IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 9, 2021, accepted July 15, 2021, date of publication July 19, 2021, date of current version July 27, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3098121

Short-Term Electric Load Forecasting With Sparse Coding Methods

NIKOLAOS GIAMARELOS[1], ELIAS N. ZOIS[1], MYRON PAPADIMITRAKIS[1], (Student Member, IEEE), MARIOS STOGIANNOS[1], NIKOLAOS-ANTONIOS I. LIVANOS[2], AND ALEX ALEXANDRIDIS[1], (Member, IEEE)

[1]Telecommunications, Signal Processing and Intelligent Systems Laboratory, Department of Electrical and Electronic Engineering, University of West Attica, 122 41 Aigaleo, Greece

[2]EMTECH SPACE P.C., 144 51 Athens, Greece

Corresponding author: Alex Alexandridis (alexx@uniwa.gr)

This work was supported by the European Union and Greek National Funds through the Operational Program Competitiveness, Entrepreneurship, and Innovation, of the call RESEARCH–CREATE–INNOVATE under Project T1EDK-00244.

ABSTRACT Short-term load forecasting is a key task for planning and stability of the current and future distribution grid, as it can significantly contribute to the management of energy market for ancillary services. In this paper we introduce the beneficial properties of applications of sparse representation and corresponding dictionary learning to the net load forecasting problem on a substation level. In this context, sparse representation theory can provide parsimonial predictive models, which become attractive mainly due to their ability to successfully model the input space in a self-learning manner, by interacting between theory, algorithms, and applications. Several techniques are implemented, incorporating numerous dictionary learning and sparse decomposition algorithms, and a hierarchical structured model is proposed. The concept of sparsity in each case is embedded throughout the utilization of different regularization forms which include the ℓ_0, ℓ_1, ℓ_2 and ℓ_0^{tree} norms. The observed superiority of the proposed theory, especially the one which embeds the atoms and corresponding coefficients in a tree structure, stems from the construction of the dictionary so as to represent efficiently the ambient electricity signal space and the consequent extraction of sparse basis-vectors. The performance of each model is evaluated using real hourly load measurements from a high voltage/medium voltage (HV/MV) substation and compared with that of widely used machine learning methods. The provided analytical results, verify the effectiveness of hierarchical sparse representation in short-term load forecasting applications, in terms of common accuracy indices.

INDEX TERMS Generative models, hierarchical dictionaries, load forecasting, power grid, sparse representation.

I. INTRODUCTION

The ever-increasing energy requirements of modern power grids provide, beyond any doubt, evidence of the necessity to manage energy resources in the utmost efficient and cost-effective way. Electric load forecasting on a substation scale, plays central role in this effort, and this discipline, has become one of the major research fields in the context of electrical engineering [1]. The scope of this field is to provide predictions regarding the future values of load time-series based on previous collections of load measurements, while also often taking into account exogenous variables. As it is expected, a number of challenges appear when dealing with this problem. On the one hand, electricity demand is characterized by its volatility, while at the other it is subject to numerous factors, such as weather conditions, grid's extension, and more recently renewable energy sources (RES) penetration. Especially in a high voltage/medium voltage (HV/MV) distribution substation level, the data measurements combine both demand and generation, and thus the prediction becomes more complex by the ever-increasing installation of renewables. Therefore, accurate, reliable, and robust load forecasting will contribute towards appropriate operation and scheduling/planning for power systems, thus achieving a lower operating cost and higher reliability of the electricity supply. In particular, short-term load forecasting (STLF), with a time horizon that does not exceed a few hours, is of great importance, participating in a multitude of

The associate editor coordinating the review of this manuscript and approving it for publication was Bin Liu.

crucial operation tasks like economic emission dispatch, unit commitment and energy transactions [2], [3]. Moreover, providing reliable short-term forecasts, has a significant impact on the secure operation of power systems [4]. In dealing with such issues, load shifting plays a leading role and can only be accomplished by exploiting the STLF.

