1, R_2, \ldots, R_N$
Result: $\mathcal{G}, \mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \ldots, \mathbf{A}^{(N)}$
initialize $\mathbf{A}^{(i)} \in \mathbb{R}^{I_i \times R_i}$ for $i = 1, \ldots, N$ using HOSVD;
repeat
    for $i = 1, \ldots, N$ do
        $\mathcal{Y} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\top} \cdots \times_{i-1} \mathbf{A}^{(i-1)\top} \times_{i+1} \mathbf{A}^{(i+1)\top} \cdots \times_N \mathbf{A}^{(N)\top}$;
        $\mathbf{A}^{(i)} \leftarrow R_n$ leading singular vectors of $\mathbf{Y}^{(i)}$;
until fit does not improve or maximum number of iterations is reached;
$\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)\top} \times_2 \mathbf{A}^{(2)\top} \cdots \times_N \mathbf{A}^{(N)\top}$;
```

One of the natural applications of the Tucker decomposition is feature reduction (as in PCA). This is because the core tensor G can be seen as a compressed version of X [38], while the factor matrices can be seen as the principal components in each mode [38][48]. In this work we exploit this property for the forecasting task described in Section 1.

3.2. The learning task and the feature extraction process

The learning task is performed according to the self-adaptive online training strategy, using a time-based sliding window S [30] of size s (past s days of historical data) or using a time-based landmark window L of increasing size s (considering all historical data available), whereas the forecasting horizon is one-day-ahead, hour by hour (see Fig. 7). This means that the prediction model is updated every day, that the learning phase only takes into account data collected in the last s days and that the learned forecasting model is used

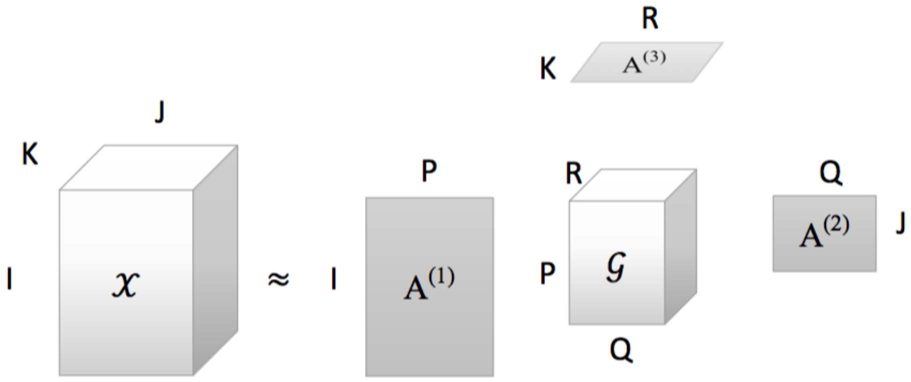


Figure 2: Tucker decomposition of a three-way array $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ modeled as a core tensor $\mathcal{G} \in \mathbb{R}^{P\times Q\times R}$ and factor matrices $\mathbf{A^{(1)}} \in \mathbb{R}^{I\times P}$, $\mathbf{A^{(2)}} \in \mathbb{R}^{J\times Q}$, $\mathbf{A^{(3)}} \in \mathbb{R}^{K\times R}$.

only for predicting the energy produced for the next day. In this scenario, we have two variants of the learning task:

- *Hourly variant*: We consider the hour of the day as the reference unit of observation. That is, for a specified plant, a single day consists of 24 instances represented in terms of m independent variables, each associated with a target label representing the observed production for the specified hour.

- *Daily variant*: We consider the day as the reference unit of observation. That is, for a specified plant, a single day consists of one single instance, represented in terms of m independent *time series*. Each instance is associated with a vector of 24 target labels which represents the time series of production for the whole day.

More technically, given a time-based sliding window S (or a time-based landmark window L) of historical data (training set/labeled instances), i.e. instances for which the numeric target variable (power) is known, and weather predictions W for each plant for the next day (unlabeled instances), i.e. instances for which the numeric target variable (power) is not known, we represent S (or L) as a fourth order tensor $X_{labeled}(F, P, D, H)$ and W as a third order tensor $X_{unlabeled}(F, P, H)$, where each dimension is represented as follows. F: Features (set of weather variables); P: Plants (set of plants); D:

Days (set of days in the time window; $|D| = s$); H: Hours (set of hours of the day).

It is noteworthy that the tensors $X_{labeled}(F, P, D, H)$ and $X_{unlabeled}(F, P, H)$ model the spatial, temporal and feature-based dimensions of the data,

Once $X_{labeled}(F, P, D, H)$ and $X_{unlabeled}(F, P, H)$ are generated, they can be the subject of the feature extraction process, performed via Tucker decomposition for both the Hourly or Daily variants. The goal is to obtain the following:

- A new training set $R_{labeled}$ ($n \times r$ matrix) of reduced dimensionality, with the same labels Y of the original training set S (or L);

- A new set $R_{unlabeled}$ (matrix) of reduced dimensionality for unlabeled instances with the same feature space of $R_{labeled}$.

Any additional time invariant feature which cannot be directly mapped in F, P, D, and H, can be subsequently added to $R_{labeled}$ and $R_{unlabeled}$. This is the case of the spatial coordinates of the plants (latitude, longitude).

The proposed variants for the feature extraction process are explained in detail in the following subsections.

It is important to note that the granularity considered in this study depends on the application at hand. In fact, in the energy field, a forecasting horizon of 24 hours at a hourly granularity is the most useful to perform renewable energy scheduling, integration, and trading. However, the proposed method is not dependent on a specific granularity, neither in the time dimension nor in the spatial dimension. Therefore, our method is flexible and the selection of the time granularity can be generalized and decided for any application at hand.

3.3. The hourly variant

Given the fourth order tensor $X_{labeled}(F, P, D, H)$ and the third order tensor $X_{unlabeled}(F, P, H)$, they are joined in the same structure X of size $|F| \cdot |P| \cdot (|D| + 1) \cdot |H|$.

The Tucker decomposition is performed on X with a desired value of $rank$ for each mode (R_D, R_F, R_P, R_H), resulting in a core tensor G and the factor matrices R (U_D of size $|D| + 1 \cdot R_D$, U_F of size $|F| \cdot R_F$, U_P of size $|P| \cdot R_P$ and U_H of size $|H| \cdot R_H$).

The core tensor G and its entries show the level of dependency between the different dimensions, whereas the factors (which contain orthogonal vectors) can be thought of as the principal components in each mode [38]. This motivates us to exploit factor matrices (namely $\mathbf{A^{(1)}}, \mathbf{A^{(2)}}, ..., \mathbf{A^{(N)}}$) in the feature extraction, which is also coherent with the usual exploitation of PCA for feature extraction from matrices, as well as with the exploitation of factor matrices obtained by PARAFAC tensor factorization (see [48]).

$R_{labeled}$ is obtained by performing a "Cartesian product"[1] between the matrices U_D, U_P, U_H. More specifically, the first $|D|$ rows of the matrix U_D are concatenated with all the rows of U_H and U_P. Similarly, $R_{unlabeled}$ is obtained as the "Cartesian product" between the $(|D|+1)$-th row of U_D and the matrices U_H, U_P. Analytically, given the matrices U_D,U_P and U_H, as well as the matrices $\tilde{U}^D$, $\tilde{U}^H$, $\tilde{U}^P$, $\overline{U}^D$, $\overline{U}^H$, $\overline{U}^P$ defined in Fig. 3, $R_{labeled}$ and $R_{unlabeled}$ are defined as follows:

$$\begin{aligned} R_{labeled} = \quad & (\tilde{U}^D(1:|D|\cdot|H|\cdot|P|, 1:|D|) \times U^D(1:|D|,*) \times \\ & \times\overline{U}^D) + (\tilde{U}^P(1:|D|\cdot|H|\cdot|P|,*) \times U^P \times \overline{U}^P) + \\ & +(\tilde{U}^H(1:|D|\cdot|H|\cdot|P|,*) \times U^H \times \overline{U}^H) \end{aligned} \tag{4}$$

$$\begin{aligned} R_{unlabeled} = {} & (\tilde{U}^D(|D|\cdot|H|\cdot|P|+1:|D|+1\cdot|H|\cdot|P|, |D|+1) \times \\ & \times U^D(|D|+1,*) \times \overline{U}^D) + \\ & +(\tilde{U}^P(|D|\cdot|H|\cdot|P|+1:|D|+1\cdot|H|\cdot|P|,*) \times U^P \times \overline{U}^P) + \\ & +(\tilde{U}^H(|D|\cdot|H|\cdot|P|+1:|D|+1\cdot|H|\cdot|P|,*) \times U^H \times \overline{U}^H) \end{aligned} \tag{5}$$

At the end of the process, we obtain a matrix $R_{labeled}$ of $|D|\cdot|H|\cdot|P|$ rows and $R_D + R_H + R_P$ columns (representing historical data) and a matrix $R_{unlabeled}$ of $1\cdot|H|\cdot|P|$ rows and $R_D + R_H + R_P$ columns (representing the next day). See Fig. 5 for a schematic representation of $R_{labeled}$ and $R_{unlabeled}$.

$R_{labeled}$ and $R_{unlabeled}$ don't natively include labels (target space), as they are a new representation of the original feature space. The labels (namely

[1] The term "Cartesian product" is a clear abuse of notation. This is necessary in order to simplify the explanation of the method. Analytic definitions are provided in the following. The symbols : and $*$ denote the selection of a range of rows or columns, and all rows or columns, respectively.

