energies

MDPI

Article

A Review of the Enabling Methodologies for Knowledge Discovery from Smart Grids Data [†]

Fabrizio De Caro [1,*], Amedeo Andreotti [2], Rodolfo Araneo [3], Massimo Panella [4], Antonello Rosato [4], Alfredo Vaccaro [1] and Domenico Villacci [1]

[1] Deptartment of Engineering, University of Sannio, 82100 Benevento, Italy; vaccaro@unisannio.it (A.V.); villacci@unisannio.it (D.V.)

[2] Electrical Engineering Department, University of Naples Federico II, 80125 Naples, Italy; amedeo.andreotti@unina.it

[3] Electrical Engineering Division of DIAEE, University of Rome "La Sapienza", 00184 Rome, Italy; rodolfo.araneo@uniroma1.it

[4] Deptartment of Information Engineering, Electronics and Telecommunications, University of Rome "La Sapienza", 00184 Rome, Italy; massimo.panella@uniroma1.it (M.P.); antonello.rosato@uniroma1.it (A.R.)

[*] Correspondence: fdecaro@unisannio.it

[†] This paper is an extended version of our paper published in 20th IEEE International Conference on Environment and Electrical Engineering, 9–12 June 2020, Madrid, Spain.

Received: 10 November 2020; Accepted: 8 December 2020; Published: 14 December 2020

Abstract: The large-scale deployment of pervasive sensors and decentralized computing in modern smart grids is expected to exponentially increase the volume of data exchanged by power system applications. In this context, the research for scalable and flexible methodologies aimed at supporting rapid decisions in a data rich, but information limited environment represents a relevant issue to address. To this aim, this paper investigates the role of Knowledge Discovery from massive Datasets in smart grid computing, exploring its various application fields by considering the power system stakeholder available data and knowledge extraction needs. In particular, the aim of this paper is dual. In the first part, the authors summarize the most recent activities developed in this field by the Task Force on "Enabling Paradigms for High-Performance Computing in Wide Area Monitoring Protective and Control Systems" of the IEEE PSOPE Technologies and Innovation Subcommittee. Differently, in the second part, the authors propose the development of a data-driven forecasting methodology, which is modeled by considering the fundamental principles of Knowledge Discovery Process data workflow. Furthermore, the described methodology is applied to solve the load forecasting problem for a complex user case, in order to emphasize the potential role of knowledge discovery in supporting post processing analysis in data-rich environments, as feedback for the improvement of the forecasting performances.

Keywords: smart grids computing; knowledge discovery; power system data compression, high-performance computing

1. Introduction

On-line smart grid operation asks for quickly identifying reliable decisions in a complex data rich, but information-limited domain [1]. In this context, the data streaming generated by the network of pervasive sensors distributed along the entire power system do not always provide smart grids operators with the necessary information to react to external disturbances in the time-frames required to minimize their impacts. Even in the presence of fast computing algorithms aimed at converting data into information, the smart grid operator must face the challenge of not having a clear picture of the information context and, therefore, the obtained information cannot be deployed with any high degree of confidence [2].

To address this complex issue, the most promising research directions are oriented toward the conceptualization of improved information processing paradigms and smart decision support systems aimed at enhancing standard operating procedures. A set of interactive information services, which could promptly provide the right information at the right moment to the right decision maker, are adopted based on predefined grid conditions and static operating thresholds [3]. Advanced techniques and algorithms for reliable power system data acquisition and processing, which should support semantics and content-based data extraction and integration from heterogeneous sensor networks, should be developed in order to effectively support the deployment of these services in modern smart grids [4]. The integration of these computational intelligence-based tools in online smart grid applications, Knowledge Discovery, which is the process of extracting features, complex relationships, and patterns from large heterogeneous datasets generated by pervasive sensor networks and distributed information sources, is one of the most promising enabling methodologies. The large scale deployment of Knowledge Discovery-based techniques allows processing and classifying the large data-streams generated from various sources, which include grid sensors, SCADA systems, and phasor measurement units, in order to build domain specific knowledge, which can be discovered and shared. The latter process allows for converting data into information, and actionable intelligence at different application domains. To this aim, novel computing algorithms aimed at providing interactive tools for data management, storage, compression, and inference are necessary in order to enable resource discovery, generating semantic metadata.

Armed with such a vision, in [5], formal methods for knowledge discovery from a large quantity of data aimed at reducing the complexity of optimal power flow problems have been conceptualized. The proposed knowledge-based paradigms allow for extracting complex features, hidden relationships, and useful hypotheses characterizing the regularities in the optimal power flow solutions from historical power system operation data. The actionable intelligence extracted by these paradigms is then used to formalize the problem into a transformed domain, where the problem equations can be solved more effectively because the problem cardinality is sensibly reduced, and the corresponding solutions can be obtained more efficiently.

In [6], a big-data visualization platform for knowledge discovery from massive smart grid data has been applied in the task of solving solving operation, control, and situation awareness problems. To discover the hidden knowledge from the large volume of heterogeneous data streaming, an open source cluster computing framework has been designed, and a high-speed communication architecture, which is based on the Open System Interconnection model, has been designed to visualize and present the data to the operators.

The role of knowledge discovery processes in real-time stability monitoring, online control, proactive operation, and optimal planning of modern smart grids has been outlined in [7]. In particular, starting from the analysis of the new challenges induced by the emerging elements characterizing modern power system operation, this research work outlines the inadequacy of conventional analysis tools in effectively addressing the main system operation problems, identifying the cutting-edge computational intelligence techniques, and their potential role in solving these problems.

The important role that computational intelligence-based techniques can play in knowledge discovery from smart grids data has been confirmed in [8], which conceptualizes a holistic distributed stream clustering for decision support and data analytic in user-centric power systems. This holistic clustering paradigm could be used to effectively solving several mart grid intelligent layer research problems, as far as contingency analysis, asset management, and dynamic energy pricing are concerned.

Although the effectiveness of these knowledge discovery-based paradigms have been successfully assessed in the task of solving specific smart grid problems, their global integration in realistic decision support systems requires the development of ontology middleware, which provides functionalities aimed at facilitating operational data acquisition and handling in interoperable formats, enabling information services through a coordinated process chain [9]. These functions can be obtained by processing heterogeneous smart grids data-sets by ontology-based techniques, and smart reasoning system, which enable access to the information content rather than keyword-based searches. This paradigm allows for accomplishing knowledge discovery, providing decision support to smart grid operators by focusing on making computing systems more closely interact at human conceptual levels, modeling the semantics of the data, instead of just relying on the syntactic and structural representations. These features allow the ontology middleware to become a flexible and extendable platform for knowledge management solutions in smart grids.

According to the research directions identified by these papers, the Task Force on "Enabling Paradigms for High-Performance Computing in Wide Area Monitoring Protective and Control Systems" of the IEEE PSOPE Technologies and Innovation Subcommittee analyzed the open problems, the challenging issues, and the most promising enabling technologies for knowledge discovery from smart grids data. The main results of this analysis are analyzed in this paper, and the experimental results obtained on an complex case study are presented and discussed in order to emphasize the potential role of computational and cognitive techniques for situation awareness in smart grid applications.

2. Knowledge Discovery from Massive Data

The recent technological advancements in data storing and processing allow the growth of electronic archives, coupled to a large and pervasive diffusion of online sensors, which transmit high frequency information streams about the operation states of complex and distributed systems [10]. Online processing of these massive data allows for improving the knowledge about the behavior of complex systems characterized by large uncertainty sources, which make the deterministic modeling of the analyzed system difficult [11].

Unfortunately, the massive increase of data volume has deteriorated the effectiveness of the traditional approaches employed to extract useful information. Indeed, a large amount of data is not guaranteed to be a reliable source of information, but in the majority of cases, the data need to be processed to reveal their true intrinsic knowledge value [12]. Furthermore, the process of acquisition and storing data are related to a certain cost in terms of equipment and storage technologies. For this reason, the extraction of the most profitable information from them is playing a strategic role in modern complex systems analysis [13]. In this context, the main objective of the data analyst is to develop strategies aimed at giving value to this process, promoting reliable software and hardware architecture able to effectively perform this task.

For this reason, when the cardinality of heterogeneous data becomes too large for a complete human management or traditional approaches, it is time for Artificial Intelligence (AI) to support analysts in extrapolating reliable and useful information [14]. In this domain, Knowledge Discovery in large Database (KDD) represents a strategic solution as it allows the identification of *valid, novel, potentially useful, and ultimately understandable patterns in data* [15]. Valid, novel and potentially useful data or anything, such as models or relation, represent an added value with respect to a certain aspect. Finding prediction models or deeper insights about an economy or product system that allow a better management of them are explicative examples.

This research is necessary because large datasets cannot be understood immediately, containing more information than they appear to have. Trends, regularities, and patterns can be revealed only after a complex procedure of data processing. In particular, the KDD process is an activity made of different interaction and retrieval steps, which requires the human action in certain phases. The process is commonly confused with Data Mining, which is one of the KDD steps. The interactivity of the KDD process is related to the crucial role played by humans in supervision and validation of the discovered information. Its contribution is related both to its expertise with mining tools and its knowledge domain, which means the ability to exploit the understanding of data to filter knowledge from irrelevant and incorrect data [15].

In particular, according to [15], the main steps composing the KDD process are:

- Definition of the KDD process goal from the customer point of view. Understanding of the domain and of the a priori knowledge;
- Selection of the target data from the available ones performing the KDD process;
- Data cleaning and preprocessing: it includes the basic operation of noise removal, the collecting and merging procedures of samples, and the accounting of date and time information;
- Data reduction and projection: the features of the samples are processed by adopting cardinality reduction or feature selection techniques aimed at either reducing the set of data to the most relevant feature or finding invariant transformation of data;
- Goal matching of the KDD process to the choice of a particular data mining methods (e.g., clustering, regression, classification, etc.);
- Data mining algorithm selection to find patterns in the data in consideration of the goal and data available;
- Performing the data mining algorithm to search for patterns in data;
- Mined pattern interpretation, it involves the possibility to visualize the results of mining and coming back to the previous steps to adjust patterns or select a different algorithm to improve the results;
- Knowledge consolidation, it consists of processing data in the most suitable form for either successive KDD processees or visual report generation for the customer.