From a methodological point of view, STLF has been tackled with a plethora of machine learning (ML) methods. Due to their simplicity, models based on linear regression (LR) [5], [6] have been widely employed. On the other hand, the need for more sophisticated approaches, has led to the use of support vector regression (SVR) [7], [8], as well as neural networks and deep learning [9], [10]. However, it should be noted that despite years of thorough study, electric load forecasting still remains an open field of active research. The main reason behind this, is that it is a complex problem which is difficult to address adequately, due to the versatile nature of electric load [11]. As a result, there is no specific ML method which achieves universal superiority in terms of performance. For example, neural-network-based models have been found to be able to tackle satisfactory the non-linear behavior of load, but they are also sensitive to noisy input data. On the other hand, linear regression models provide much higher robustness, even though they are unable to capture more complex, nonlinear dynamics. Having this in mind, it is imperative to explore alternative methods that incorporate generative models, achieving a fair trade-off between the aforementioned merits.

In recent years an assortment of ML methods has emerged, which harness their power from the theory of sparse representation (SR) and corresponding dictionary learning (DL) [12], [13]. Sparse coding refers to the modeling of data signals as a linear combination of a few basis elements (atoms), which are part of a larger group (dictionary). Typical issues which have to be addressed are [14]: a) the dictionary learning stage which is responsible for discovering the elements of the dictionary (atoms) given a training set of electric load sequences and b) sparse representation or coding which provides the set of (few) coefficients which participate to the reconstruction of any questioned signal, given that the dictionary is now provided. These techniques have been proven to provide an extraordinary powerful solution to a wide range of application fields, especially in signal processing, image processing, machine learning, and computer vision [15]–[17]. Not surprisingly, their use continues to expand to additional scientific disciplines such as medical imaging and more recently, power supply systems. The use of sparse representation methods for modeling and forecasting individual household electricity loads is studied in [18] by proposing an alternating direction method of multipliers (ADMM) algorithm for solving the dictionary learning problem. A number of research papers based on sparse Bayesian learning (SBL) have been also published during the last decade, featuring weighted SBL [19], [20] or combined kernels SBL [21]. Moreover, besides load forecasting, several power grid-related problems have been addressed using sparse coding approach. In [22], real-time wind-power forecasting is achieved by online nonnegative sparse coding with elastic net regularization. The efficient and well-known algorithm K-SVD [23] has also recently been used in the smart grid framework [18], either for disaggregating a building's energy into the energy consumed by individual appliances [24], or compressing data from individual smart meters and extracting partial usage – sparse – patterns [25]. The daily consumption load forecasting based on a large amount of smart meters data, is studied in [26], where sparse encoders are proposed for feature extraction and dimensionality reduction of aggregated load data. However, according to the relative literature, the development of sparsity-based models in order to predict the electric load is still limited. It should be noted that up to the authors' best knowledge, load forecasting of high voltage/medium voltage (HV-MV) electricity data through sparse-based models has not been investigated yet.

The proposed work addresses STLF using a powerful model based on structured sparsity. Exploring the contribution and effectiveness of sparse coding to the problem at hand, further parsimonial DL/SR regression style-oriented methods are implemented. Dictionary learning as well as sparse representation is performed by well-defined methods which include among others popular ℓ_0, ℓ_1, ℓ_2 and ℓ_0^{tree} based regularization norms. Once the dictionary has been obtained, evaluation of the proposed parsimonial regression follows, in order to find the optimal coefficient vector among a large number of possible solutions. Experiments show clearly that the ℓ_0^{tree} outperforms in terms of accuracy any other SR-based, as well as standard, ML methods such as multilayer perceptrons (MLPs) or SVR.

A. CONTRIBUTIONS

In accordance with the aforementioned discussion, our primary interest is to introduce methods and algorithms for sparse representation and dictionary learning, aiming to provide an efficient and robust solution to the problem of short-term load forecasting. Intuitively, parsimony is a biologically motivated perception which has been explored in several machine learning applications with benefits to various scientific disciplines [27], [28]. Therefore, sparse representation and associated dictionary learning is a parsimonious inspired method which provides a robust signal representation and delivers superior qualitative, as well as quantitative, performance compared to methods based e.g., on orthonormal transforms. Under this concept, signals are approximated by means of a linear combination of only a few atoms, which are members of an overcomplete set (or dictionary). We will provide in detail the way that sparse coding and dictionary learning can be realized in the induced domain and we will devise a method for embedding a form of parsimonial regression within the context of short-term electric load forecasting. It should be pointed out that the proposed parsimonial regression is a methodology of low complexity that supposes linear correlation between sequences of load data, but on the other

hand, in contrast to other ML methods, it is more robust to overfitting.