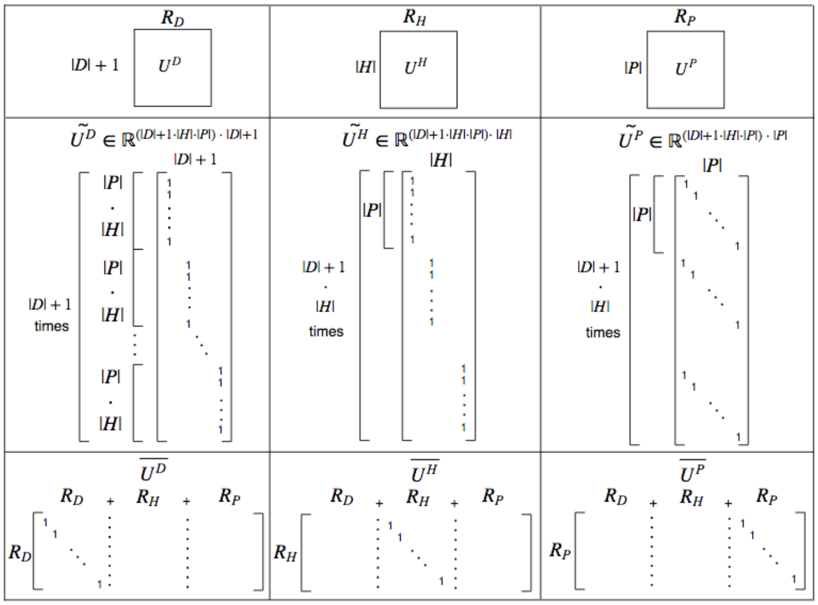

R_D
$|D|+1$
U^D
R_H
$|H|$
U^H
R_P
$|P|$
U^P
$\widetilde{U^D} \in \mathbb{R}^{(|D|+1 \cdot |H| \cdot |P|) \cdot |D|+1}$
$\widetilde{U^H} \in \mathbb{R}^{(|D|+1 \cdot |H| \cdot |P|) \cdot |H|}$
$\widetilde{U^P} \in \mathbb{R}^{(|D|+1 \cdot |H| \cdot |P|) \cdot |P|}$
$|D|+1$ times
$|P| \cdot |H|$
$|D|+1 \cdot |H|$ times
$\overline{U^D}$
$\overline{U^H}$
$\overline{U^P}$
$R_D + R_H + R_P$

Figure 3: Hourly variant: definition of the matrices $\tilde{U}^D$, $\tilde{U}^H$, $\tilde{U}^P$, $\overline{U}^D$, $\overline{U}^H$, $\overline{U}^P$.

the production observed for a specified plant and hour of the day) can be then incorporated from the original dataset (one value per row). The Hourly variant of the feature extraction process is graphically depicted in the upper part of Fig. 4.

Finally, in order to avoid problems due to different ranges for the values of features, we perform a column-wise min-max normalization [32] of the feature space of $R_{labeled}$ and $R_{unlabeled}$ to scale the values of each feature in [0, 1].

3.4. The daily variant

This variant works similarly to the Hourly variant: the Tucker decomposition on a tensor structure X is derived from $X_{labeled}$ and $X_{unlabeled}$. However, in order to consider the whole day as a reference unit of observation, the factor matrices considered for the feature extraction are U_D and U_P. The resulting matrix $R_{labeled}$ is obtained by a "Cartesian product" between the first $|D|$ rows of U_D and U_P, whereas the matrix $R_{unlabeled}$ is derived from a "Cartesian product" between the last row $|D|+1$ of U_D and U_P.

Formally, given the matrices U_D and U_P, as well as the matrices $\tilde{U}^D$, $\tilde{U}^P$ defined in Fig. 6, $\overline{U}^D$, $\overline{U}^P$, $R_{labeled}$ and $R_{unlabeled}$ are defined as follows:

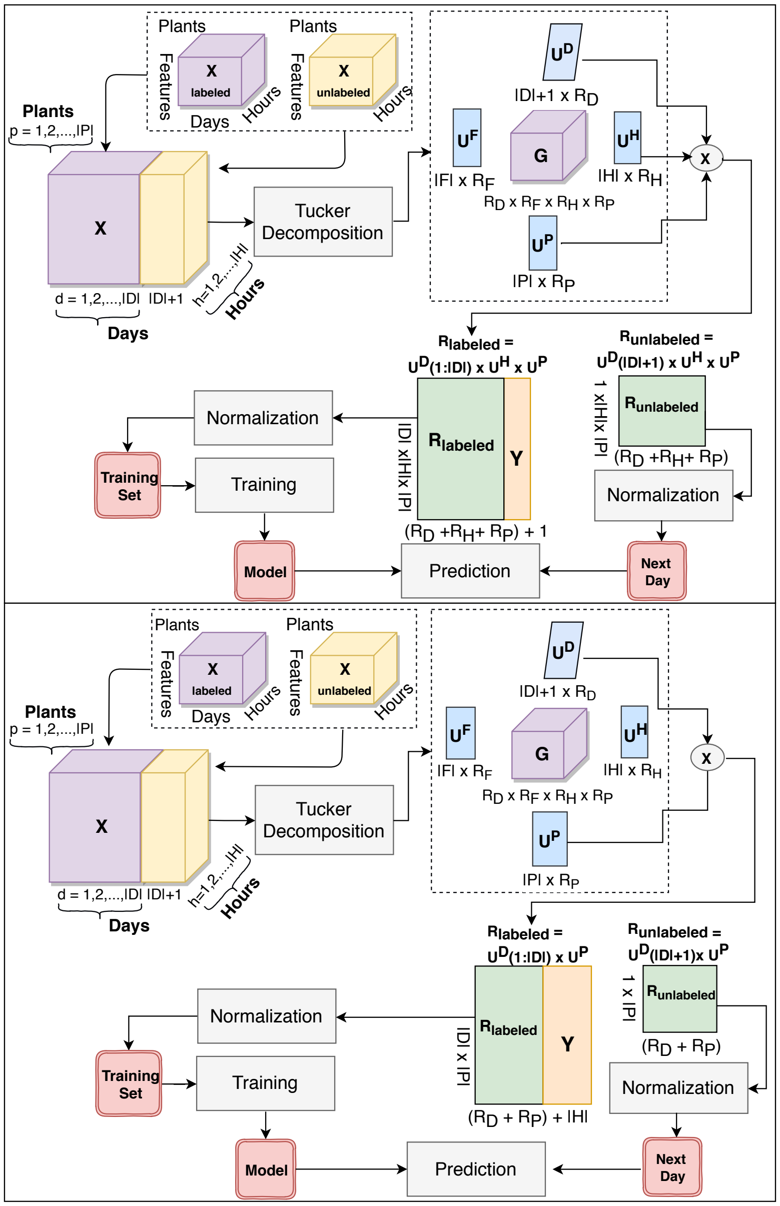


Figure 4: The feature extraction process in detail (top: Hourly variant, bottom: Daily variant). In the Hourly variant, Y represents the energy produced (single value) for a given triple (day, hour, plant), and ”X” denotes the “Cartesian product” among the factor matrices U^D,U^H,U^P. In the Daily variant,Y represents the time series (vector) of the energy produced for a given pair (day, plant), and ”X” denotes the “Cartesian product” among the factor matrices U^D and U^P.

$$
\begin{aligned}
R_{labeled} = \quad & \tilde{U}^D(1:|D|\cdot|P|, 1:|D|) \times U^D(1:|D|,*) \times \overline{U}^D + \\
& +\tilde{U}^P(1:|D|\cdot|P|,*) \times U^P \times \overline{U}^P
\end{aligned} \tag{6}
$$

$$
\begin{aligned}
R_{unlabeled} = \quad & \tilde{U}^D(|D|\cdot|P|+1:|D|+1\cdot|P|, |D|+1) \times \\
& \times U^D(|D|+1,*) \times \overline{U}^D + \\
& + \tilde{U}^P(|D|\cdot|P|+1:|D|+1\cdot|P|,*) \times U^P \times \overline{U}^P
\end{aligned} \tag{7}
$$

$R_{labeled}$ is of size $|D|\cdot|P|$ rows and $R_D + R_P$ columns, whereas $R_{unlabeled}$ is of size $1\cdot|P|$ rows and $R_D + R_P$ columns.

The labels (target space) to be subsequently incorporated in $R_{labeled}$ and $R_{unlabeled}$ are the time series of production observed for a specified plant for the whole day considered, hour by hour (multi-target representation consisting of a vector of $|H|$ elements per row).

The Daily variant of the feature extraction process is graphically depicted in the lower part of Fig. 4.

As in the hourly variant and with the same motivations, we perform a min-max normalization [32] of the feature space.

4. Evaluation

We implemented the feature extraction tool in Matlab and Java exploiting the Tensor Toolbox [2]. Extracted features were then used to train predictive clustering tree models (a form of regression trees), using the CLUS algorithm [37]. This choice was motivated by recent literature which has shown that predictive clustering trees outperform all other approaches in the one-day-ahead power forecasting task [14]. CLUS considers a tree as a hierarchy of clusters (Predictive Clustering Trees - PCTs): the top-node corresponds to one cluster containing all the data, which is recursively partitioned into smaller clusters while moving down the tree. CLUS, including PCTs for multi-target regression [**?**], is available at `http://clus.sourceforge.net`.