The data mining step is the core of a KDD process involving a repeated iteration of data mining algorithms, where the kind of applied algorithm depends on the goals to pursue, where the latter can be classified as verification and discovery. When the objective is simply validating the user hypothesis, the goal is called 'verification', whereas, when it is necessary that the developed system will find new patterns, the goal is called 'discovery'. Furthermore, prediction refers to the following data mining tasks:

- Prediction: the goal is the patterns development for the prediction of the behavior of certain features given a forecasting horizon;
- Description: the goal is the patterns development aimed at presenting data in a more understandable form.

Nevertheless, the described classification between the possible goals of data mining the boundary between them is not sharp. Indeed, the description models could be employed also for predicting further classification and vice versa. The data mining methods range between a wide spectrum of techniques, where the employment of one or more methods depends on the considered objective. The canonical classification considers the following family of methods [15]:

- Classification: learning a function that maps a data to a certain class;
- Regression: learning a function that finds a relation between an observed set of input–output data discovering possible functional relations;
- Clustering: grouping data in a given set based on their similarity, by identifying samples (or patterns) with similar features;
- Summarizing: finding compact representation of multi-variate data;

- Dependency Modeling: learning model describing the dependencies; between variables in probabilistic and graphical terms;
- Change and Deviation Detection: learning model to find differences or strong deviation measured in a flow process.

The outlined KDD process goals can be reached by the construction of specific algorithms, which are characterized by a large variety of typologies, all decomposable in three key concepts [15]:

- model representation;
- model evaluation criteria;
- search method.

In this case, model representation is the employed language to describe discoverable patterns and it includes the data analyst knowledge about the assumption done related to the application of a certain model. This is fundamental because too simplistic hypothesis about the process to study will lead to poor results independently from the amount of data and training time.

The model evaluation criteria are the quantitative representation of how well a discovered pattern meets the goal of the KDD process, where the case of predictive models is limited to evaluating the accuracy of the estimated quantities with respect to the observed ones for each case. In the case of descriptive models, the evaluation concerns assessment on the novelty, utility, and understandability of the fitted model. Finally, once the models are selected and evaluation criteria fixed, the search method is aimed at finding the parameters/family of models, maximizing the fixed objectives, and reducing the task to an optimization problem. In particular, the employed data mining methods are classifiable in the following families [15]:

- Decision Trees and Rules;
- Nonlinear Regression and Classification methods;
- Data-driven models;
- Probabilistic Graphic Dependency models;
- Relation Learning Models.

Decision Trees are one of the most common methods employed in data mining for classification [16]. The goal of the method is to train a model for assigning a class to a sample by considering the values of its features. The model is based on the partitioning of the domain in sub-domains by applying tree branching. The process is extended to the class of regression problems when the values domain lies in that of real numbers where the methods are called 'regression trees' [17]. Nonlinear regression is instead based on developing predictive models, which combine basic functions, such as polynomial, sigmoid, and spline [18]. The polynomial regression is one of the simplest approaches, and it aims at fitting a model by using curves of order $n > 2$ (quadratic, cubic, etc.), while the spline approach aims at producing a piecewise model in which each model is trained with only the value lying in a specified interval.

Artificial Neural Network (ANN) is the most representative class in the data-driven learning domain [19]. ANNs are based on parametric regression and classification models whose structure imitates the behavior and the topology of biological nervous systems, in particular their connections, and where parameters are estimated in a supervised fashion by means of input–output examples of the task to be accomplished. In early traditional ANNs, the number of layers is limited and they are also called *shallow* neural networks. ANNs can also be used in combination with fuzzy logic to implement fuzzy neural networks that are able to deal with the uncertainty of data more naturally [20]. Some recent studies considered the application of such networks to the prediction of load forecasting [21], where the robustness of fuzzy logic to handle noisy and unreliable measures is exploited with the characteristic of ANNs to learn by means of numerical examples rather than by linguistic rules (as in the case of general fuzzy inference systems).

As an extension of shallow ANNs, *deep* ANNs have been proposed in the context of deep learning, which finds a large application in solving complex classification tasks typically involving a huge amount of data as in the case of image-based datasets and information processing [22]. Here, the word 'deep' stands for emphasizing the learning process based on successive layer representation of data. In most cases, the data transformation consists of hundreds of successive representation layers [23]. The enormous data size increment has pushed the emerging of deep learning algorithms and architectures in many power system applications. The most common architectures employed in the deep learning field is the Convolutional Neural Network (CNN) [24], which has shown great capability to deal with large spatial data. Many developed libraries, such as TensorFlow, Torch/PyTorch, and Theano, have been developed for several programming languages, allowing a reliable application of deep learning for their specific needs on CPU/GPU architectures. CNNs are largely employed in computer vision and for dealing with data having spatial relationships. The name derives from the convolution mathematical operation, which is employed in specific *convolutional layers*. The data processing in a CNN aimed at extracting progressively features from sub-samples of the original data, which have to be arranged in an input tensor. According to their capability, they have been strongly employed in spatial load forecasting applications such as in [25].

A special kind of ANN is the Recurrent Neural Network (RNN), which is capable of keeping the memory of the past in an internal state while it incrementally processes data; for this reason, RNN has a big potential for managing time series. It was developed based on the [26] proposal in the framework of 'Reservoir Computing', acquiring even more consideration in speech and text recognition due its capability to consider all the dynamic process under study. In a basic RNN architecture, the output is generated by a combination between the input data and a recurrent correlation. An RNN can be equivalently considered as many feed-forward ANNs operating sequentially to supply outputs over the time sequence to predict. Starting from randomized versions of shallow ANN architectures, as in the case of the Echo State Network (ESN) [27], over the years, several advancements have been developed in order to overcome the RNN unit limits in the deep learning field. The Long Short-Term Memory (LSTM) network is the most popular approach to this end [28]. It is based on computational units whose basic structure is composed by a cell, which keeps the memory in the unit, and three regulators or gates, which manage the information flow inside the sequential units. They are called input, output, and forget gate, but they are not present in all architectures. LSTSM is particularly suited to deal with the vanishing of gradient, a typical problem of deep learning [29]. Furthermore, another type of RNN unit, called a Gated Recurrent Unit (GRU) unit, has been developed in order to avoid overfitting issues, by increasing the forecasting accuracy as shown in [30].

Among data-driven approaches to solve the regression/classification task, there are also nonparametric models based, for instance, on Case Base Reasoning [31] and Nearest Neighbors regression or classification [32]. One of the main critical issues in this kind of application is adopting a well-defined metric for weighting the similarity between the stored examples with respect to the query sample properties. Because of the increase in the amount of the databases' cardinality, these kinds of methods often also consider the support of techniques for cardinality reduction to avoid the so-called *curse of dimensionality* [33].

Probabilistic Graph models are employed for characterizing the dependency between variables, where the variable dependencies are taken into account via graph structure. This approach has been initially employed by considering categorical discrete variables, for it then to be successively extended to continuous variables with Gaussian density. One of the most employed models is that, based on Bayesian networks, where the graphical relation between variables is expressed in the form of conditional probabilities, which can be assigned by the expert system, or by applying inference procedures, by learning the parameters from the observed data [34]. Finally, the Relation Learning Models combines machine learning with the logic of first order, defining the *Inductive Logic Programming*. It is a form of investigation aimed at finding patterns and discovering insights in data. It is based on the employment of clausal first order logic as a representation language for both data and hypothesis [35].

Research and Application Challenges in Smart Grids

The KDD process allows for harnessing the effectiveness of big data in power systems for a large number of research fields, where the possible applications range over the entire chain of power electric infrastructure. In particular, the big data employment in power systems can be seen from a holistic point of view, where the improvements produced by the discovered knowledge for each component of the system allow for improving the reliability and flexibility of the overall system [36]. The main data stream in power system operation is generated by Supervisory Control and Data Acquisition (SCADA), Phasor Measurement Units (PMU), and Advanced Metering Interface (AMI) [10]. The SCADA system is widely spread in power stations and power grids (transmission and distribution) and its measurement frequency is on the order of few seconds. The system collects a wide range of variables depending on the monitored system type.

The PMU is a measurement device operating at higher sampling frequency (30–60 measurements per second), which allows for acquiring the voltage and current phasors synchronized with a common time reference (e.g., provided by a Global Positioning System). These devices are mainly deployed in transmission networks, where they represent the backbone of the WAMSs (Wide Area Monitoring Systems) [37], and, more recently, in active distribution networks, where they are typically referred as "micro-PMU" [38]. Moreover, AMI is a system interacting with multiple metering sources (electric, heat, gas), which allows for collecting multiple heterogeneous variables in distribution networks. It is one of the most promising enabling technologies for demand response-based frameworks by allowing interaction with home devices, and IoT-based sensors [39].

The availability of different data sources, which characterize different subsystems in power grids, causes a deep heterogeneity in the corresponding data streams. In particular, the latter can be classified as:

- Raw waveform data (voltage and currents, exchanged active, reactive power at bus, conductor temperature, etc.);
- Preprocessed waveforms (voltage and currents, weather parameters over the grid);
- Status variables of system components;
- Consumer consumption/distributed generation data;
- Power Plants operation and energy bidding data;
- Electricity Market data.

To extract actionable information from this large set of heterogeneous data, many papers outline the potential role of big-data based knowledge discovery in solving several power system operation problems. Predictive maintenance, process and control optimization, analysis, and prediction of the electricity-market prices have been solved by recurring to the KDD process. Furthermore, the spreading of Variable Renewable Energy (VRE) power plants has extended the application of KDD in time and spatial prediction of the wind/solar power profile for several forecasting horizons [40–42]. In addition, the harness of visualization and data description in KDD process allows for introducing advanced and exhaustive analysis of the forecasting performance, by adopting a rigorous comparison of metric and statistical tests for accuracy and performance analysis.

The enhancement of accuracy in VREs forecasting is a clear example of the previously described holistic approach, where a reduction in uncertainty in the power generation amount leads to benefits for all systems, by reducing the cost related to the reserve procurement. Furthermore, KDD is useful in the estimation and forecasting of the water amount in hydroelectric power stations. In transmission networks, the role of KDD is related to the detection [43], classification, and analysis of faults, detection of the most sensitive substation to disturbances [44], impact of severe weather events on the network for resilience study, and analysis of conductor temperature for Dynamic Thermal Rating (DTR) application.

Generally, the distribution networks still do not have the same density of installed sensors with respect to the transmission networks. Anyway, the increasing in Distributed Generation (DG) and complex load active in demand response require an effort in the improvement in the communication infrastructure of the distribution grid [45,46]. In this sense, the role of KDD is enabling in extracting precious information on the limited number of data stream available. An example is related to power system state and topology estimation, where the graph configuration of the distribution network is identified by analyzing voltage measure at buses in the presence of radial networks with active connections and switchable root nodes [47].