Although the presence of non-linear characteristics in electric load time-series is a fact, one must clearly acknowledge that, the existence of non-linearities becomes more obvious for long-term prediction horizons; on the other hand, for short-term forecasts, the correlation between the output and input variables can often be approximated by linear models [29]. We are particularly interested whether hierarchical priors can be considered more effective a) in supervised setups or b) in the matrix-factorization framework, which will be used in this work. Additionally, in numerous cases the nature of the problem under consideration has an impact towards the existence of parameters (or variables) with strong associations among them, something that should be taken into account when dictionary elements are formed. Thus, it seems reasonable to make use of this knowledge regarding the problem in order to shape the corresponding sparse space. Therefore, we can explore block structure methods which impose a partition in the dictionary elements by defining groups which correspond to different types of features. In other words, we are permitting the simultaneous activation of specific coefficients that are part of a group, which will advance the dictionary elements to self-organize patterns in order to adapt the prior. This reinforces the exploitation of sparse-based models for short-term net load forecasting as highlighted by the results presented below, especially for those that utilize a structured type of sparse representation. Concluding, the proposed work has a significant contribution towards the following directions:

1. Formulation: A novel approach is introduced in order to acquire predictions during the sparse decomposition process. In other words, we create a simple or structured dictionary from the training set of electric load sequences and then we propose the use of a truncated dictionary in order to model any questioned sequence by means of the corresponding truncated coefficients, which in their turn will provide the desired forecasted load value.

2. Accuracy: The proposed method provides the means for embedding to the produced models a priori information of the problem under consideration through adopting a hierarchical structure. This approach is shown to be highly effective, as it manages to outperform not only baseline linear models, but also non-linear methods like SVR and MLP, in terms of mean absolute error (MAE), mean squared error (RMSE) and R^2.

3. Problem: According to the author's knowledge, this is the first time that the proposed DL/SR architecture has been used in order to cope with short-term load forecasting (HV-MV scale). Due to large RES exploitation, the development of an accurate forecasting model is getting significantly difficult, as the predictions are affected directly both by demand and generation. Given the fact that parsimonial methods are considered generative (i.e. explain the way that data is synthesized) machine learning approaches [15], we anticipate that the dictionary learning stage will provide robust modeling and generalization of the ambient sequence space which will be mapped to the corresponding coefficient vector and thus will provide enhanced prediction efficiency. That is, generative models help towards a realization of a methodical path for algorithm design, while in addition they provide a theoretical analysis of their performance.

Without doubt, the most significant issue that one must address when sparse representation algorithms are utilized, is the motivation behind its use on several types of signals. To simply state it, is there something appealing if one pursues a line of research based on sparse representation? The answer to this question is that sparse representation models have been found to be able to adapt universally, faithfully and effectively to the raw data exposed to them, due to the vast number of potential supports enabled by the dictionary. Therefore, sparse representation models are considered unique due to the interaction that occurs among theory, algorithms, and applications [15].

The paper is organized as follows: Section II provides a description as well as details of the task at hand and summarizes the proposed approach. Section III includes an overview and the background for sparse coding and associated dictionary learning methods and techniques, while at the same time it addresses details of the proposed load forecasting implementation. Section IV presents the experimental protocol and delivers the corresponding results, followed by an extensive discussion in section V. Finally, conclusions are drawn in Section VI.

II. PROBLEM DESCRIPTION

The load time-series to be forecasted has traditionally consisted of the summation of power demand from individual consumers. This power demand has shown little volatility and has substantial periodicity on a daily and monthly basis; therefore, in the past it was modelled using traditional methods, with relatively high accuracy [3]. Today, this load time-series also contains the summation of the distributed renewable energy sources generation that is downstream of the centralized generation. This new RES-based generation paradigm has jeopardized the accuracy of traditional prediction methods due to the intermittency of the weather phenomena. This is detrimental for two reasons: First, from an energy management perspective, uncertainty in distributed RES generation compromises the ability to effectively plan short-term power generation scheduling [30], while from a RES-aggregator perspective, stochasticity constrains their bidding strategies and thus, compromises their profit margins [31]. These shortcomings prove that accurate short-term predictions are of paramount importance in the context of multiple operational aspects of the smart grid. To conclude, a successful prediction model for the mixed power-load time-series should combine accurate modelling of load periodicity, as well as accommodating for volatility of RES generation.