4.1. Experimental setting

For evaluation, a random sampling of 10% of the days has been performed repeatedly, generating 5 splits (for each of the three datasets considered in this empirical evaluation). For each split, the production of each day selected

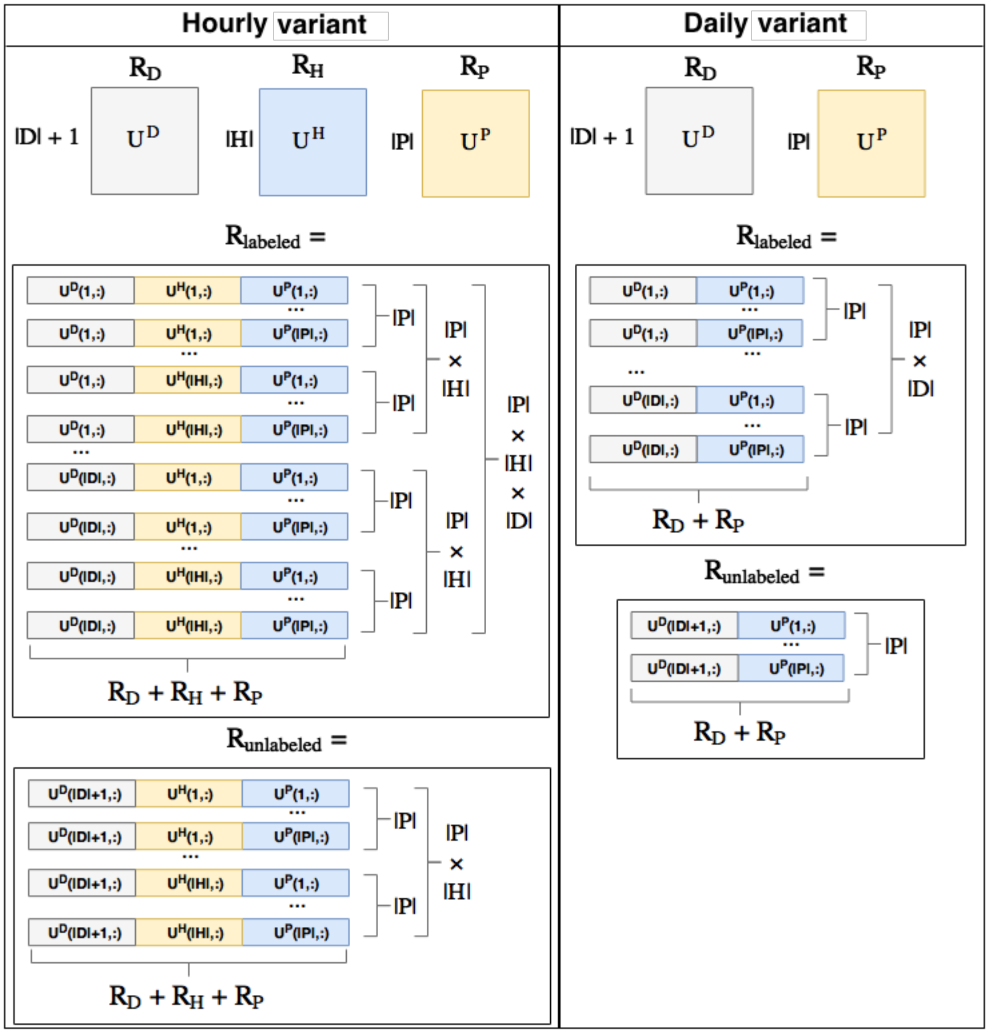


Figure 5: Graphical representation of the resulting matrices $R_{labeled} \in \mathbb{R}^{(|P|\cdot|H|\cdot|D|)\times(R_D+R_H+R_P)}$ and $R_{unlabeled} \in \mathbb{R}^{(|P|\cdot|H|)\times(R_D+R_H+R_P)}$ for the Hourly variant and $R_{labeled} \in \mathbb{R}^{(|P|\cdot|D|)\times(R_D+R_P)}$ and $R_{unlabeled} \in \mathbb{R}^{|P|\times(R_D+R_P)}$ for the Daily variant.

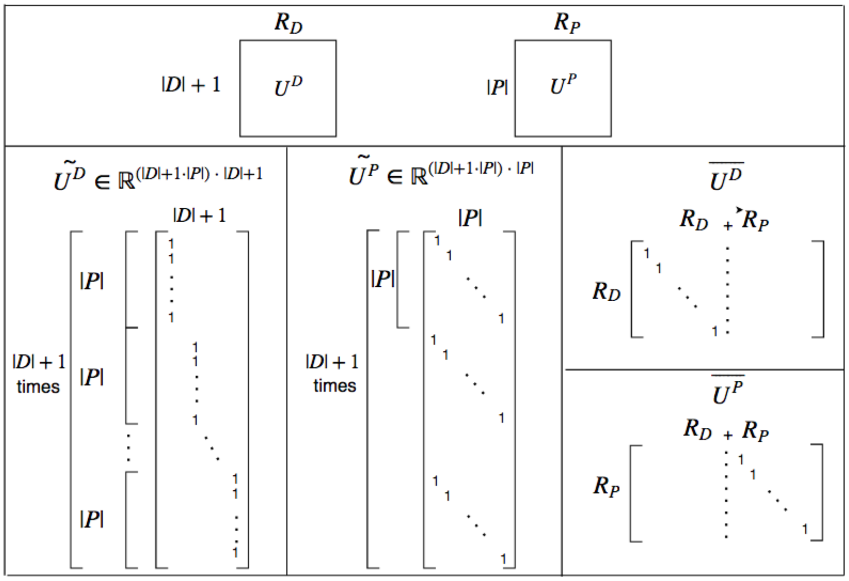

R_D
R_P
$|D|+1$
U^D
$|P|$
U^P
$\tilde{U}^D \in \mathbb{R}^{(|D|+1 \cdot |P|) \cdot |D|+1}$
$\tilde{U}^P \in \mathbb{R}^{(|D|+1 \cdot |P|) \cdot |P|}$
$\overline{U^D}$
$R_D + R_P$
$\overline{U^P}$
$|D|+1$ times
$|P|$
1

Figure 6: Daily variant: definition of the matrices $\tilde{U}^D$, $\tilde{U}^P$, $\overline{U}^D$ and $\overline{U}^P$.

as the testing day is obtained by using a model trained on a sliding window of 30, 60 or 90 days preceding the testing day, respectively, or on a landmark window model (see Fig. 7). In this way, we guarantee that data from future time instants are not being considered in the training procedure (an aspect that wouldn't be taken into account using a standard evaluation approach like cross-validation). For a fair evaluation, this setting is applied both to our method, that we call TUCKER-CLUS, and to competitive approaches.

Moreover, in order to guarantee a realistic evaluation, the values of irradiance (PV dataset) and wind speed (Wind dataset) considered for the next day (testing set) are the values queried by numerical weather prediction (NWP) models and not values observed by sensor data (not available beforehand).

In order to assess the contribution of the tensor-based feature extraction in terms of improvement in the predictive capabilities of the model, we compare its performances with three additional approaches, namely, Principal Components Analysis (PCA) [36], Auto-Encoder neural network embeddings (AE) [34, 19] [34], and Empirical Mode Decomposition (EMD) [50]. Therefore, we include comparisons with three additional methods, PCA-CLUS, AE-CLUS, and EMD-CLUS which are characterized by their own feature

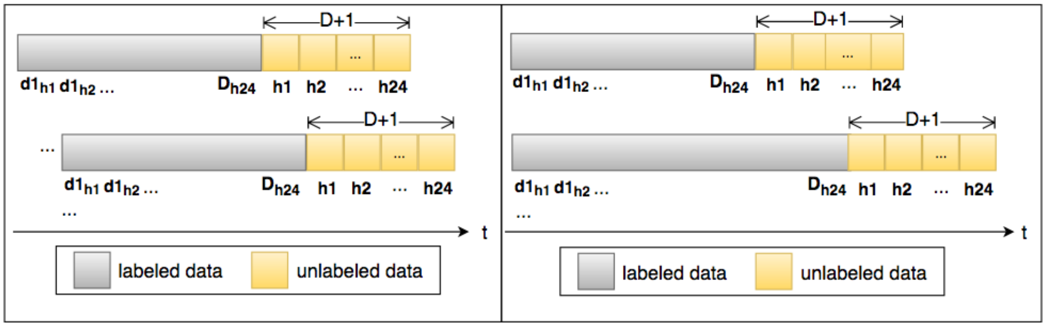


Figure 7: Evaluation (training and testing) procedure with time-based sliding window model of size D days (left) and landmark window model of increasing size D days (right).

extraction phase, and subsequently exploit CLUS as learning algorithm. For fairness, we compare the performances of TUCKER-CLUS, PCA-CLUS, AE-CLUS and EMD-CLUS when they extract the same number of features (equal value for the *rank* parameter). However, EMD decomposes each univariate time series into a finite number of intrinsic mode functions. Therefore, it returns a higher number of extracted features compared to the original feature space. To avoid an excessive increase in the number of features that would lead to the curse of dimensionality, we limit the number of dimensions extracted to 3 for each univariate time series.

All the results obtained by the method proposed in this paper (TUCKER-CLUS) are also compared with the results obtained by the ARIMA (AutoRegressive Integrated Moving Average) model. We consider ARIMA models as baseline models since they have been successfully applied in several time series analyses and forecasting tasks in the last years [10] and, for this reason, are considered state-of-the-art . Finally, we also compare our results with Long Short-Term Memory neural networks (LSTM), for their demonstrated reliability in predictive tasks with time series [35]. In particular, we adopt a DL4J[2] implementation for regression. For LSTM, the best model is selected by performing a grid search on the dropout parameter ($d \in \{0.1, 0.3, 0.5\}$) and we report the results obtained with the best configuration.

[2] https://deeplearning4j.org/

4.2. Evaluation Measures

The performance of regression models is usually evaluated by means of the standard root mean square error (RMSE) and mean absolute error (MAE) criteria. Their values for a sample of size n is calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}};\ MAE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n}; \tag{8}$$

where y_i is the actual value (measured) and $\hat{y}_i$ is the predicted value. We report RMSE and MAE results averaged over the split, the testing day and the hour of the day.

Besides RMSE and MAE, in order to evaluate simultaneously the model simplification introduced by the feature extraction and the error reduction, we also report the results in terms of the Minimum Description Length Penalization measure (MDLP) proposed in [22]. This measure, according to the Minimum Description Length principle, prefers a simpler model to a more complex model, if they are equally accurate.

In our work, this measure is employed to compare TUCKER-CLUS with CLUS, run using the initial feature set.

The MDLP measure is calculated as:

$$MDLP = -log_2 P(\mathbf{Y}|\hat{\mathbf{w}}) + q(log_2\ p + 2), \tag{9}$$

where:

- q is the number of features in the model (reduced set)
- p is the initial number of features
- $-log_2\ P(\mathbf{Y}|\hat{\mathbf{w}})$ represents the number of bits necessary to define a probability distribution of the residuals, given some parameters $\hat{\mathbf{w}}$:

  $$-log_2 P(\mathbf{Y}|\hat{\mathbf{w}}) = \frac{n}{2\ ln\ 2}\left[ln\left(2\pi \cdot RMSE_p\right) + \left(\frac{RMSE_p}{RMSE_q}\right)^2\right], \tag{10}$$

  where $RMSE_p$ and $RMSE_q$ represent the RMSE obtained with the initial and reduced number of features, respectively. (We refer to [22] for additional details).

Smaller values of MDLP indicate simpler (constructed on a smaller number of features) and/or more accurate models.