The KDD process supplies an important support in characterizing the load profiles in distribution grids, especially for those hosting a large capacity of DG [48]. In particular, the Net Load characterization (Demand minus DG) and its forecasting represents one of the greatest challenges in the management of grid flexibility. A large support for the energy consumer profile is supplied by approaching the problem with clustering techniques and auto-correlation analysis [49]. Finally, the KDD process is employed for electricity market analysis by both power generation companies and customers in order to reveal useful insights to be used in developing advance bidding strategies in electricity markets [50].

3. Cardinality Reduction and Data Compression

The large scale diffusion of sensor networks in Smart Grids represents a severe issue to address in data storing and transmission, which affect many online applications, such as load flow studies, state estimation, and contingency analysis. Despite the improvements in data transmission capabilities, these massive amount of data streaming may cause bottlenecks in communication networks, which are not infrequent in Smart Grids where the development of dedicated wide area communication networks is not feasible due to the presence of large dispersed energy resources on both customers and distributed generation side. In this context, the adoption of techniques for reducing the volume of data is crucial to satisfy the time constraints in supplying the required data processing. Clearly, the typology of data compression depends on the specific needs, such as the data type (numerical or categorical variables), if the process is lossy or lossless, etc. [51].

In particular, the reduction process for the data compression can perform on: (i) features; (ii) samples. The compression is performed by aging of the features of the processed dataset. Most employed linear techniques are Factor Analysis (FC) and Principal Component Analysis (PCA), whereas nonlinear approaches include Locally Linear Embedding (LLE), Isomap, and derivatives [51]. The aim of these methods is transforming the original variables in new ones through a combination of them according to the principles of the adopted method, where the result is the reduction of the data cardinality by deleting the most irrelevant or redundant features. Further techniques, such as minimum Redundancy Maximum Relevancy (mRMR) [52], aim to extract a subset of the variables from the original dataset. The extracted variables have the highest mutual dependency with respect to a target in a dataset by using statically information metrics.

Differently, sample reduction involves the following techniques applied to data mining:

- sampling;
- squashing;
- clustering;
- binning.

The data sampling is basically the simplest form of sample reduction because it acts on a naïve extraction from the original dataset of a subset of samples by considering non-complex rules [53]. On the contrary, data squashing produces artificial samples having the same statistical moments characteristics of the original data [54]. Data clustering aims at grouping samples with common features. The number of developed clustering is very wide with effective results in the task of classification. Binning methods consist of transforming a continuous variable in a category where the approach ranges from the naïve method to the statically based.

In this domain, the Principal Component Analysis is one of the most employed methods for linear data reduction [55]. It performs this through an unsupervised process that projects the data from the original space into a lower dimensional one where the axes, called Principal Components (PCs), of this new space are computed by combining the original variables. The first PC is oriented along the direction with the maximum variance of data [18]. This mathematically corresponds to find the vector $\mathbf{a} = [a_1, \ldots, a_n] \in \Re^n$ which a generic data pattern $\mathbf{x}$ is projected onto, so as to maximize the variance of the projection z:

$$z = a_1 x_1 + \cdots + a_n x_n = \mathbf{a}^T \mathbf{x}\,. \tag{1}$$

It is proved that a value maximizing the variance of z is obtained when $\mathbf{a}$ is the eigenvector of $\mathrm{var}(\mathbf{x})$ corresponding to its largest eigenvalue; thus, in the case of basic PCA, the algorithmic procedure is the following for a given matrix $\mathbf{X}$ with dimensions $[N, f]$, where N and f are the number of samples and features, respectively:

1. Normalize the data matrix $\mathbf{X}$ so that each column of $\bar{\mathbf{X}}$ will assume a null mean and unitary variance;
2. Compute the Singular Value Decomposition on $\bar{\mathbf{X}}$:

   $$\bar{\mathbf{X}} = \mathbf{U}\mathbf{D}\mathbf{V}^T\,. \tag{2}$$

   where $\mathbf{U}$ is the orthogonal matrix of order N, $\mathbf{D}$ is a rectangular diagonal matrix with dimensions $[N, f]$, where the diagonal elements of $\mathbf{D}$ assume values $d_1 \geq d_2 \geq \ldots \geq d_f$, and $\mathbf{V}$ is an orthogonal matrix of order f;
3. The new variables in lower dimensional space are computed by choosing the first $k \leq f$ columns of matrix $\mathbf{Z}$ where:

   $$\mathbf{Z} = \bar{\mathbf{X}}\mathbf{V} = \mathbf{U}\mathbf{D}\,. \tag{3}$$

   There are many ways to choose the optimal number of PC, where one of them is to take into account the percentage amount of variance in the chosen components where a value greater than 95% is considered satisfactory.

PCA-based methods have been applied for reducing the computational burden in a large number of smart grid applications. In particular, in [5], the PCA has been applied in order to solve power flow and optimal power flow problems in large-scale power systems. In this study, a new formalization of the system equations in the PCA domain allowed for reducing the problems cardinality by identifying the hidden relations between the state variables obtained from the analysis of the historical problem solutions. Furthermore, the application of PCA has proved to be effective in wide-area smart grid monitoring, where it allows for developing effective online power system security analysis, by reducing the complexity of the contingency screening process [56]. Other interesting application domains include the definition of strategic bidding strategies for wind power generators, where PCA has been applied in the task of finding hidden correlations between spatially distributed wind farms [57], and the development of spatial and temporal wind power forecasting tools based on Knowledge Discovery from large datasets [41].

If cardinality reduction does not supply adequate results, an alternative is represented by the feature selection ones. Differently from these former, the latter does not transform the original variables, but they subset the original dataset to the most relevant features according to a certain metric [29]. In literature, the research started to explore selecting the best features in order to choose those that maximize the mutual information between them and a target variable, and this is called maximum relevancy strategy. Unfortunately, several works of literature have proved that the best selected features by maximum relevancy do not guarantee the best prediction accuracy [58]. The reason for this is related to the neglecting of feature redundancy. Considering this, a trade-off between lesser redundancy and greater maximum relevancy was considered through the development of the minimum Relevancy Maximum Redundancy technique [59] to overcome the maximum relevancy limit. Mathematically, applying the mRMR technique corresponds to maximizing the following function:

$$\max_{x_j \in X - B_{d-1}} \left[I\left(x_j; v\right) - \frac{1}{d-1} \sum_{x_j \in B_{d-1}} I\left(x_j; x_i\right) \right] \tag{4}$$

where X is a set of generic features, B is a set of the features already considered, d is the number of desired best features, v is a generic target variable, $I(.)$ is the mutual dependency function, and x_j and x_i a generic feature of B and X, respectively. In (4), the left member in the parenthesis is the relevancy between the jth feature and the target variable, whilst the right member is the redundancy between the jth feature and the others of B.

4. Proposed Methodology

The Knowledge Discovery Process aims at extracting useful hidden information from the available data. Usefulness stands for the quality of having something to supply an advantage to the user. In particular, revealed information is useful when it is used either for gaining a direct knowledge from its visualization or for being processed in a further information process in order to extract new knowledge. For this reason, the proposed methodology aims at proving the capability to develop an accurate full data-driven model based on KDP for multi-temporal forecasting. Hence, revealed information is useful when it is processed for visualization or to be used for further data processing, as in case of the prediction models.

When the number of time step ahead to predict increases, the challenge is to characterize the behavior of the signal to predict in order to catch correlations for different periods. Hence, harnessing the hidden information content of the available data is crucial for developing a good forecasting model, since raw data are seldom suitable for an immediate effective use. For this reason, the proposed methodology, whose workflow scheme is reported in Figure 1, includes: (i) a tool for transforming date and time information in numerical predictor variables; (ii) a procedure of feature engineering; (iii) a tool for adapting the time series prediction problem in a supervised learning one; (iv) a procedure of feature selection; (v) two predictive model based on based on random forest and lazy learning; (vi) time rolling windows validation; (vii) statistical analysis of the results.

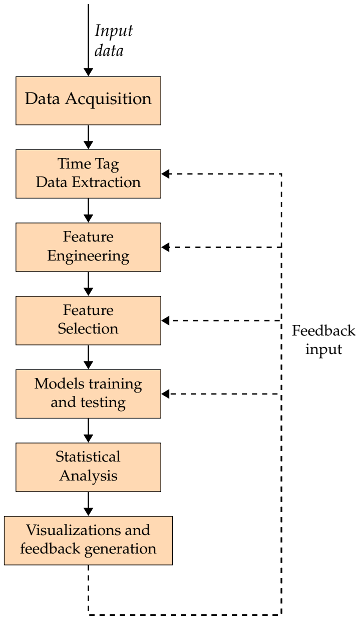

Figure 1. Visualization of the described methodology workflow.

1. Time referenced datasets about load consumption acquired by smart meters and customer substations are precious information sources to extract in order to catch the user behavior profile. Generally, electric load trajectory shapes assume similarity patterns according to the season, the day type (workweek, weekend, and annual holidays), and load type (households, tertiary sector, industrial, etc.). Electric prices, weather conditions, and spot social events complete the phenomena list affecting the electric load. It is clear that much of this information such as date–time are available in string or character format, needing adequate transformations to allow the application of regression models. Given a date–time sample, a simple preprocessing step allows for extracting several useful codified variables, including their type and timestamp, which are relevant to season, month, day of the week, day of the month, and so on.
2. A raw time series matrix $\mathbf{Y}_0$, which is characterized by n_0 samples and c_0 variables (or features), is often characterized by noise or chaotic behavior, which do not allow a clear understanding of the signal trajectory over the time. Excessive volatility needs to be managed in order to have more stable signals, which are able to catch the time series trend. For this reason, the application of feature engineering moves toward this direction, by allowing the extraction of a large number of hidden features and smooth signals from the original time series, producing the matrix $\mathbf{Y}$, which has dimensions $[n_0, c]$, with $c > c_0$. In this sense, Table 1 summarizes the main smoothing variables used in the literature and the corresponding variable. For the sake of clarity, matrix dimensions are summarized in Table 2.
3. The supervised learning approach for time series forecasting requires a transformation of data, which are usually arranged in a matrix form. Preparing data for this approach requires producing a couple of input–output set for each sample t (the jth rows of matrix $\mathbf{Y}$), which considers a portion of the predictor trajectories (how many samples in the past are considered as process memory) and the forecasting horizon of the target variables (how many samples ahead we want to predict) (Figure 2). The embedding procedure is a map between the samples of a time-series, which produces two matrices $\mathbf{P}$, whose dimensions are n_1 and p, and $\mathbf{R}$, whose dimensions are n_1 and r, called predictors and target matrices, respectively, given an input matrix $\mathbf{Y}$, once assigned an auto-regressive lag L, a

delay d, and a forecasting horizon H. The parameter r is computed by the product between c_r and H, where c_r is the number of variables in $\mathbf{Y}$ to predict. The delay is crucial since it shifts the most recent available sample in the past at time t. A rough indication about number of L can be chosen on the basis of the signal auto-correlation analysis. $\mathbf{P}$ and $\mathbf{R}$ were consequently split into $\mathbf{P}_t$, $\mathbf{P}_v$, $\mathbf{R}_t$, and $\mathbf{R}_v$, which are the training and test set of the predictors and target matrices. For the sake of clarity, the variable list is summarized in Table 2.