A. SHORT-TERM LOAD FORECASTING

The short-term load forecasting of a medium voltage distribution network with heavy RES penetration refers to

the prediction of mixed power-load readings, which in turn correspond to the net active power (AP) demand of the distribution grid from the HV/MV substation. The present study is concerned with the hourly net AP forecasting, which constitutes a primal task in optimizing several grid operations, such as economic dispatch, as well as ancillary services pertaining to voltage and frequency control. More specifically, STLF models are included in the formulation of the reactive power optimization problem, which aims to minimize power losses and voltage deviations by toggling tap changer positions and capacitor bank switches [2]. In the case of an isolated network, there is also the need for load balancing (and consequently, frequency control), which is executed through energy management systems that also use short-term load forecasts. Lastly, STLF has to be an integral part during the development of electricity market clearing models [32]. It is therefore understood that it is difficult to achieve the efficient operation of an electrical network with the current requirements and specifications, without the existence of accurate short-term load forecasts.

The available data, besides the measured load also contain weather data, as measured from the substation's weather station. Normally, no data is shared between the substation and the distributed RES locations, meaning that individual RES generation readings are unavailable for inclusion in prediction model creation. The aforementioned shortcoming coupled with the fact that weather data measured at the substation's weather station will hardly be representative of the grid-wide conditions, significantly contributes to input data noise, which in turn narrows the pool of suitable modelling methods. To conclude, in order to create a short-term prediction model of the net active power demand of the grid, one could use as inputs the substation's historical timeseries of load and weather conditions.

B. CASE STUDY DETAILS

The case study data have been collected from an HV/MV substation in mainland Europe, from the timespan September 2017-December 2018. The recorded measurements refer to mixed power-load values, due to multiple photovoltaic (PV) systems contained in the MV distribution electricity network. These values correspond to the net active power demand of the distribution grid from the transmission network. The measurements in question have been sampled with one-minute intervals, showing remarkably denser sampling than in most similar applications and physically contain the net active power demand, as well as data from the substation's weather station, namely cloud coverage, wind speed, humidity, and temperature.

A primary issue during raw data preprocessing is dealing with missing and corrupt data, which result from sensor downtime mainly because of malfunction or maintenance works. For the scope of this case study, the preprocessing stage includes the removal of corrupted data and outliers, while missing data are ignored. For reasons of easier applicability and size reduction of the available dataset, the creation of an automatic detection routine for problematic data was preferred rather than manual extraction. In contrast to corrupted values, which are indicated by the low fluctuation of the net active power, outlier values were not as easy to detect. Several effective data handling techniques can be found in [33]. In the present case study, a rolling median window threshold approach is implemented, so as to conceptualize satisfactory outlier classification, while avoiding false positives (Fig. 1).

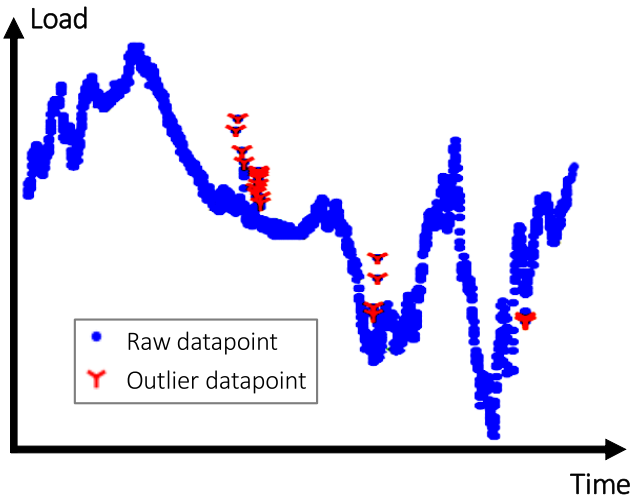

FIGURE 1. Operation of the rolling median threshold outlier detection routine.

It should also be noted that the objective of the model is to predict the electrical load one hour ahead, but it is also critical to be able to renew the predictions in a short time interval, i.e. without having to wait for a whole hour to obtain a new prediction. For this reason, we followed a resampling procedure, where each sample was calculated as the average of a 15-minute interval. Thus, the model can provide a prediction every 15 minutes, each time predicting the value of the electrical load after one hour.