The results presented in the next section consider both the RMSE and MDLP measures. For the latter, in order to facilitate comparison, we perform min-max normalization.

4.3. Datasets

In this work, three datasets are considered:

- **PV Italy.** The data are collected at regular intervals of 15 minutes (measurements start at 2:00 and stop at 20:00 every day) by sensors located on 17 plants in Italy. The time period spans from January 1$^{\text{st}}$, 2012 to May 4$^{\text{th}}$, 2014. More details about data preparation steps performed on this dataset can be found in [14].

- **Wind NREL.** This dataset was modeled by 3TIER using the Weather Research & Forecasting (WRF) model. Five plants with the highest rated production have been selected, obtaining the time series of wind speed and production observed every 10 minutes, for a time period of two years (from January 1$^{\text{st}}$, 2005 to December 31$^{\text{st}}$, 2006). Hourly aggregation was performed. The data was not affected by outliers or missing values.

- **LightSource**[3]. Solar energy production data for the year 2017 from 7 plants located in the United Kingdom. Spot values, collected at a time granularity of 1 minute, are aggregated hourly.

For all the datasets the following input features are represented: latitude, longitude of the *i*-th plant; day and hour, respectively; altitude and azimuth; plant ID; weather parameters, such as ambient temperature, irradiance, pressure, wind speed, wind bearing, humidity, dew point, cloud cover, descriptive weather summary. Weather parameters are either measured (training phase) or forecast (testing phase).

From these features, only altitude, azimuth and weather parameters are the subject of the feature extraction process, since they are time variant data.

Weather data are extracted from Forecast.io (`http://forecast.io/`), the expected altitude and azimuth are extracted from SunPosition (`http://www.susdesign.com/sunposition/index.php`), whereas the expected irradiance (PV Italy and LightSource datasets only) is extracted from PVGIS (`http://re.jrc.ec.europa.eu/pvgis/apps4/pvest.php`). The description of the datasets is summarized in Table 1.

[3] This dataset is not publicly available, even if anonymized, due to legal reasons.

Table 1: Brief description of datasets

Dataset	Plants	Days	Hours	Instances (Hourly)	Instances (Daily)
PV Italy	17	856	19	276488	14552
Wind NREL	5	730	24	87590	3650
LightSource	7	365	19	48545	2555

4.4. Results

The first analysis performed was aimed at investigating the performances of the proposed method, in terms of RMSE and MAE, with different rank values for the Tucker decomposition[4]. The selection of the candidate rank values, has been carried out to extract a feature space that represents a trade-off between accuracy and compactness of the data representation. From the results, shown in Table 2, 3 and 4 it is possible to observe that TUCKER-CLUS accuracy is rather stable w.r.t. different values of rank and different training window sizes.

Additionally, the RMSE and MAE results in Table 2, 3 and 4 show that TUCKER-CLUS clearly outperforms all the other approaches in the majority of cases. To better compare all the approaches globally, we used the corrected Friedman test and the post-hoc Nemenyi test following the indications reported in [21].

From the results, presented in Fig. 8, we can see that TUCKER-CLUS outperforms all other algorithms. By comparing TUCKER-CLUS with CLUS, we can see that they are not statistically different, but CLUS is also similar to AE-CLUS, which is not the case of TUCKER-CLUS. The superiority of TUCKER-CLUS is confirmed by a signed Wilcoxon rank test for all pairwise combinations of methods (see Table 5). Here, we can see that, in terms of RMSE, TUCKER-CLUS significantly outperforms all the competitors, taken independently (including CLUS).

It is noteworthy that the execution of EMD-CLUS with the LightSource dataset (Daily setting) was not successful due to memory overhead. In all other cases where the execution was successful, the results show that decom-

[4] The Tucker decomposition is run using a fixed tolerance on the difference in fit equal to 10^{-4}, the maximum number of iterations set to 50, and different training window sizes, whereas the values of the rank parameter are set equally for all dimensions.

Table 2: Forecasting results with TUCKER-CLUS (RMSE) for the PV Italy dataset, considering different rank values and different training window sizes, and comparison with other algorithms. Best results for each configuration are highlighted in bold.

PV Italy dataset	Hourly							
	Window size (days)							
Method	30		60		90		All	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AE-CLUS (Rank 3)	0.1591	0.0976	0.1573	0.0959	0.1579	0.0964	0.1505	0.0992
AE-CLUS (Rank 4)	0.1442	0.0928	0.1426	0.0919	0.1442	0.0927	0.1417	0.0945
AE-CLUS (Rank 5)	0.1497	0.0959	0.1505	0.0964	0.1556	0.0998	0.1396	0.0922
PCA-CLUS (Rank 3)	0.2902	0.2361	0.2864	0.2309	0.2860	0.2323	0.2478	0.1720
PCA-CLUS (Rank 4)	0.2883	0.2346	0.2793	0.2255	0.2783	0.2268	0.2457	0.1718
PCA-CLUS (Rank 5)	NA	NA	NA	NA	NA	NA	NA	NA
EMD-CLUS	0.2018	0.1622	0.2055	0.1684	0.2089	0.1736	0.2048	0.1748
TUCKER-CLUS (Rank 3)	**0.0913**	**0.0535**	**0.0949**	**0.0562**	**0.0929**	**0.0552**	0.1029	0.0623
TUCKER-CLUS (Rank 4)	0.0937	0.0548	0.0952	0.0566	0.0936	0.0556	**0.1011**	**0.0607**
TUCKER-CLUS (Rank 5)	0.0917	0.0539	0.0958	0.0568	0.0940	0.0557	0.1015	0.0614
LSTM (Full feature set)	0.2403	0.1881	0.2383	0.1852	0.2383	0.1852	0.2340	0.1818
CLUS (Full feature set)	0.1315	0.0807	0.1324	0.0807	0.1322	0.0807	0.1271	0.0838
ARIMA	0.1512	0.0928	0.1687	0.1029	0.2013	0.1229	0.2568	0.1693

PV Italy dataset	Daily							
	Window size (days)							
Method	30		60		90		All	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AE-CLUS (Rank 3)	0.0966	0.0605	0.1003	0.0637	0.1047	0.0677	0.1206	0.0821
AE-CLUS (Rank 4)	**0.0854**	**0.0524**	0.0875	0.0547	0.0925	0.0585	0.1152	0.0771
AE-CLUS (Rank 5)	0.0857	0.0524	0.0881	0.0546	0.0902	0.0566	0.1215	0.0829
PCA-CLUS (Rank 3)	0.1229	0.0784	0.1294	0.0838	0.1369	0.0896	0.1363	0.0914
PCA-CLUS (Rank 4)	0.1242	0.0791	0.1288	0.0835	0.1382	0.0903	0.1365	0.0901
PCA-CLUS (Rank 5)	0.1236	0.0786	0.1290	0.0837	0.1384	0.0905	0.1398	0.0927
EMD-CLUS	0.0974	0.0611	0.1032	0.0660	0.1111	0.0726	0.1226	0.0845
TUCKER-CLUS (Rank 3)	0.0872	0.0526	0.0900	0.0549	**0.0878**	**0.0536**	0.0942	**0.0589**
TUCKER-CLUS (Rank 4)	0.0888	0.0538	0.0892	0.0546	0.0887	0.0544	0.0946	0.0591
TUCKER-CLUS (Rank 5)	0.0870	0.0528	**0.0867**	**0.0529**	0.0897	0.0549	0.0950	0.0594

posing the signal in modes via EMD does not provide advantages in terms of increased accuracy in the forecasting task, in the domain addressed in this study. In fact, the predictive performance of EMD-CLUS models appear sub-optimal compared to TUCKER-CLUS and other methods.

Additional considerations arise from Table 7, which shows the number of winning configurations for all the methods considered. From this table (and from Table 6) we can see that TUCKER-CLUS with *rank*=4 or 5 provides the best results in terms of RMSE. Also this view of the results shows that TUCKER-CLUS is the best performing method.

A summarized view of the number of features extracted and the obtained reduction w.r.t. the complete feature set is shown in Table 8. Obviously, a reduction is expected only in the Daily setting (in TUCKER-CLUS the number of features depends on the *rank*). As for the Hourly setting, even if there is no reduction in the number of features, the new features, obtained

Table 3: Forecasting results with TUCKER-CLUS (RMSE) for the Wind NREL dataset, considering different rank values and different training window sizes, and comparison with other algorithms. Best results for each configuration are highlighted in bold.