4. The previous steps cause a huge increase in the number of variables; indeed, L new predictors (the lagged variables in the past) are produced for each starting variable (columns of $\mathbf{Y}$). Unfortunately, the consequence of this cardinality growth may cause collateral effects on the prediction accuracy, since a large dimension of data causes the previously described "curse of dimensionality", which causes critical issues in the right operation of learning models. For this reason, techniques for cardinality reduction, as PCA, and feature selection, as MRMR, were considered. As described, the main difference between them is that the former produces a new set of uncorrelated variables in the PCA domain, whilst the latter extracts the most correlated and lesser redundant variables with respect to a target variable without transforming the original dataset. The reduced predictor training and test matrices are defined as $\mathbf{P}_{t,r}$ and $\mathbf{P}_{v,r}$.
5. Two different machine-learning models such as Lazy Learning [60] and Random Forest Regression [61] are assessed in this methodology. Random Forest (RF) origins arise from the bootstrap aggregation (bagging), which is a technique aimed at reducing the variance of the prediction function by averaging several prediction functions trained with random extracted samples from the dataset. RF extended this concept to the features in order to build decorrelated trees, where a random selection of variable is considered for each split. On the contrary, a Lazy Learning model as the K-Nearest Neighbors is based on local regression, where the predictor training set is used to extract the nearest neighbor samples given a query one. These latter and the corresponding targets are consequently used for building a local learner that supplies the prediction. Since the nearest neighbors are chosen by discriminating them considering a distance metric, the reduction of cardinality is crucial to reduce the number of dimensions (features) to consider in the distance computation. According to the multi-step nature of the problem, a direct strategy was applied, which, even if it requests a more computational effort with respect to an iterative approach, is less subject to the error explosion. Hence, the multi-step load forecasting problem was decomposed in H MISO problems, one for each time step ahead.
6. An exhaustive proposed methodology validation requires testing on a large number of cases in order to appreciate the spreading of accuracy performance at the changing of training and test sets. For this reason, a time-rolling window validation was employed to slice $\mathbf{Y}$ in the ith training and test sets, according to a sequence of splitting points.
7. The model performance data were analyzed in order to assess the effectiveness of the proposed methodology, where a Naive model was considered as a benchmark. The tests were performed by progressively increasing the forecasting horizons. The MSE was computed for both sample (jth row of $\mathbf{R}_v^{(i)}$) and wth target variable over the considered forecasting horizon span according to the (5):

$$\mathrm{MSE}_{j,w}^{(i)} = \sqrt{\frac{1}{H}\sum_{h}^{H}\left(\hat{\mathbf{R}}_v^{(i)}[j,(w-1)H+h] - \mathbf{R}_v^{(i)}[j,(w-1)H+h]\right)^2} \tag{5}$$

where $\hat{\mathbf{R}}_v^{(i)}$ is the predicted value matrix for the ith test case, $w \in [1, c_r]$ is an indexing variable used for slicing over the columns both $\hat{\mathbf{R}}_v^{(i)}$ and $\mathbf{R}_v^{(i)}$ in order to extract the forecasting horizon span for the wth target, where n_v is the row number of $\hat{\mathbf{R}}_v^{(i)}$ and $\mathbf{R}_v^{(i)}$:

$$\text{NN-MSE}^{(i)} = \frac{1}{n_v \cdot c_r} \sum_{j=1}^{n_v} \sum_{w=1}^{c_r} \text{MSE}_{j,w}^{(i)}. \tag{6}$$

8. Aggregate data are performed by considering statistical tests as Friedman tests [62]. The aim is to assess if the model performs differently or not. In particular, the Friedman test is a non-parametric randomized block of analysis of variance, where the null hypothesis H_0 considers all methods having the same error distribution. The test does not assume any hypothesis about data distribution. If the test rejects the null hypothesis, the Tukey-based Post Hoc test is performed in order to analyze the difference between the performance of each couple of models. In particular, the Tukey's test supplies an upper triangular matrix where the elements are sorted by an accuracy rank. This information is processed for producing useful visualizations for the choice of the best model.

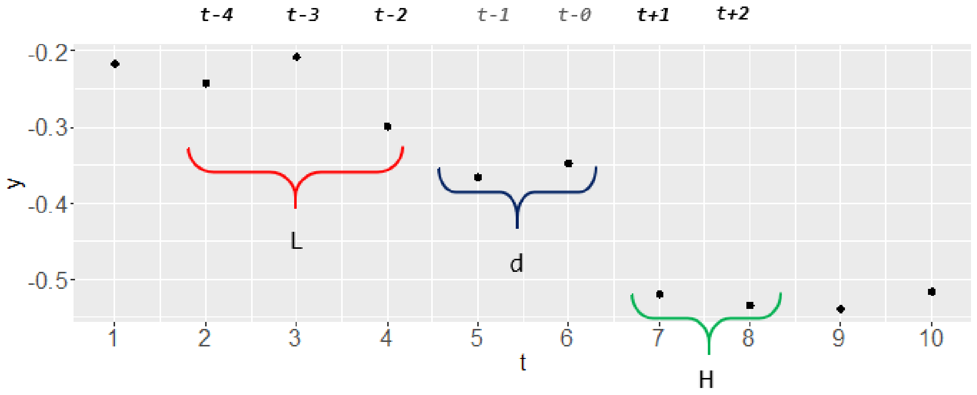

Figure 2. Visualization of key parameters in time-series supervised learning.

Table 1. Most common smooth univariate features.

Feature	**Equation**	**Notes**
smoothing average	$\frac{1}{m+1}\sum_{\gamma=0}^{m} \mathbf{Y}[t-\gamma, z]$	t and z are the generic time sample and raw variable, respectively m is the size of the rolling window
rolling upper bound	$\max_{\gamma\in[0,m]} (\mathbf{Y}[t-\gamma, z])$	
rolling lower bound	$\min_{\gamma\in[0,m]} (\mathbf{Y}[t-\gamma, z])$	
1st order difference	$\mathbf{Y}[t, z] - \mathbf{Y}[t-1, z]$	
absolute 1st order difference	$\lvert\mathbf{Y}[t, z] - \mathbf{Y}[t-1, z]\rvert$	
p-quantile	$\inf\{x \in \Re : p \leq F(x)\} = p$	where X is the population of $\mathbf{Y}[t-\gamma, z]\forall\gamma \in [0, m]$

Table 2. Matrix relation in a Supervised Learning strategy for Time Series Forecasting.

Matrix	No. of Samples (Rows)	No. of Variables (Columns)	Notes
$\mathbf{Y}_0$	n_0	c_0	raw signal matrix
$\mathbf{Y}$	n_0	c	signal matrix after feature engineering process $c = c_0 \cdot (1 + q)$, q is the number of features made per variable of $\mathbf{Y}_0$
$\mathbf{Y}^{(i)}$	n	c	slice of $\mathbf{Y}$ used in the ith case test
$\mathbf{P}^{(i)}$	n_1	p	predictor matrix $n_1 = n - (L + H + d - 1)\ ;\ \ p = (c - c_r) \cdot L$ where c_r is the number of target variables
$\mathbf{R}^{(i)}$	n_1	r	target matrix $r = c_r \cdot H$
$\mathbf{P}_t^{(i)}$	n_t	p	training predictor matrix $n_1 = n_t + n_v$
$\mathbf{P}_v^{(i)}$	n_v	p	training predictor matrix
$\mathbf{R}_t^{(i)}$	n_t	r	test target matrix
$\mathbf{R}_v^{(t)}$	n_v	r	test target matrix
$\mathbf{P}_{t,r}^{(i)}$	n_t	f	reduced training target matrix $f << p$ is the number of selected features/Principal Components by applying MRMR / PCA
$\mathbf{P}_{v,r}^{(i)}$	n_t	f	reduced test target matrix

5. Case Study

The proposed methodology was tested in the task of analyzing a large dataset generated by a pervasive smart meter network deployed on a large commercial user located in the south of Italy, whose main features are summarized in Figure 3. In particular, the heat maps show the consumption level of active/reactive power, and the power factor over the whole day considering a full month. The active power heat map (above inset) reveals the highest consumption level is mainly related to time window 8–18. The central inset shows the reactive power level, which the observed pattern deflates from the active power ones. This is confirmed by the below inset, which depicts the distribution of power factor over the day.

The sample time resolution is 5 min for a period of one month. The considered dataset includes the following time referenced measurements: apparent [kVA], active [kW], and reactive [$kVAR$] three phase power, line current [A], and phase–voltage [V], and Total Harmonic Distortion [%] for each phase. The date and time column was decomposed into day of the week (0–6), of the month (1–30), in hours and minutes. The target variable is the active power for an assigned forecasting horizon. Furthermore, according to the data, the time resolution at the 1 h forecasting horizon corresponds to 12 steps ahead. The simulations were performed on an Intel®I7-9700 CPU, by running a single core instance of R.