An important task in load forecasting model creation is that of input variables selection. Since the desired horizon is considered to be short-termed, we anticipate that the use of historical load as well as weather data can be considered sufficient in order to build an effective forecasting model. We begin our analysis by providing details regarding the nature, as well as the role of all relative input and output training variables, common for all models created in this study with the help of Table 1 analytically. More details regarding the terminology used throughout this work are also provided here:we use bold capital letters (e.g., $\mathbf{X} \in R^{d \times N}$) and bold lower-case letters (e.g., $\mathbf{x}$ or $\mathbf{x}^{(k)} \in R^{d}$) to denote entire matrices and d-dimensional column vectors. Let us also use the notations $[\cdot]_i$ and $[\cdot]_{i,j}$ in order to represent the corresponding element indices of an one-dimensional vector, or a two-dimensional matrix. Furthermore, in order to provide a connection between the terminology and the problem at hand, let us denote with $\mathbf{x}^{(t)} \in \mathbb{R}^{11}$ any agnostic eleven-valued

TABLE 1. Description of training variables of the forecasting models.

Variable type	Index	Train variable	**W**eather or **P**ower	Expression	Description
INPUTS	*1*	$[\mathbf{x}^{(t)}]_1$	$[\mathbf{w}^{(t)}]_4$	$[\mathbf{t}^{(t+4)}]_1 - [\mathbf{t}^{(t)}]_1$	Temperature: Difference between current and one hour ahead values
	2	$[\mathbf{x}^{(t)}]_2$	$[\mathbf{w}^{(t)}]_3$	$[\mathbf{h}^{(t+4)}]_1 - [\mathbf{h}^{(t)}]_1$	Humidity: Difference between current and one hour ahead values
	3	$[\mathbf{x}^{(t)}]_3$	$[\mathbf{w}^{(t)}]_2$	$[\mathbf{s}^{(t+4)}]_1 - [\mathbf{s}^{(t)}]_1$	Wind speed: Difference between current and one hour ahead values
	4	$[\mathbf{x}^{(t)}]_4$	$[\mathbf{w}^{(t)}]_1$	$[\mathbf{c}^{(t+4)}]_1 - [\mathbf{c}^{(t)}]_1$	Cloud coverage: Difference between current and one hour ahead values
	5	$[\mathbf{x}^{(t)}]_5$	$[\mathbf{p}^{(t)}]_6$	$(1/4)\sum_{n=0}^{3}[\mathbf{p}^{(t-n)}]_1$	Active power: Average value of the last hour (i.e. four quarters of an hour)
	6	$[\mathbf{x}^{(t)}]_6$	$[\mathbf{p}^{(t)}]_5$	$[\mathbf{p}^{(t)}]_1 - [\mathbf{p}^{(t-8)}]_1$	Active power: Difference between current and two hours ago values
	7	$[\mathbf{x}^{(t)}]_7$	$[\mathbf{p}^{(t)}]_4$	$[\mathbf{p}^{(t)}]_1 - [\mathbf{p}^{(t-4)}]_1$	Active power: Difference between current and one hour ago values
	8	$[\mathbf{x}^{(t)}]_8$	$[\mathbf{p}^{(t)}]_3$	$[\mathbf{p}^{(t)}]_1 - [\mathbf{p}^{(t-1)}]_1$	Active power: Difference between current and 15 minutes ago values
	9	$[\mathbf{x}^{(t)}]_9$	$[\mathbf{p}^{(t)}]_2$	$[\mathbf{p}^{(t-96)}]_1$	Active power: Twenty-four hours ago value
	10	$[\mathbf{x}^{(t)}]_{10}$	$[\mathbf{p}^{(t)}]_1$	$[\mathbf{p}^{(t)}]_1$	Active power: Current value
OUTPUT	*11*	$[\mathbf{x}^{(t)}]_{11}$	$[\mathbf{p}^{(t+4)}]_1$	$[\mathbf{p}^{(t+4)}]_1$	Active power: One hour ahead value (known at the training stage)