Wind NREL dataset	Hourly							
	Window size (days)							
Method	30		60		90		All	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AE-CLUS (Rank 3)	0.3512	0.2876	0.3570	0.2942	0.3630	0.3005	0.3627	0.3193
AE-CLUS (Rank 4)	0.3300	0.2654	0.3339	**0.2685**	0.3330	**0.2675**	0.3381	0.2790
AE-CLUS (Rank 5)	0.3409	0.2769	0.3417	0.2786	0.3459	0.2818	0.3546	0.2865
PCA-CLUS (Rank 3)	0.4366	0.3908	0.4490	0.4041	0.4622	0.4216	0.4706	0.4306
PCA-CLUS (Rank 4)	0.4347	0.3842	0.4471	0.3974	0.4604	0.4149	0.4688	0.4240
PCA-CLUS (Rank 5)	NA	NA	NA	NA	NA	NA	NA	NA
EMD-CLUS	0.3355	0.2919	0.3388	0.2996	0.3485	0.3104	0.3683	0.3366
TUCKER-CLUS (Rank 3)	0.3286	0.2778	0.3324	0.2825	0.3624	0.3114	0.3959	0.3446
TUCKER-CLUS (Rank 4)	**0.3151**	**0.2638**	**0.3314**	0.2811	**0.3269**	0.2756	**0.3223**	**0.2712**
TUCKER-CLUS (Rank 5)	0.3214	0.2714	0.3229	0.2731	0.3312	0.2825	0.3287	0.2777
LSTM (Full feature set)	0.4862	0.4374	0.4862	0.4374	0.4862	0.4374	0.4862	0.4374
CLUS (Full feature set)	0.3501	0.2867	0.3508	0.2891	0.3506	0.2903	0.3412	0.2816
ARIMA	0.3654	0.2992	0.4047	0.3336	0.4220	0.3494	0.4623	0.4070

Wind NREL dataset	Daily							
	Window size (days)							
Method	30		60		90		All	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AE-CLUS (Rank 3)	0.3089	0.2783	0.3066	0.2765	0.3146	0.2848	0.3581	0.3342
AE-CLUS (Rank 4)	0.3084	0.2778	0.3242	0.2931	0.3304	0.3004	0.3589	0.3348
AE-CLUS (Rank 5)	0.3070	0.2766	0.3162	0.2852	0.3199	0.2903	0.3414	0.3125
PCA-CLUS (Rank 3)	0.3440	0.3075	0.3633	0.3273	0.3779	0.3420	0.3948	0.3575
PCA-CLUS (Rank 4)	0.3426	0.3067	0.3628	0.3263	0.3817	0.3454	0.4027	0.3656
PCA-CLUS (Rank 5)	0.3419	0.3051	0.3589	0.3221	0.3788	0.3425	0.4087	0.3713
EMD-CLUS	0.3092	0.2804	0.3204	0.2921	0.3296	0.3016	0.3571	0.3324
TUCKER-CLUS (Rank 3)	0.2975	0.2597	0.3031	0.2670	0.3270	0.2910	0.3443	0.3082
TUCKER-CLUS (Rank 4)	**0.2857**	**0.2490**	**0.2916**	**0.2549**	0.2959	0.2592	0.2921	0.2557
TUCKER-CLUS (Rank 5)	0.2928	0.2555	0.2970	0.2597	**0.2940**	**0.2586**	**0.2882**	**0.2528**
LSTM (Full feature set)	0.3814	0.3192	0.3884	0.3263	0.3884	0.3263	0.3884	0.3263
CLUS (Full feature set)	0.3268	0.2945	0.3240	0.2922	0.3246	0.2939	0.3085	0.2880
ARIMA	0.3654	0.2992	0.4047	0.3336	0.4220	0.3494	0.4623	0.4070

Table 4: Forecasting results with TUCKER-CLUS (RMSE) for the LightSource dataset, considering different rank values and different training window sizes, and comparison with other algorithms. Best results for each configuration are highlighted in bold.

LightSource dataset	Hourly							
	Window size (days)							
Method	30		60		90		All	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AE-CLUS (Rank 3)	0.1254	0.0814	0.1239	0.0797	0.1201	0.0781	0.1313	0.0900
AE-CLUS (Rank 4)	**0.1195**	**0.0762**	0.1178	0.0750	**0.1189**	0.0745	0.1230	0.0804
AE-CLUS (Rank 5)	0.1222	0.0782	0.1211	0.0776	0.1212	0.0774	0.1191	0.0770
PCA-CLUS (Rank 3)	0.2696	0.2061	0.2606	0.1960	0.2775	0.2103	0.2583	0.1943
PCA-CLUS (Rank 4)	0.2690	0.2091	0.2582	0.1966	0.2769	0.2110	0.2676	0.2072
PCA-CLUS (Rank 5)	NA	NA	NA	NA	NA	NA	NA	NA
EMD-CLUS	0.2356	0.1792	0.2440	0.1881	0.2466	0.1880	0.2575	0.1961
TUCKER-CLUS (Rank 3)	0.1223	0.0781	0.1278	0.0832	0.1319	0.0870	0.1205	0.0767
TUCKER-CLUS (Rank 4)	0.1209	0.0768	0.1169	0.0749	0.1196	0.0771	**0.1139**	0.0730
TUCKER-CLUS (Rank 5)	0.1196	**0.0762**	**0.1161**	**0.0743**	0.1225	0.0792	0.1203	0.0774
LSTM (Full feature set)	0.2403	0.1881	0.2383	0.1852	0.2383	0.1852	0.2340	0.1818
CLUS (Full feature set)	0.1219	0.0762	0.1185	0.0733	0.1197	**0.0733**	0.1145	**0.0708**
ARIMA	0.1596	0.1035	0.1729	0.1112	0.2284	0.1484	0.3734	0.2559

LightSource dataset	Daily							
	Window size (days)							
Method	30		60		90		All	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AE-CLUS (Rank 3)	0.1217	0.0811	0.1215	0.0826	0.1251	0.0859	0.1208	0.0774
AE-CLUS (Rank 4)	0.1220	0.0810	0.1221	0.0826	0.1244	0.0849	0.1188	0.0762
AE-CLUS (Rank 5)	0.1191	0.0792	0.1207	0.0814	0.1200	0.0811	0.1261	0.0827
PCA-CLUS (Rank 3)	0.1497	0.1000	0.1582	0.1079	0.1672	0.1156	0.1978	0.1353
PCA-CLUS (Rank 4)	0.1515	0.1014	0.1544	0.1051	0.1687	0.1167	0.1966	0.1343
PCA-CLUS (Rank 5)	0.1504	0.1006	0.1542	0.1050	0.1676	0.1159	0.1916	0.1308
EMD-CLUS	NA	NA	NA	NA	NA	NA	NA	NA
TUCKER-CLUS (Rank 3)	0.1273	0.0847	0.1204	0.0810	0.1244	0.0849	0.1183	0.0782
TUCKER-CLUS (Rank 4)	0.1198	0.0789	0.1146	0.0761	0.1173	0.0785	0.1174	0.0774
TUCKER-CLUS (Rank 5)	0.1172	0.0772	0.1132	0.0751	0.1210	0.0807	**0.1119**	0.0776
LSTM (Full feature set)	0.2291	0.1817	0.2291	0.1817	0.2291	0.1817	0.2291	0.1817
CLUS (Full feature set)	**0.1115**	**0.0732**	**0.1112**	**0.0735**	**0.1159**	**0.0773**	0.1122	**0.0739**
ARIMA	0.1596	0.1035	0.1729	0.1112	0.2284	0.1484	0.3734	0.2559

Table 5: p-values of the signed Wilcoxon rank tests for all pairwise combinations of methods. In bold statistically significant values (confidence=0.01, unless specified otherwise).

Pairwise comparison	p-value	winner
RMSE criterion		
TUCKER-CLUS VS LSTM	**1.66E-13**	**TUCKER-CLUS**
TUCKER-CLUS VS ARIMA	**1.66E-13**	**TUCKER-CLUS**
TUCKER-CLUS VS CLUS	**0.001204**	**TUCKER-CLUS**
TUCKER-CLUS VS PCA-CLUS	**1.66E-13**	**TUCKER-CLUS**
TUCKER-CLUS VS AE-CLUS	**3.44E-09**	**TUCKER-CLUS**
TUCKER-CLUS VS EMD-CLUS	**7.71E-13**	**TUCKER-CLUS**
MDLP criterion		
TUCKER-CLUS VS CLUS	**2.88E-11**	**TUCKER-CLUS**
TUCKER-CLUS VS PCA-CLUS	**1.66E-13**	**TUCKER-CLUS**
TUCKER-CLUS VS AE-CLUS	**2.81E-08**	**TUCKER-CLUS**
TUCKER-CLUS VS EMD-CLUS	**1.96E-13**	**TUCKER-CLUS**

Table 6: RMSE reduction (%) with respect to ARIMA and MDLP improvement (%) with respect to CLUS.

Method	RMSE reduction w.r.t. ARIMA	MDLP improvement w.r.t. CLUS
LSTM	6.76%	
CLUS	42.08%	
PCA-CLUS (Rank 3)	8.54%	-108.59%
PCA-CLUS (Rank 4)	8.67%	-106.86%
PCA-CLUS (Rank 5)	9.50%	-1.06%
AE-CLUS (Rank 3)	37.53%	-8.08%
AE-CLUS (Rank 4)	40.09%	6.69%
AE-CLUS (Rank 5)	39.55%	0.03%
EMD-CLUS	28.44%	-68.77%
TUCKER-CLUS (Rank 3)	**43.18%**	**25.59%**
TUCKER-CLUS (Rank 4)	**46.25%**	**25.65%**
TUCKER-CLUS (Rank 5)	**46.15%**	**26.00%**

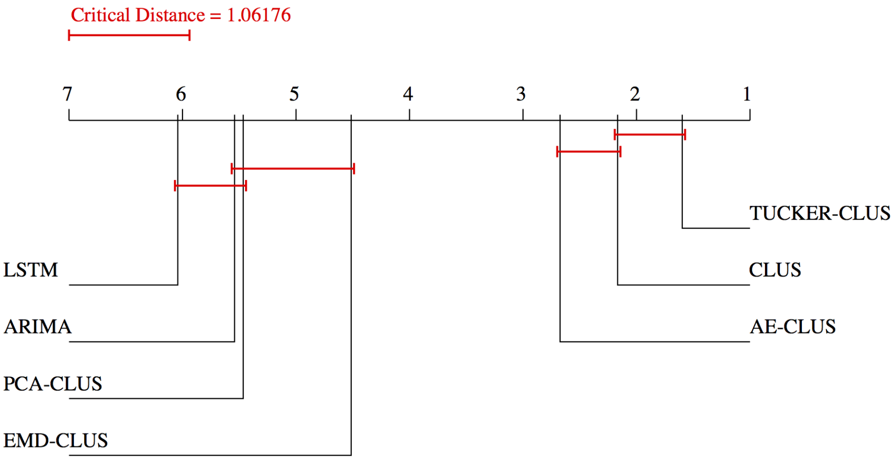


Figure 8: Nemenyi test considering all datasets (RMSE criterion). The algorithms positioned at the rightmost side are the best performing.

Table 7: Number and percentage of winning configurations for all methods.