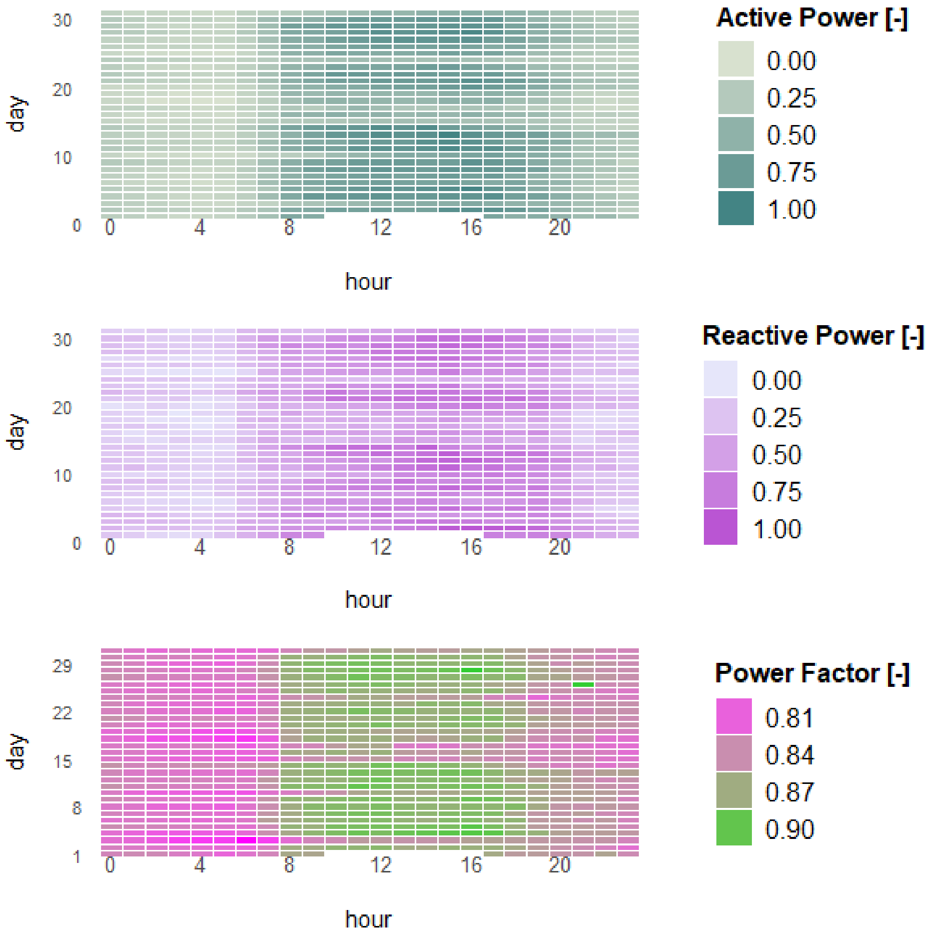

Figure 3. Visualization of the analyzed utility distinctive features.

Two case studies, named 'A' and 'B', were conducted on the prediction of three-phase active power. In particular, case A analyzed three forecasting horizons $H = \{12, 72, 144\}$, which are equivalent to 2, 6, and 12 h ahead; for the sake of conciseness, the case study set-up was depicted in Table 3 with time resolution of 5 min. Several forecasting horizons are chosen for assessing the methodology and accuracy performance at the increasing of the different forecasting horizon. The considered values are related to the most frequent time constraints for the submission of offers in electricity markets considering the possibility for the utility to participate in energy/ancillary services markets. Clearly, this kind of forecasting may be used to manage the utility, to schedule several activities considering external needs, such as to avoid system stress conditions caused by huge load levels. For each forecasting horizon, the raw data were processed according to the described pipeline.

The raw dataset $\mathbf{Y}_0$ was normalized and smooth variables were computed for each dataset variable for different lag time adding them to the available variable set. According to a generated splitting point set, a subset of $\mathbf{Y}^{(i)}$ is extracted in order to be transformed in the predictor and target matrices $\mathbf{P}^{(i)}$ and $\mathbf{T}^{(i)}$ through the embedding procedure. Each one of these matrices was consequently split into $\mathbf{P}_t^{(i)}$ $\mathbf{P}_v^{(i)}$, $\mathbf{T}_t^{(i)}$, $\mathbf{T}_v^{(i)}$, which are the training and test sets of the predictor and target matrices.

Since it is not reasonable that all predictors have the same information, PCA and MRMR were considered to reduce the dataset to the most meaningful variables. As shown by preliminary results, we selected the MRMR since PCA has shown a reduced capability to reconstruct the predictor matrix test set in the presence of high noisy data, reducing the prediction accuracy. Unfortunately, the adoption of a direct prediction strategy has required the production of a number of models equal to the time steps ahead to predict. Consequently, MRMR had to be performed the same number of times in order to find the most correlated predictors to the hth time step ahead. For this reason, a sub-optimal solution was to apply the MRMR only one time between the predictors training matrix $\mathbf{P}_t^{(i)}$ and the nearest hth time step ahead to the half width of the forecasting horizon, where the optimal number of selected features was chosen by a preliminary analysis.

Once the f most meaningful predictors were selected, the training set matrices were processed by supervised learning models to train them. In particular, Random Forest, Lazy Learning, and Naive were compared. The Naive models supply each prediction over the forecasting horizons by averaging the available samples according to (7):

$$\mathbf{Y}[t,h] = \frac{1}{g}\sum_{k=0}^{g-1}\mathbf{Y}[t-H-k,h]\ \forall h \in [1,H] \tag{7}$$

where h is the hth time step ahead and g is the number of samples considered for computing the expected value.

Table 3. Experimental setup: Case A.

Parameter	Value	Parameter	Value
H	$\{24, 36, 144\}$	d	0
L	$5 \cdot H$	f	5
NN	5	n_0	∼8800
c_0	20	c	∼100
c_r	1	p	$(c - c_r) \cdot L$
n_t	∼2000	n_v	∼4·H
n_{tw}	n_v		

The case study B changes the resolution of the described data from 5 min to 30 for reducing both the high volatility of the time series, which is shown by Figures 2–5 and the computational costs. In particular, the tested forecasting horizons are 2, 3, 6 h, which correspond to $H = 4$, 6, 12, and where the experimental set-up is summarized in Table 4. In this case, RF and Lazy Learning are performed by reducing the predictor training set by using both MRMR and PCA. An important difference in the described framework between case A and B is related to the choice of the best features by MRMR. Indeed, considering the reduced computational cost deriving by the reduction of time resolution, the application of MRMR for each step of the forecasting horizon span becomes feasible. Furthermore, this case study includes a further Naive model (Naive 2), where the predicted value for a certain time of the day is computed by averaging the occurred values for the same time in the days behind. This model was added because it works differently from the traditional time series forecasting model in order to try to catch some difference in the performance behavior.

Table 4. Experimental setup: Case B.

Parameter	Value	Parameter	Value
H	$\{4, 6, 12\}$	d	0
L	5	f_{MRMR}, f_{PCA}	6.5
NN	5	n_0	∼600
c_0	20	c	∼100
c_r	1	p	$(c - c_r) \cdot L$
n_t	∼700	n_v	$\sim 4 \cdot H$
n_{tw}	n_v		

6. Experimental Results

6.1. Case A

According to the analyzed case studies, Random Forest and Lazy Learning have shown a better prediction accuracy than the Naive model, especially for large forecasting horizons as proved by boxplot visualization in Figure 4. Particularly, the Naive model performs similarly to more complex ones as shown by the left plot of Figure 4. Indeed, when the signal is much noisier, it may compromise the entire data processing system, decreasing the prediction accuracy of the more complex models, which try to catch relationship inside data. Differently, the latter does not affect Naive since it neglects any form of data analysis.

Obviously, the Naive model predicts the forecasting horizon by performing a simple moving average of the available past samples of the signal to be predicted, revealing the dramatic detriment of its performance at the increasing of forecasting horizon as shown in Figures 5 and 6. In particular, these latter show the actual and predicted load trajectories for two samples of the forecasting horizon span, where the volatility of the signal is well highlighted by the current signal trajectories (red lines).

The computational burden rises linearly at the increasing of the forecasting horizon, where the maximum waiting time is 3 min for predicting a 300 time sample test target matrix with a 12 forecasting horizon span per time sample. Each time sample ahead was predicted by applying a direct strategy for both Random Forest and Lazy Learning models. According to the set-up of rolling window, the number of test cases are a, b, and c for 2, 6, and 12 h ahead case studies, respectively.

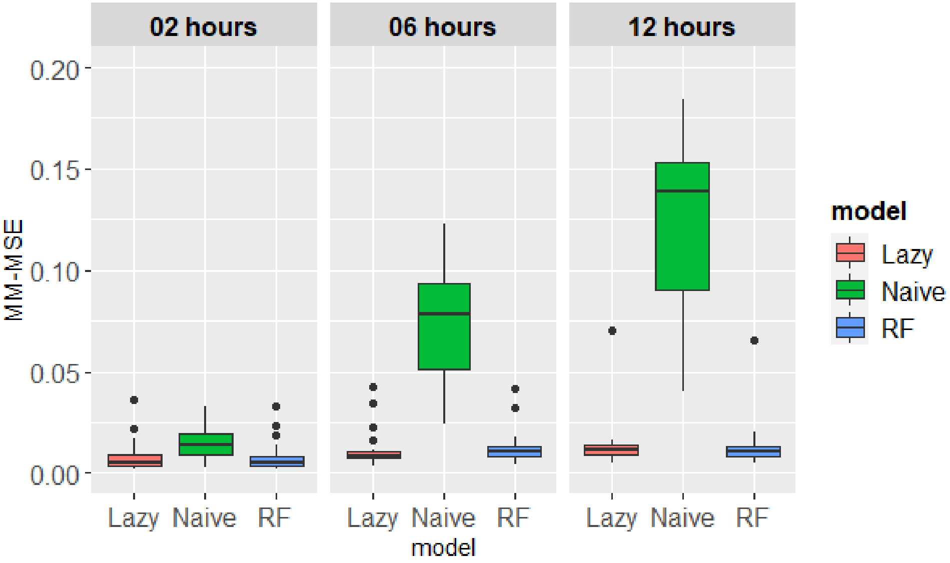

Figure 4. Visualization of MM-MSE at the changing of Forecasting Horizon.