$\mathbf{x}$ represents the agnostic feature values, while $\mathbf{w}$, $\mathbf{t}$, $\mathbf{h}$, $\mathbf{s}$, $\mathbf{c}$, $\mathbf{p}$ represent the weather and active power measurements

training sequence identified at any specific time (t: the current time in steps of fifteen minutes intervals), comprised by the following values: a) one output variable $[\mathbf{p}^{(t+4)}]_1 \in \mathbb{R}$, i.e. the net power load one hour (i.e. four, fifteen-minute intervals) ahead, b) Six $\mathbf{p}^{(t)} \in \mathbb{R}^6$ values with the current and historic net power load and c) four $\mathbf{w}^{(t)} \in \mathbb{R}^4$ values from the weather measurements. In addition to the above representation, Table 1 denotes the current time index with t, the time index for one hour ahead, with $t + 4$ and the time index for one hour and one day ago with t-4 and t-96, respectively. The total number of load and weather variables during training is denoted by the variable d(equals to eleven) and finally N corresponds to the number of existing training sequences. According to the aforementioned terminology, the six $[\mathbf{p}^{(t)}]_n$, $n \in [1:6]$ values contain the current, past, average and difference measures of the active power values, the $[\mathbf{w}^{(t)}]_m$, $m \in [1:4]$ components contain the respective weather related inputs and finally, we denote with $[\mathbf{c}^{(t)}]_1$, $[\mathbf{s}^{(t)}]_1$, $[\mathbf{h}^{(t)}]_1$ and $[\mathbf{t}^{(t)}]_1$ the one-valued ($\in \mathbb{R}$) measurements of cloud coverage, wind speed, humidity and temperature, respectively. It is important to mention that during the training stage of the forecasting model, the weather inputs $[\mathbf{w}^{(t)}]_{1:4}$ are introduced as measured values of actual weather data acquired at the t time index. On the contrary, in an online implementation of the model, future weather data $[\mathbf{w}^{(t+futuretimestep)}]_{1:4}$ will be unknown and replaced by weather predictions, therefore introducing an additional uncertainty.

The use of inputs 9-10 described in Table 1 are justified as the strong dependency of electric load time-series to current value is commonly reported in the literature, as is also the use of the previous day value [34]. A common tactic [35] is also to involve the trend of electrical load, introducing the differences between current and previous AP values, as in inputs 6-8, while the implementation of past values average, as declared by input 5, is also quite important according to the literature [36]. Finally, the introduction of weather features, represented by inputs 1-4, has been found to have an improving effect on the predictions [37], [5]. The resulting values are considered to form the agnostic ambient space which will be used in order to train the dictionaries and provide the sparse codes.

After the pre-processing, resampling and input selection operations are completed, the training subset has to be selected. Following a careful examination of the available data, it was concluded that data in the interval September – November 2017 are more appropriate for the training dataset, leaving the rest of the data available for testing. The training data selection was based on the fact that data in this interval presented the least possible amount of missing data and outliers compared to other intervals; furthermore, this selection allows to keep a whole year available for testing, which helps to better assess the generalized forecasting abilities of the model in different seasons of the year. This is an important issue, as the time series consists of a load and a generation component and the statistical properties of these two components are not static through the year, mainly due to weather variability. Finally, a point of high significance is that no data permutation is implemented before splitting the available data to the training and testing datasets. Thus, the data used for testing are completely unknown to the proposed model, enhancing the reliability of predictions.

III. PROPOSED METHODS

This section provides an overview and the background for sparse coding and associated dictionary learning methods and techniques. At this point, some additional annotations

have to be defined. One-dimensional column vectors of zeros and ones are also represented as $\mathbf{0}_d \in R^d$ and $\mathbf{1}_d \in R^d$, respectively. Let us also denote with ℓ_0 and ℓ_q (for $q \geq 1$) the corresponding norms of a column vector: $\ell_0(\mathbf{x}) =, \|\mathbf{x}\|_0 \triangleq \#\{j \text{ s.t. } [\mathbf{x}]_j \neq 0\}$, $\|\mathbf{x}\|_q \triangleq \left(\sum_{i=1}^{m} |[\mathbf{x}]_i|^q\right)^{1/q}$. Finally, the Frobenius norm of a matrix is denoted as $\|\mathbf{X}\|_F \triangleq \left(\sum_{i=1}^{m}\sum_{j=1}^{n}\left|([\mathbf{X}]_{i,j})^2\right|\right)^{1/2} = \sqrt{Tr(\mathbf{X}^T\mathbf{X})}$ with $Tr(\cdot)$ the trace operator and T the transpose operator.