Method	Rank 3	Rank 4	Rank 5	Overall
EMD-CLUS	0/24 (0.00%)	0/24 (0.00%)	0/24 (0.00%)	0/72 (0.00%)
AE-CLUS	0/24 (0.00%)	4/24 (16.66%)	1/24 (4.16%)	5/72 (6.9%)
PCA-CLUS	0/24 (0.00%)	0/24 (0.00%)	0/24 (0.00%)	0/72 (0.00%)
TUCKER-CLUS	10/24 (41.66%)	**15/24 (65.50%)**	**17/24 (70.83%)**	**42/72 (58.33%)**
ARIMA	0/24 (0.00%)	0/24 (0.00%)	0/24 (0.00%)	0/72 (0.00%)
LSTM	0/24 (0.00%)	0/24 (0.00%)	0/24 (0.00%)	0/72 (0.00%)
CLUS	**14/24 (58.33%)**	5/24 (20.83%)	6/24 (25%)	25/72 (34.72%)

Table 8: Feature set size achieved with different rank values and achieved percentage of reduction w.r.t. the initial feature set

PV Italy	Initial Feature Set Size	Extracted Feature Set Size		
		Rank 3	Rank 4	Rank 5
Hourly	15	12 (20.00%)	15 (0.00%)	18 (-20.00%)
Daily	193	9 (95.33%)	11 (94.30%)	13 (93.26%)

Wind NREL	Initial Feature Set Size	Extracted Feature Set Size		
		Rank 3	Rank 4	Rank 5
Hourly	12	12 (0.00%)	15 (-25.00%)	18 (-50.00%)
Daily	172	9 (94.76%)	11 (93.60%)	13 (92.44%)

LightSource	Initial Feature Set Size	Extracted Feature Set Size		
		Rank 3	Rank 4	Rank 5
Hourly	15	12 (20.00%)	15 (0.00%)	18 (-20.00%)
Daily	193	9 (95.33%)	11 (94.30%)	13 (93.26%)

Table 9: Distribution of prediction errors for all methods and datasets.

	PV Italy		Wind NREL		LightSource	
Method	**Mean**	**Variance**	**Mean**	**Variance**	**Mean**	**Variance**
LSTM	-0.059	0.061	-0.122	0.120	-0.161	0.100
ARIMA	0.046	0.043	-0.044	0.251	0.035	0.039
CLUS	0.030	0.025	-0.172	0.116	-0.009	0.033
PCA-CLUS	0.011	0.046	-0.018	0.088	0.001	0.053
AE-CLUS	0.010	0.039	-0.020	0.074	-0.001	0.038
EMD-CLUS	0.004	0.033	0.001	0.131	-0.001	0.038
TUCKER-CLUS	-0.009	0.028	-0.029	0.071	-0.008	0.030

after the Tucker decomposition, are as much as possible orthogonal and the learning phase does not suffer from collinearity problems.

A better perspective on this aspect is provided by the results in terms of the MLDP measure, which takes into account model complexity and error reduction. In Table 5 we show the results of the signed-rank Wilcoxon test for all pairwise comparisons of algorithms. As we can see, TUCKER-CLUS always outperforms other methods, and the improvement is always statistically significant. This is also confirmed by the corrected Friedman with the post-hoc Nemenyi test that we used to globally compare all the methods (see Fig. 9).

By comparing the different values of the *rank* parameter (see Table 6), we can see that the best results in terms of MDLP are obtained with *rank*=5, although there is no clear difference among the three values.

Another aspect that is worth investigating is the distribution of the error in the predictions. As we can see from Table 9, TUCKER-CLUS exhibits very small variance in the errors compared to the other methods. This indicates that TUCKER-CLUS does not only generate more accurate and simpler

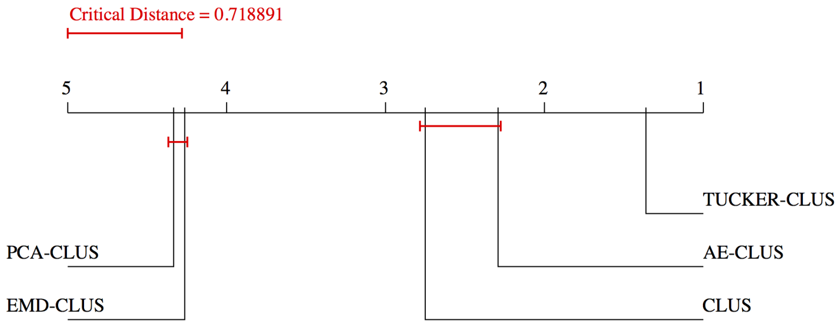


Figure 9: Nemenyi test considering all datasets (MDLP criterion). The algorithms positioned at the rightmost side are the best performing.

models, but also leads to more reliable and unbiased prediction models. A more detailed analysis is shown in Fig. 10, 11 and 12. The error distribution histograms reveal that, although all the methods show a Gaussian-like distribution of the error, TUCKER-CLUS presents a better balancing between the two tails and a smaller dispersion.

In short, from the results, it is clear that considering any learning variant (Hourly and Daily), any error measure (RMSE and MDLP) and any dataset, TUCKER-CLUS is globally the best performing method in terms of accuracy, simplicity and reliability of the prediction models.