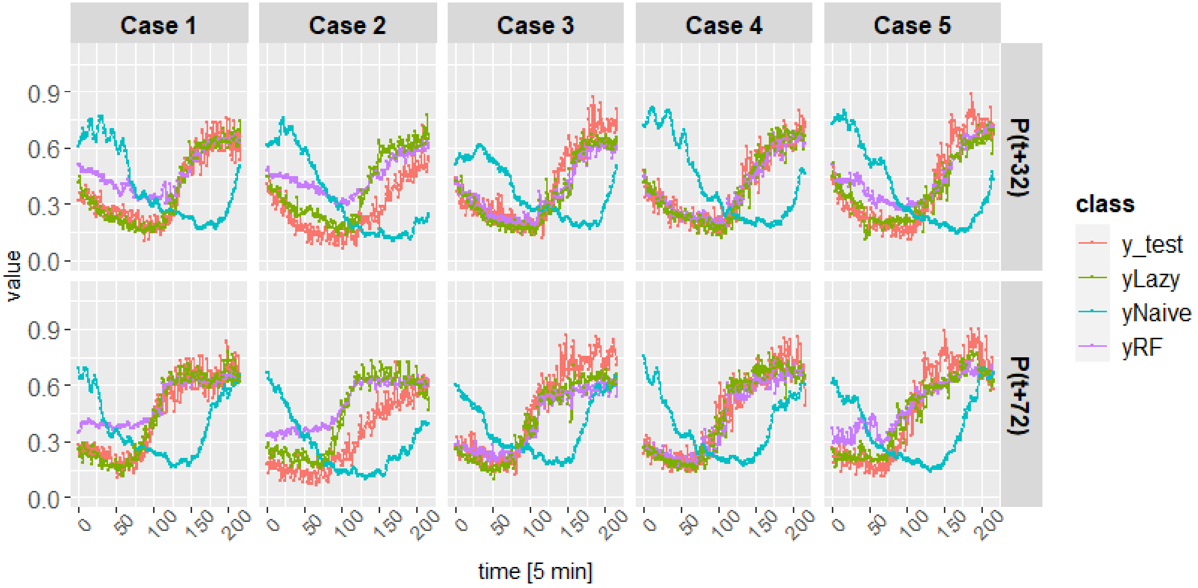

Figure 5. Visualization of actual (*ytest*) and predicted load trajectories for a 6 h forecasting horizon.

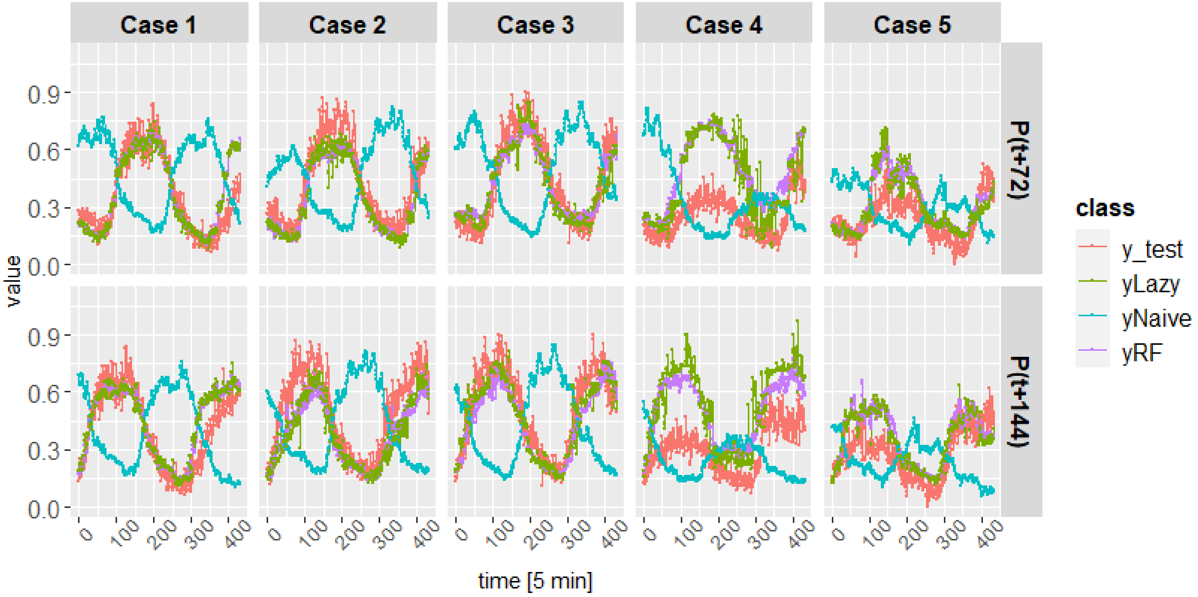

Figure 6. Visualization of actual (*ytest*) and predicted load trajectories for a 12 h forecasting horizon.

6.2. Case B

The case study B is focused on comparing the performance of PCA and MRMR in forecasting applications. In particular, their effectiveness appears related to the type of coupled machine learning algorithms as shown in Figure 7. Indeed, the PCA performs well in combination with Random Forest, whereas the combination of PCA with Lazy Learning shows the worst performance for all forecasting horizons. It is interesting noting that the MRMR-based model better performs than any PCA-based model. These results are confirmed by observing the trajectories for the considered forecasting horizon (Figures 8–10).

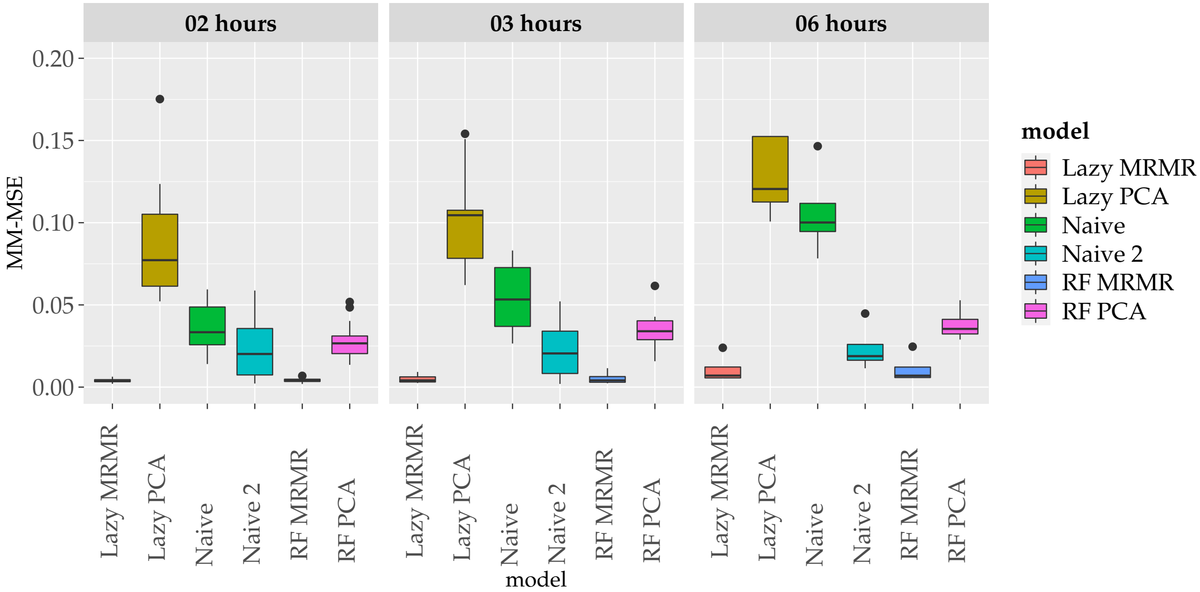

Figure 7. Visualization of MM-MSE at the changing of Forecasting Horizon.

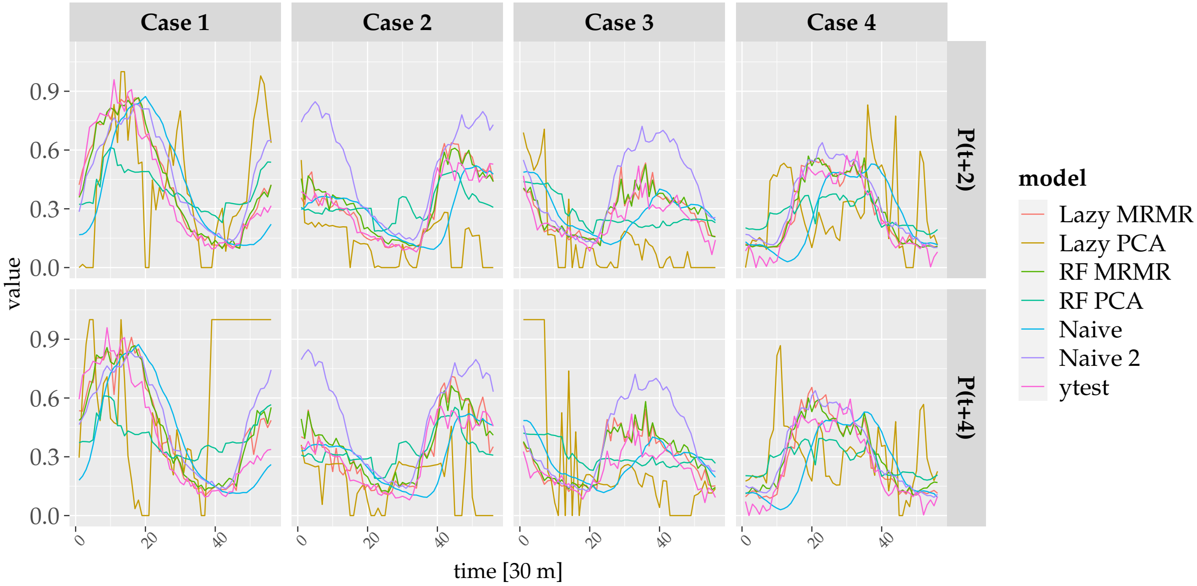

Figure 8. Visualization of actual (*ytest*) and predicted load trajectories for a 2 h forecasting horizon.

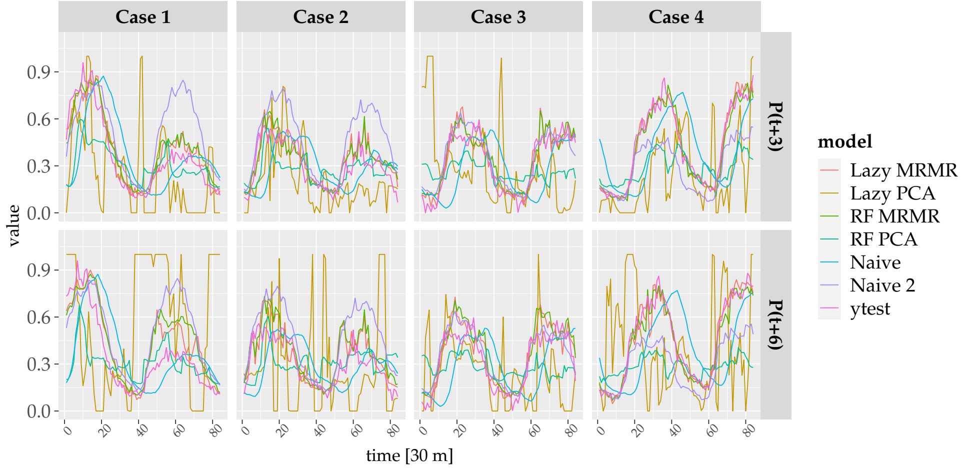

Figure 9. Visualization of actual (*ytest*) and predicted load trajectories for a 3 h forecasting horizon.

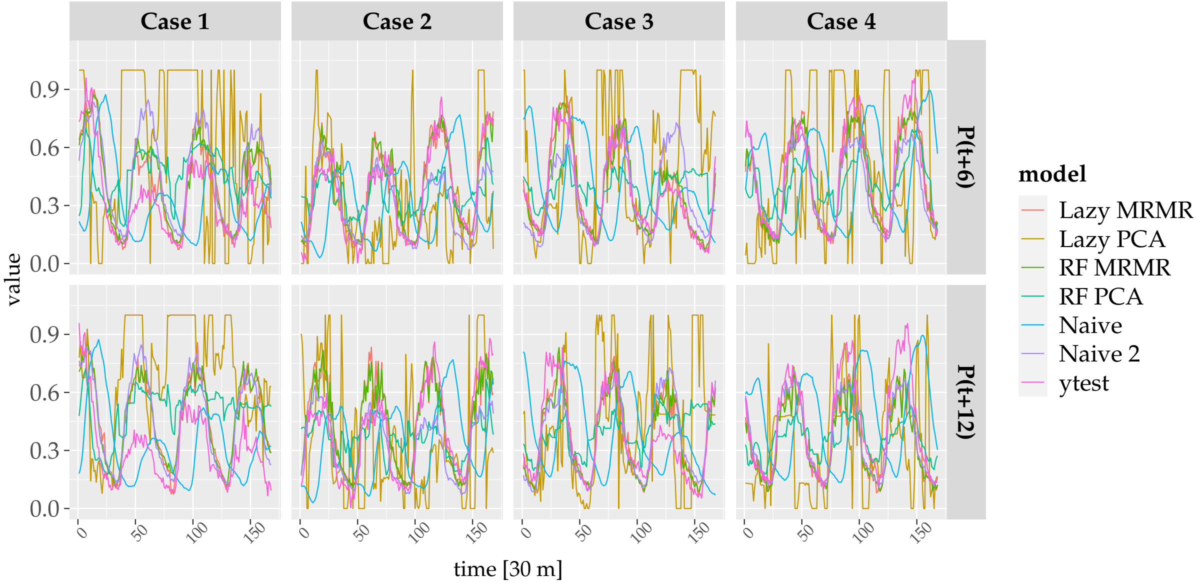

Figure 10. Visualization of actual (*ytest*) and predicted load trajectories for a 6 h forecasting horizon.

The latter figures show the Lazy Learning-PCA lower accuracy than the others forecasting models because it is unable to follow the true value (*ytest*). On the contrary, the employment of Lazy Learning in combination with MRMR produces an accurate forecasting, where the actual and predictor trajectories are often very close. The low-accuracy of PCA-based models may be related to the low capacity of PCA to transform in the original domain new data, where Recursive PCA may improve the accuracy [63].

According to the workflow in Figure 11, the model performance aggregation is processed by considering the Friedman's test. Since for each forecasting horizon the null hypothesis is rejected, the Tukey-based Post Hoc test for checking dissimilarities between each couple of models is performed. In particular, as observed in Figure 11, the Post Hoc test outputs are fused in a heat map according to the KDD principles. The latter has the models arranged according to the Post Hoc rank on both axes, where each cell of the map is the result of Post Hoc test between two models. The first element of the rank is arranged on the lower left corner. The green colored cell means that the model performs equally, whereas the orange cell means that the model is statistically different. In particular, for $H = 2$ h, the MRMR-based

Lazy Learning model is the most accurate one, but its performance cannot be considered significantly different from the second model in the rank, which is the MRMR-based Random Forest. The Post Hoc test for $H = 3$ h and $H = 6$ h does not show relevant differences with respect to $H = 2$ h. In conclusion, it is clear that a similar visualization is effective because it allows a rapid understanding of the performance differences between two models, supporting the decision maker in the analysis of the most suitable model.

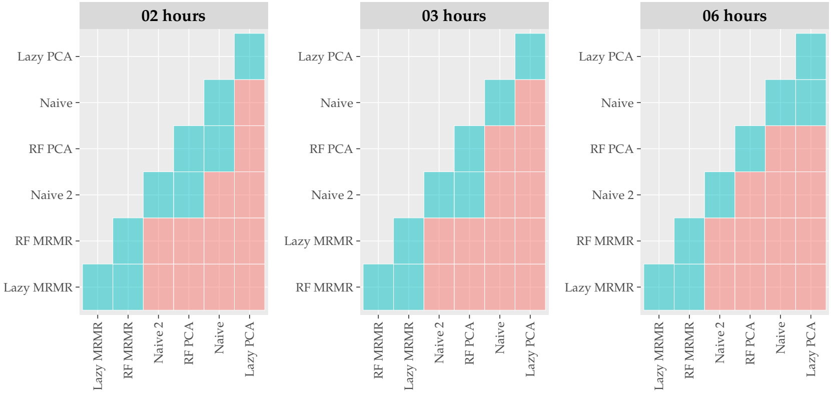

Figure 11. Visualization of a Post Hoc Test.

7. Critical Discussion

In particular, according to both methodology workflow description and the obtained results, the main advantage of this framework is its generalization capability. Indeed, the authors similarly addressed forecasting problems that applied to different environments such as in wind power forecasting [64].

Furthermore, as happens in every machine learning framework, one of the drawbacks is the prediction accuracy, which depends on the training/validation set features. If out of knowledge patterns appear in the validation set, it is highly probable that the forecasting accuracy will decrease. In this case, the decision maker is supported by the KDD in the preliminary data-analysis steps, which allows for recognizing possible seasonal cycles in the target profile. The latter allows for making a correct tune-up of the model, by considering an adequate size of the training set or a certain number of smooth/lagged variables.

In particular, one of the potential limits is processing data evolving without a certain pattern over the time. Indeed, in the case of utility load, where the consumption profile over the days assumes similar schemes, the methodology works well also for high forecasting horizons since it not hard find correlation between the predictor and the target over the time.

With the presence of high volatility data, a reasonable approach may be combined different models, based on different learners or trained with different data features. In particular, adaptive ensemble forecasting, where the forecasting is supplied by averaging the prediction of single learners according to weights reflecting their local accuracy, may increase the prediction accuracy without recurring to complex and time-consuming deep learning models.

8. Conclusions

In even more connected and liberalized power systems, the information volume exchange is dramatically growing, causing the generation of massive data sets, which may deteriorate the effectiveness of the traditional exploration and data mining tools in supplying useful knowledge to the power system stakeholders. For this reason, we explored the current scenario about the employment of artificial intelligence in smart grids, with particular interest to the decision support systems and data extraction.

For this reason, we propose this review, which aims at characterizing the employment of artificial intelligence in power systems, analyzing the main critical issues, and of the most relevant KDD-based methodology in power systems, exploring their advantages and drawbacks. At the same time, we conduct a critical analysis of a forecasting framework inspired by the KDD fundamental steps, analyzing it in a data-driven load forecasting case study.

In particular, from the analysis of the literature, the Knowledge Discovery has emerged as a fundamental tool in smart grid computing by allowing system operators to model the semantics of the data, instead of just relying on the syntactic and structural representations, and to access the data resources solving the heterogeneity problems. This could allow smart grids computing entities to closely interact at human conceptual levels, providing functionalities for ontology management, query, and inference services. In this context, the future research activities will be oriented toward the conceptualization of an ontology middleware system, which processes real or near real-time data streaming generated by heterogeneous data-sources, ontology-based services, and intelligent reasoning. In particular, they allow for enabling a Knowledge Discovery process based on the information context instead of just keyword based searches.

Furthermore, the second part of this manuscript, by analyzing a specific KDD-based methodology for data-driven load forecasting, aims at analyzing its potential in a real case study, describing how the KDD may improve the development of a decision support system. The conducted experimental analysis allows for assessing the quality of the proposed KDD-based methodology for load forecasting, where the obtained results clearly indicate the future research trends in this field.

Author Contributions: Conceptualization, F.D.C., A.V.; methodology, F.D.C., A.V., software, F.D.C., validation, F.D.C., A.V., investigation, F.D.C., A.V., resources, D.V., data curation, F.D.C., supervision, A.V., writing—original draft preparation, F.D.C., A.V.; writing—review and editing, R.A., A.A., M.P., A.R.; visualization, F.D.C.; supervision, D.V., A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
AMI	Advanced Metering Interface
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
DG	Distributed Generation
ESN	Echo State Network
GRU	Gated Recurrent Unit
KDD	Knowledge Discovery Process
LSTM	Long Short Term Memory unit
mRMR	Minimum Redundancy Maximum Relevancy
PCA	Principal Component Analysis
PMU	Phasor Measurement Unit
PSOPE	Power System Operation, Planning, and Economics
RNN	Recurrent Neural Network
SCADA	Supervisory Control And Data Acquisition
WAMS	Wide Area Measurement Systems