Finally, Fig. 13 shows the average execution time required by all the methods analyzed. Overall, it is possible to observe that TUCKER-CLUS exhibits a higher running time than basic auto-regressive algorithms such as ARIMA, but its execution time is comparable (see the running time for the landmark window model) to that of other approaches, which do not reach the same accuracy.

4.5. Availability

The system and the datasets are publicly available to replicate the experiments at the following URL: `http://www.di.uniba.it/~corizzo/tucker-clus/`.

5. Conclusion

In this paper we have proposed a new renewable energy forecasting approach which exploits tensor factorization as a feature extraction technique. Extracted features are used to learn predictive models for hour-by-hour one-day-ahead energy produced by multiple renewable energy plants. The proposed approach appears to be suited for the specific task in hand, mainly

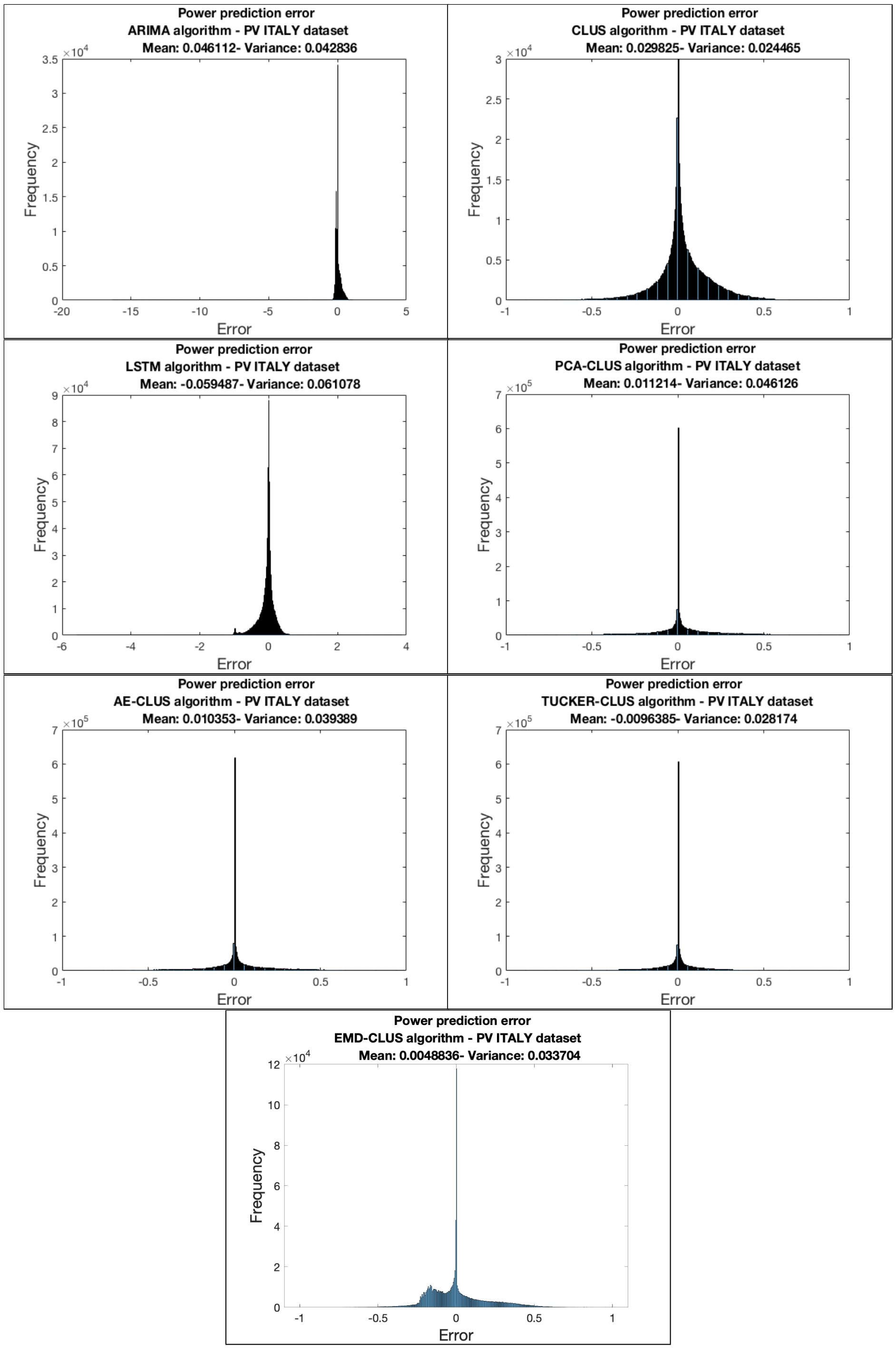


Figure 10: Error histograms for all methods - PV Italy dataset

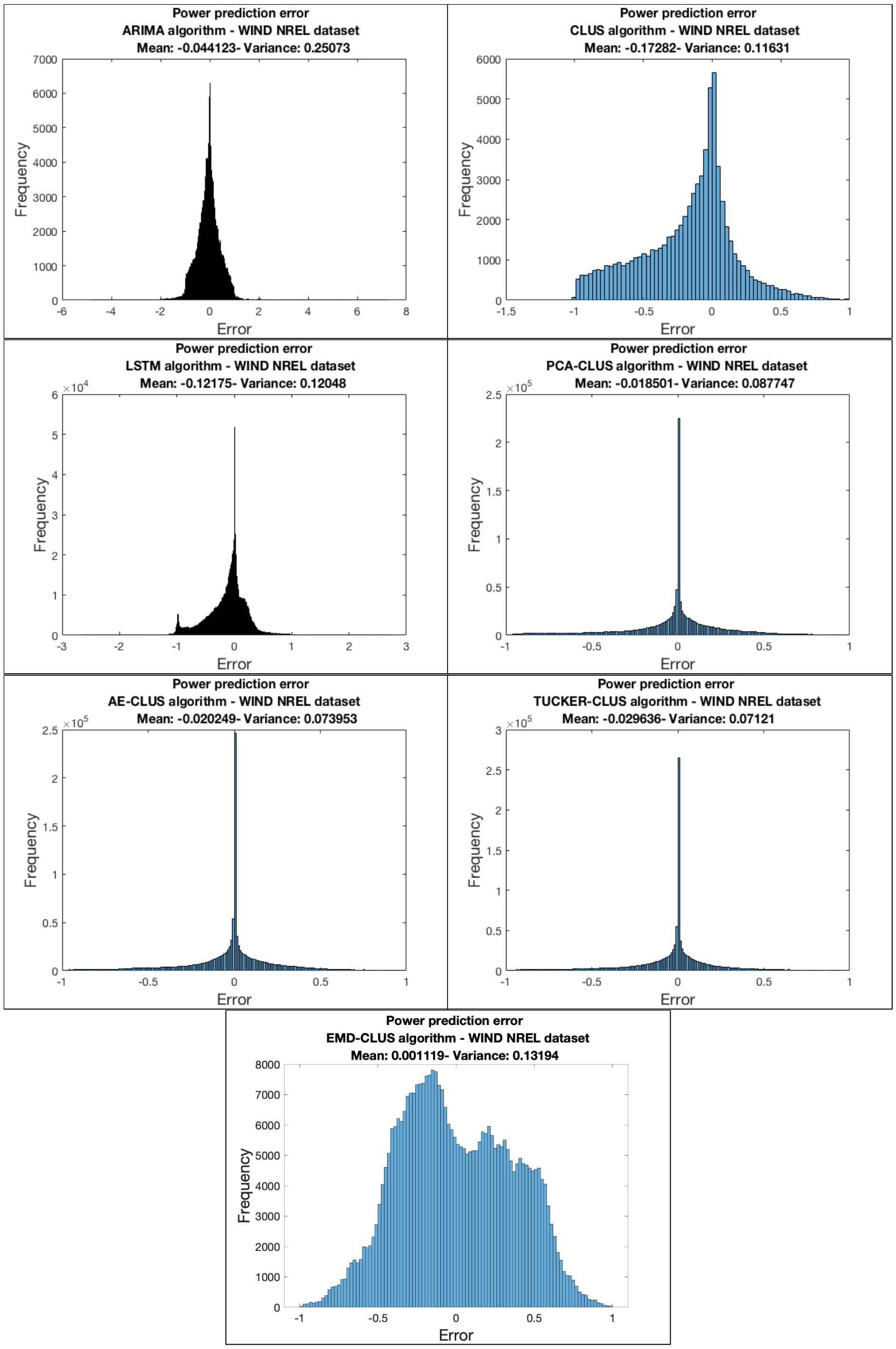


Figure 11: Error histograms for all methods - Wind NREL dataset

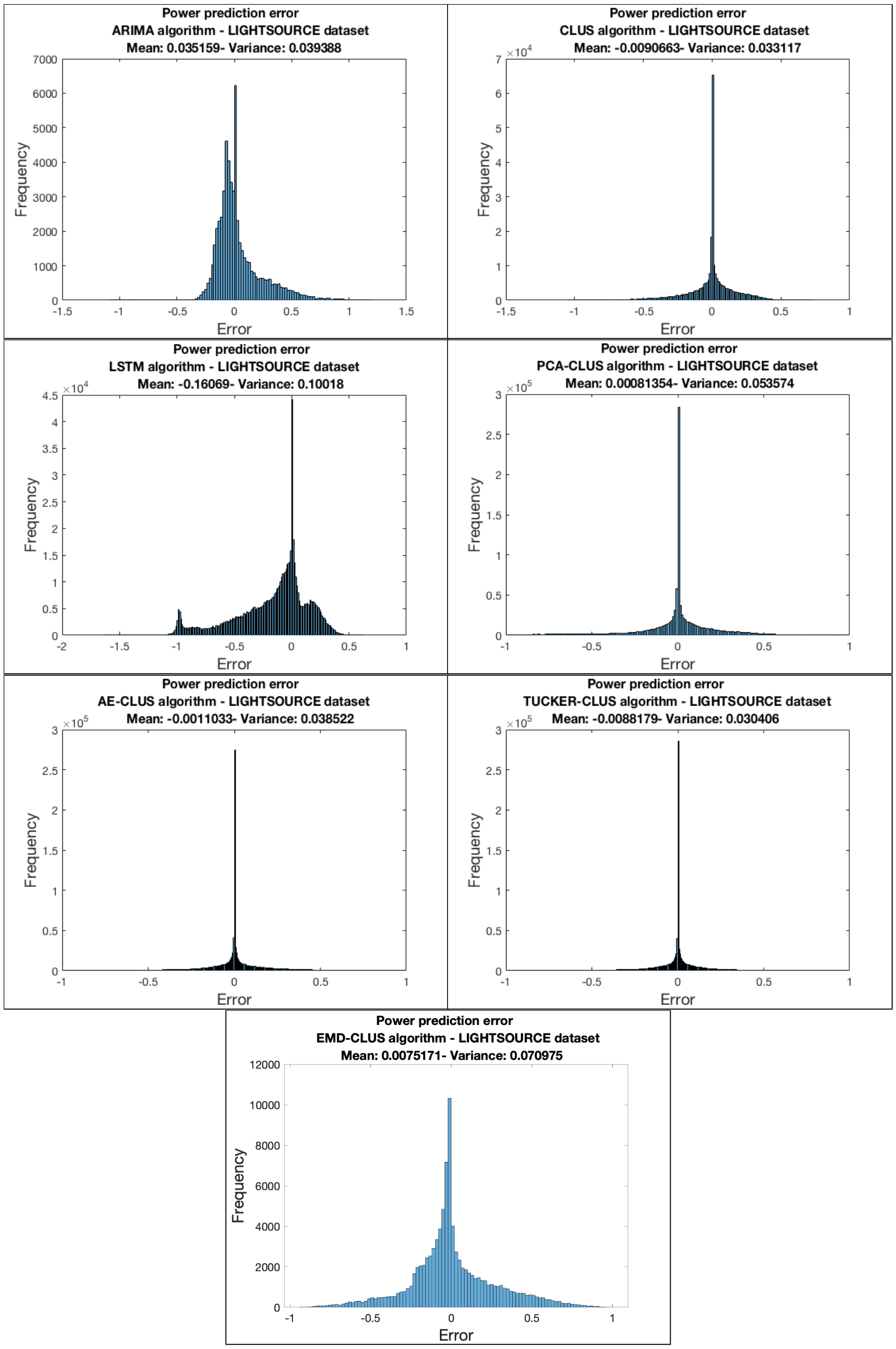


Figure 12: Error histograms for all methods - LightSource dataset

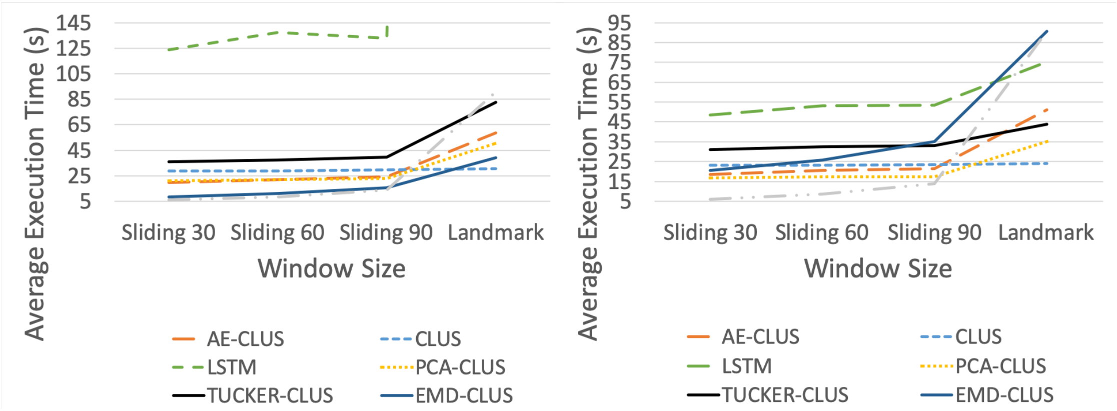


Figure 13: Average execution time in seconds of all methods for the Hourly (left) and the Daily (right) variants. Results for AE-CLUS, PCA-CLUS and TUCKER-CLUS are averaged over the different values of rank (3,4,5).

because of the multi-dimensional (or multi-way) nature of the data. Two variants of the proposed approach have been investigated: the first aims at generating features for predicting the energy produced at a specific hour, and the second aims at generating features for predicting the hour-by-hour time series of the energy produced on a specific day.

An extensive empirical evaluation has been performed on three renewable energy datasets which differ among each other in their size (number of examples), the number of plants, the characteristics of the geographical distribution of the plants, etc. The results obtained with the proposed approach have been compared with state-of-the-art algorithms, and statistical tests have been performed to validate the comparisons between the different methods. They show that the proposed approach globally outperforms competitors in terms of accuracy, especially when predicting time series. The reason is twofold: *1)* a reduced set of features (i.e., a simpler model), with the effect of reducing possible problems due to overfitting and *2)* a mitigation of the collinearity problem, with the result of increasing the effectiveness of predictive models and reducing possible biases in the error distribution.

We identify two potential limitations of the proposed approach. One limitation is the rank estimation problem in the Tucker decomposition. Since the tensor decomposition is applied in an automated setting, a proper estimation of the rank is important. In our experiments, we show that the performance of the proposed method is rather stable with different rank values. Moreover, this aspect can be easily tackled by performing recurrently a grid search on a plausible set of rank values over a restricted validation dataset of recent observations.

Another potential limitation is the presence of noise in historical data used in the tensor data model. In fact, it is possible that anomalies and missing values could affect the ability of our method of extracting high-quality features that are subsequently used to train machine learning models. In our experiments, we had no evidence of this phenomenon, since our data was not affected by such issues.