References

1. Madani, V.; King, R.L. Strategies and roadmaps to meet grid challenges for safety and reliability. In *Innovations in Power Systems Reliability*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 1–11.
2. King, R.L. Information services for smart grids. In Proceedings of the 2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century, Pittsburgh, PA, USA, 20–24 July 2008; pp. 1–5.
3. Zobaa, A.F.; Vaccaro, A.; Lai, L.L. Guest editorial enabling technologies and methodologies for knowledge discovery and data mining in smart grids. *IEEE Trans. Ind. Inform.* **2016**, *12*, 820–823. [CrossRef]
4. Loia, V.; Furno, D.; Vaccaro, A. Decentralised smart grids monitoring by swarm-based semantic sensor data analysis. *Int. J. Syst. Control. Commun.* **2013**, *5*, 1–14. [CrossRef]
5. Vaccaro, A.; Cañizares, C.A. A Knowledge-Based Framework for Power Flow and Optimal Power Flow Analyses. *IEEE Trans. Smart Grid* **2018**, *9*, 230–239. [CrossRef]
6. Gu, Y.; Jiang, H.; Zhang, Y.; Zhang, J.J.; Gao, T.; Muljadi, E. Knowledge discovery for smart grid operation, control, and situation awareness—A big data visualization platform. In Proceedings of the 2016 North American Power Symposium (NAPS), Denver, CO, USA, 18–20 September 2016; pp. 1–6. [CrossRef]
7. Xu, Y.; Zhang, Y.; Dong, Z.Y.; Zhang, R. *Intelligent Systems for Stability Assessment and Control of Smart Power Grids: Security Analysis, Optimization, and Knowledge Discovery*; CRC Press: Boca Raton, FL, USA, 2020.
8. Rodrigues, P.P.; Gama, J. Holistic distributed stream clustering for smart grids. In Proceedings of the Workshop on Ubiquitous Data Mining, Montpellier, France, 27 August 2012; p. 18.
9. Shanmuganathan, S. From data mining and knowledge discovery to big data analytics and knowledge extraction for applications in science. *J. Comput. Sci.* **2014**, *10*, 2658–2665. [CrossRef]
10. Green, R.C.; Wang, L.; Alam, M. Applications and trends of high performance computing for electric power systems: Focusing on smart grid. *IEEE Trans. Smart Grid* **2013**, *4*, 922–931. [CrossRef]
11. Soroudi, A.; Amraee, T. Decision making under uncertainty in energy systems: State of the art. *Renew. Sustain. Energy Rev.* **2013**, *28*, 376–384. [CrossRef]
12. Wang, Y.; Zhang, N.; Kang, C.; Miao, M.; Shi, R.; Xia, Q. An efficient approach to power system uncertainty analysis with high-dimensional dependencies. *IEEE Trans. Power Syst.* **2017**, *33*, 2984–2994. [CrossRef]
13. Bhattarai, B.P.; Paudyal, S.; Luo, Y.; Mohanpurkar, M.; Cheung, K.; Tonkoski, R.; Hovsapian, R.; Myers, K.S.; Zhang, R.; Zhao, P.; et al. Big data analytics in smart grids: State-of-the-art, challenges, opportunities, and future directions. *IET Smart Grid* **2019**, *2*, 141–154. [CrossRef]
14. Allam, Z.; Dhunny, Z.A. On big data, artificial intelligence and smart cities. *Cities* **2019**, *89*, 80–91. [CrossRef]
15. Piatetsky-Shapiro, G.; Fayyad, U.; Smith, P. From data mining to knowledge discovery: An overview. *Adv. Knowl. Discov. Data Min.* **1996**, *1*, 35.
16. Kamiński, B.; Jakubczyk, M.; Szufel, P. A framework for sensitivity analysis of decision trees. *Cent. Eur. J. Oper. Res.* **2018**, *26*, 135–159. [CrossRef] [PubMed]
17. Elith, J.; Leathwick, J.R.; Hastie, T. A working guide to boosted regression trees. *J. Anim. Ecol.* **2008**, *77*, 802–813. [CrossRef] [PubMed]
18. Bontempi, G.; Ben Taieb, S. *Statistical Foundations of Machine Learning*; Université Libre de Bruxelles: Bruxelles, Belgium, 2008.
19. Haykin, S. *Neural Networks and Learning Machines*, 3rd ed.; Pearson Education, Inc.: Hoboken, NJ, USA, 2009.
20. Jang, J.S.; Sun, C.T.; Mizutani, E. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*; Prentice-Hall: Upper Saddle River, NJ, USA, 1997.
21. Rosato, A.; Rosa, A.; Araneo, R.; Panella, M. Prediction in Photovoltaic Power by Neural Networks. *Energies* **2017**, *10*, 1003. [CrossRef]
22. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
23. Chollet, F.J.; Allaire, J. *Deep Learning with R*; Manning Publication: Shelter Island, NY, USA, 2018

24. Zhang, D.; Han, X.; Deng, C. Review on the research and practice of deep learning and reinforcement learning in smart grids. *CSEE J. Power Energy Syst.* **2018**, *4*, 362–370. [CrossRef]
25. Dong, X.; Qian, L.; Huang, L. Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach. In Proceedings of the 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Korea, 13–16 February 2017; pp. 119–125.
26. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Cogn. Model.* **1988**, *5*, 1. [CrossRef]
27. Jaeger, H. *A Tutorial on Training Recurrent Neural Networks, Covering BPPT, RTRL, EKF and the "Echo State Network" Approach*; Technical report; German National Research Center for Information Technology: Sankt Augustin, Germany, 2005.
28. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
29. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef]
30. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
31. Kolodner, J. *Case-Based Reasoning*; Morgan Kaufmann: Burlington, MA, USA, 2014.
32. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
33. Bellman, R.E. *Adaptive Control Processes: A Guided Tour*; Princeton University Press: Princeton, NJ, USA, 2015.
34. Jensen, F.V. *An Introduction to Bayesian Networks*; UCL Press: London, UK, 1996; Volume 210.
35. De Raedt, L. A perspective on inductive logic programming. In *The Logic Programming Paradigm*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 335–346.
36. Arghandeh, R.; Zhou, Y. *Big Data Application in Power Systems*; Elsevier: Amsterdam, The Netherlands, 2017.
37. Cai, J.Y.; Huang, Z.; Hauer, J.; Martin, K. Current status and experience of WAMS implementation in North America. In Proceedings of the 2005 IEEE/PES Transmission & Distribution Conference & Exposition: Asia and Pacific, Dalian, China, 15–18 August 2005; pp. 1–7.
38. Zhou, Y.; Arghandeh, R.; Konstantakopoulos, I.; Abdullah, S.; von Meier, A.; Spanos, C.J. Abnormal event detection with high resolution micro-PMU data. In Proceedings of the 2016 Power Systems Computation Conference (PSCC), Genoa, Italy, 20–24 June 2016; pp. 1–7.
39. Zhang, J.; Chen, Z. The impact of AMI on the future power system. *Autom. Electr. Power Syst.* **2010**, *34*, 20–23.
40. Azimi, R.; Ghofrani, M.; Ghayekhloo, M. A hybrid wind power forecasting model based on data mining and wavelets analysis. *Energy Convers. Manag.* **2016**, *127*, 208–225. [CrossRef]
41. De Caro, F.; Vaccaro, A.; Villacci, D. Spatial and Temporal Wind Power Forecasting by Case-Based Reasoning Using Big-Data. *Energies* **2017**, *10*, 252. [CrossRef]
42. Quan, H.; Srinivasan, D.; Khambadkone, A.M.; Khosravi, A. A computational framework for uncertainty integration in stochastic unit commitment with intermittent renewable energy sources. *Appl. Energy* **2015**, *152*, 71–82. [CrossRef]
43. Singh, C.; Wang, L. Role of artificial intelligence in the reliability evaluation of electric power systems. *Turk. J. Electr. Eng. Comput. Sci.* **2008**, *16*, 189–200.
44. Tso, S.; Lin, J.; Ho, H.; Mak, C.; Yung, K.; Ho, Y. Data mining for detection of sensitive buses and influential buses in a power system subjected to disturbances. *IEEE Trans. Power Syst.* **2004**, *19*, 563–568. [CrossRef]
45. Rosato, A.; Panella, M.; Araneo, R. A Distributed Algorithm for the Cooperative Prediction of Power Production in PV Plants. *IEEE Trans. Energy Convers.* **2019**, *34*, 497–508. [CrossRef]
46. Rosato, A.; Panella, M.; Araneo, R.; Andreotti, A. A Neural Network Based Prediction System of Distributed Generation for the Management of Microgrids. *IEEE Trans. Ind. Appl.* **2019**, *55*, 7092–7102. [CrossRef]
47. Deka, D.; Chertkov, M. Topology Learning in Radial Distribution Grids. In *Big Data Application in Power Systems*; Elsevier: Amsterdam, The Netherlands, 2018; pp. 261–279.
48. Wang, Y.; Zhang, N.; Chen, Q.; Kirschen, D.S.; Li, P.; Xia, Q. Data-driven probabilistic net load forecasting with high penetration of behind-the-meter PV. *IEEE Trans. Power Syst.* **2017**, *33*, 3255–3264. [CrossRef]

49. Chicco, G. Overview and performance assessment of the clustering methods for electrical load pattern grouping. *Energy* **2012**, *42*, 68–80. [CrossRef]
50. Lu, X.; Dong, Z.Y.; Li, X. Electricity market price spike forecast with data mining techniques. *Electr. Power Syst. Res.* **2005**, *73*, 19–29. [CrossRef]
51. García, S.; Luengo, J.; Herrera, F. *Data Preprocessing in Data Mining*; Springer: Berlin/Heidelberg, Germany, 2015.
52. Peng, H.; Long, F.; Ding, C. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1226–1238. [CrossRef] [PubMed]
53. Han, J.; Pei, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Elsevier: Amsterdam, The Netherlands, 2011.
54. DuMouchel, W.; Volinsky, C.; Johnson, T.; Cortes, C.; Pregibon, D. Squashing flat files flatter. Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 15–18 August 1999; pp. 6–15.
55. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
56. Cai, L.; Thornhill, N.F.; Kuenzel, S.; Pal, B.C. Wide-Area Monitoring of Power Systems Using Principal Component Analysis and *k*-Nearest Neighbor Analysis. *IEEE Trans. Power Syst.* **2018**, *33*, 4913–4923. [CrossRef]
57. Qiao, S.; Wang, P.; Tao, T.; Shrestha, G. Maximizing profit of a wind genco considering geographical diversity of wind farms. *IEEE Trans. Power Syst.* **2014**, *30*, 2207–2215. [CrossRef]
58. Cover, T.M. The Best Two Independent Measurements Are Not the Two Best. *IEEE Trans. Syst. Man, Cybern.* **1974**, *SMC-4*, 116–117. [CrossRef]
59. Jain, A.K.; Duin, R.P.W.; Jianchang Mao. Statistical pattern recognition: a review. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 4–37. [CrossRef]
60. Birattari, M.; Bontempi, G.; Bersini, H. Lazy learning meets the recursive least squares algorithm. *Adv. Neural Inf. Process. Syst.* **1999**, *11*, 375–381.
61. Breiman, L. Random forests machine learning. *View Artic. Pubmed/Ncbi Google Sch.* **2001**, *45*, 5–32
62. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **1937**, *32*, 675–701. [CrossRef]
63. Jeng, J.C. Adaptive process monitoring using efficient recursive PCA and moving window PCA algorithms. *J. Taiwan Inst. Chem. Eng.* **2010**, *41*, 475–481. [CrossRef]
64. De Caro, F.; De Stefani, J.; Bontempi, G.; Vaccaro, A.; Villacci, D. Robust Assessment of Short-Term Wind Power Forecasting Models on Multiple Time Horizons. *Technol. Econ. Smart Grids Sustain. Energy* **2020**, *5*, 1–15. [CrossRef]