As future work, we will investigate these issues, and exploit distributed approaches for tensor factorization in order to support the analysis of large-scale streaming data in a cluster computing environment.

Acknowledgment

We acknowledge the support of the Ministry of Education, Universities and Research (MIUR) through the project ComESto - Community Energy Storage: Gestione Aggregata di Sistemi dAccumulo dellEnergia in Power Cloud (Grant No. ARS01_01259) and the PON Ricerca e Innovazione 2014–2020 project "CLOSE – Close to the Earth" (ARS01_001413).

References

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

[2] Bader, B. W., Kolda, T. G., et al. (2015). Matlab tensor toolbox version 2.6.

[3] Bedi, J. and Toshniwal, D. (2018). Empirical mode decomposition based deep learning for electricity demand forecasting. *IEEE Access*, 6:49144–49156.

[4] Belsley, D. A., Kuh, E., and Welsch, R. E. (2005). Regression diagnostics: Identifying influential data and sources of collinearity.

[5] Bessa, R., Miranda, V., and Gama, J. (2009). Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting. *Power Systems, IEEE Transactions on*, 24(4):1657–1666.

[6] Bessa, R. J., Trindade, A., and Miranda, V. (2015a). Spatial-temporal solar power forecasting for smart grids. *IEEE Trans. Industrial Informatics*, 11(1):232–241.

[7] Bessa, R. J., Trindade, A., and Miranda, V. (2015b). Spatial-temporal solar power forecasting for smart grids. *IEEE Transactions on Industrial Informatics*, 11(1):232–241.

[8] Bofinger, S. and Heilscher, G. (2006). Solar electricity forecast - approaches and first results. In *20th Europ. PV conf.*

[9] Bogorny, V., Valiati, J., Camargo, S., Engel, P., Kuijpers, B., and Alvares, L. O. (2006). Mining maximal generalized frequent geographic patterns with knowledge constraints. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 813–817. IEEE.

[10] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control.* John Wiley & Sons.

[11] Buhan, S. and Cadirci, I. (2015). Multistage wind-electric power forecast by using a combination of advanced statistical methods. *IEEE Trans. Industrial Informatics*, 11(5):1231–1242.

[12] Cavalcante, L., Bessa, R. J., Reis, M., and Browell, J. (2017). Lasso vector autoregression structures for very short-term wind power forecasting. *Wind Energy*, 20(4):657–675.

[13] Ceci, M. and Appice, A. (2006). Spatial associative classification: propositional vs structural approach. *Journal of Intelligent Information Systems*, 27(3):191–213.

[14] Ceci, M., Corizzo, R., Fumarola, F., Malerba, D., and Rashkovska, A. (2016). Predictive modeling of pv energy production: How to set up the learning task for a better prediction? *IEEE Transactions on Industrial Informatics*, PP(99):1–1.

[15] Ceci, M., Corizzo, R., Malerba, D., and Rashkovska, A. (2019). Spatial autocorrelation and entropy for renewable energy forecasting. *Data Mining and Knowledge Discovery.*

[16] Chakraborty, P., Marwah, M., Arlitt, M. F., and Ramakrishnan, N. (2012). Fine-grained photovoltaic output prediction using a bayesian ensemble. In *AAAI*, pages 274–280.

[17] Cong, F., Phan, A.-H., Astikainen, P., Zhao, Q., Wu, Q., Hietanen, J. K., Ristaniemi, T., and Cichocki, A. (2013). Multi-domain feature extraction for small event-related potentials through nonnegative multi-way array decomposition from low dense array eeg. *International journal of neural systems*, 23(02):1350006.

[18] Corizzo, R., Ceci, M., and Japkowicz, N. (2019a). Anomaly detection and repair for accurate predictions in geo-distributed big data. *Big Data Research*.

[19] Corizzo, R., Ceci, M., Zdravevski, E., and Japkowicz, N. (2020). Scalable auto-encoders for gravitational waves detection from time series data. *Expert Systems with Applications*, 151:113378.

[20] Corizzo, R., Pio, G., Ceci, M., and Malerba, D. (2019b). Dencast: Distributed density-based clustering for multi-target regression. *Springer Journal of Big Data*.

[21] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30.

[22] Dhillon, P. S., Foster, D., and Ungar, L. H. (2011). Minimum description length penalization for group and multi-task sparse learning. *Journal of Machine Learning Research*, 12(Feb):525–564.

[23] Dowell, J. and Pinson, P. (2016). Very-short-term probabilistic wind power forecasts by sparse vector autoregression. *IEEE Transactions on Smart Grid*, 7(2):763–770.

[24] Dragomiretskiy, K. and Zosso, D. (2013). Variational mode decomposition. *IEEE transactions on signal processing*, 62(3):531–544.

[25] Elsner, J. B. and Tsonis, A. A. (2013). *Singular spectrum analysis: a new tool in time series analysis.* Springer Science & Business Media.

[26] Fanaee-T, H. and Gama, J. (2016). Tensor-based anomaly detection: An interdisciplinary survey. *Knowledge-Based Systems*, 98:130–147.

[27] Farrar, D. E. and Glauber, R. R. (1967). Multicollinearity in regression analysis: The problem revisited. *The Review of Economics and Statistics*, 49(1):92–107.

[28] Fotheringham, A. S., Brunsdon, C., and Charlton, M. (2003). *Geographically weighted regression: the analysis of spatially varying relationships.* John Wiley & Sons.

[29] Fujimoto, Y. and Hayashi, Y. (2012). Pattern sequence-based energy demand forecast using photovoltaic energy records. In *2012 International Conference on Renewable Energy Research and Applications (ICRERA)*, pages 1–6.

[30] Gama, J. and Gaber, M. M., editors (2007). *Learning from Data Streams.* Springer.

[31] Gneiting, T., Larson, K., Westrick, K., Genton, M. G., and Aldrich, E. (2006). Calibrated probabilistic forecasting at the stateline wind energy center: The regime-switching space–time method. *Journal of the American Statistical Association*, 101(475):968–979.

[32] Han, J., Kamber, M., and Pei, J. (2011). *Data Mining: Concepts and Techniques*, pages 113–114. Morgan Kaufmann, 3rd edition.

[33] Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12.

[34] Hinton, G. and Salakhutdinov, R. (2006). Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507.

[35] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9:1735–80.

[36] Jolliffe, I. (2002). *Principal component analysis.* Wiley Online Library.

[37] Kocev, D., Ceci, M., and Perdih, T. S. (2020). Ensembles of extremely randomized predictive clustering trees for predicting structured outputs. *Mach. Learn.*, (in press).

[38] Kolda, T. G. and Bader, B. W. (2009). Tensor decompositions and applications. *SIAM review*, 51(3):455–500.

[39] Li, X. and Claramunt, C. (2006). A spatial entropy-based decision tree for classification of geographical information. *Transactions in GIS*, 10(3):451–467.

[40] Malerba, D., Ceci, M., and Appice, A. (2005). Mining model trees from spatial data. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 169–180. Springer.

[41] Malerba, D., Esposito, F., Ceci, M., and Appice, A. (2004). Top-down induction of model trees with regression and splitting nodes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(5):612–625.

[42] Mason, C. H. and Perreault Jr, W. D. (1991). Collinearity, power, and interpretation of multiple regression analysis. *Journal of marketing research*, pages 268–280.

[43] Mathiesen, P. and Kleissl, J. (2011). Evaluation of numerical weather prediction for intra-day solar forecasting in the continental united states. *Solar Energy*, 85(5):967–977.

[44] Panagakis, Y., Kotropoulos, C., and Arce, G. R. (2010). Non-negative multilinear principal component analysis of auditory temporal modulations for music genre classification. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):576–588.

[45] Papalexakis, E. E., Faloutsos, C., and Sidiropoulos, N. D. (2016). Tensors for data mining and data fusion: Models, applications, and scalable algorithms. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):16.

[46] Pelland, S., Galanis, G., and Kallos, G. (2013). Solar and photovoltaic forecasting through post-processing of the global environmental multiscale numerical weather prediction model. *Prog Photovolt Res Appl*, 21(3):284–296.

[47] Phan, A. H. and Cichocki, A. (2010). Tensor decompositions for feature extraction and classification of high dimensional datasets. *Nonlinear Theory and Its Applications, IEICE*, 1(1):37–68.

[48] Prada, M. A., Toivola, J., Kullaa, J., and Hollmén, J. (2012). Three-way analysis of structural health monitoring data. *Neurocomputing*, 80:119–128.

[49] Renard, N. and Bourennane, S. (2009). Dimensionality reduction based on tensor modeling for classification methods. *IEEE Transactions on Geoscience and Remote Sensing*, 47(4):1123–1131.

[50] Rilling, G., Flandrin, P., Goncalves, P., et al. (2003). On empirical mode decomposition and its algorithms. In *IEEE-EURASIP workshop on nonlinear signal and image processing*, volume 3, pages 8–11. NSIP-03, Grado (I).

[51] Rinzivillo, S. and Turini, F. (2007). Knowledge discovery from spatial transactions. *Journal of Intelligent Information Systems*, 28(1):1–22.

[52] Sharma, N., Sharma, P., Irwin, D. E., and Shenoy, P. J. (2011). Predicting solar generation from weather forecasts using machine learning. In *SmartGridComm*, pages 528–533. IEEE.

[53] Stojanova, D., Ceci, M., Appice, A., Malerba, D., and Dzeroski, S. (2013). Dealing with spatial autocorrelation when learning predictive clustering trees. *Ecological Informatics*, 13:22–39.

[54] Tastu, J., Pinson, P., Trombe, P.-J., and Madsen, H. (2014). Probabilistic forecasts of wind power generation accounting for geographically dispersed information. *IEEE Transactions on Smart Grid*, 5(1):480–489.

[55] Tork, H. F., Oliveira, M., Gama, J., Malinowski, S., and Morla, R. (2012). Event and anomaly detection using tucker3 decomposition. In *Workshop on Ubiquitous Data Mining*, page 8.

[56] Zhao, M. and Li, X. (2011). An application of spatial decision tree for classification of air pollution index. In *Geoinformatics, 2011 19th International Conference on*, pages 1–6. IEEE.