A. PROBLEM FORMULATION

Sparsity, implemented with the corresponding SR methods, is a concept (accompanied by the associated widespread machine learning models), whose purpose is to reconstruct in an efficient way, one or a set of signals, with only few non-zero coefficients of an appropriate basis (or dictionary). This basis is usually equipped with the overcomplete property, which states that the number of dictionary elements (or atoms) should be greater than the ambient space dimensionality. Sparse methods encode any given query observation by providing a solution which activates only few components of the dictionary as possible. In the previous years, a number of seminal research papers provide evidence that sparsity is a prevalent method for scientific disciplines like signal and image processing, computer vision and machine learning, information theory, neuroscience and related areas [38]–[42]. It has been also alleged that the V1 part of the brain is performing in a similar way under the constraints of the sparsity objective [27].

We are now ready to introduce our sparse coding formulation for short-term load forecasting according to the material exposed in section IIB. Let us denote with $\mathbf{x}^{(k)} \in \mathbb{R}^{11}$ a k indexed vector train sequence whose last (most recent) value $[\mathbf{x}^{(k)}]_{11}$ corresponds to the electricity load demand $EL(k) = [\mathbf{x}^{(k)}]_d$ at any k timeslot (e.g. "current load at any k^{th} time instance"), while $[\mathbf{x}^{(k)}]_i$, $i \in [1 : 10]$ represent previous (i.e. historical) instances of the electricity load for a specific substation. Following, we define a set of columns-sequences (i.e. a frame) by concatenating groups of N-length $\mathbf{x}^{(k)}$ sequences, $\mathbf{X} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, ..\mathbf{x}^{(N)}]$: $\mathbf{X} \in \mathbb{R}^{d\times N}$.

Ideally, if one is equipped with an overcomplete dictionary $\mathbf{D} = [\mathbf{d}^{(1)}, \mathbf{d}^{(2)}, \ldots, \mathbf{d}^{(K)}] \in \mathbb{R}^{d\times K}$ (which can be derived from the frames $\mathbf{X}$) and a query sequence $\mathbf{q} \in \mathbb{R}^d$, the formulation of SR is usually articulated by the following equivalent forms of either regularized, or constrained optimization problem, expressed by (1) and (2), respectively:

$$\min_{\boldsymbol{\alpha}}\left(\frac{1}{2}\|\mathbf{q} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\psi(\boldsymbol{\alpha})\right) \tag{1}$$

$$\min_{\boldsymbol{\alpha}}\left(\frac{1}{2}\|\mathbf{q} - \mathbf{D}\boldsymbol{\alpha}\|_2^2\right), \quad \text{s.t: } \psi(\boldsymbol{\alpha}) \leq \rho \tag{2}$$

The coefficient vector $\boldsymbol{\alpha} \in \mathbb{R}^K$ of (1) and (2), represents the sparsely distributed coefficients for the $\mathbf{q} \in \mathbb{R}^d$ query input signal, $\|\mathbf{q} - \mathbf{D}\boldsymbol{\alpha}\|_2^2$ is the representation of the reconstruction error between the query input signal $\mathbf{q}$ and its sparse representation $\mathbf{D}\boldsymbol{\alpha}$, the parameter lambda ($\lambda$) denotes the regularization parameter (or Lagrange multiplier) and finally, the embedded $\psi(\cdot)$ penalizing function denotes the sparsity-inducing term. In (2) the parameter ρ is used in order to represent a direct measure of the sparsity level (i.e. number of non-zero coefficients) of the representation vector $\boldsymbol{\alpha}$.

Typically, $\psi(\cdot)$ utilizes the ℓ_p norm, defined for $1 \leq p \leq \infty$, of the coefficient vector $\boldsymbol{\alpha}$, i.e. $\psi(\boldsymbol{\alpha}) = \ell_p(\boldsymbol{\alpha}) = \left(\sum_{j=1}^{K}([\mathbf{a}]_j)^p\right)^{1/p}$ for a specific value of p. Some of the most popular forms found in literature are the ones that utilize the ℓ_0 and ℓ_1 norms, respectively. In this work, we explore, in addition to the typical aforementioned norms, the utilization of other norms including the one which exploits atoms placed at the convex hull of the ambient feature space by means of archetypal analysis, as well as a ℓ_0^{tree} hierarchical tree-structured sparse regularization norm [43], which has been found to be useful in a number of cases exploring the correlation between previous instances of the electricity load. Although the aforementioned norms have been described thoroughly in the literature, for completeness of this paper we will provide critical details regarding their implementation in the following sections.

As stated earlier, sparse representation employs an overcomplete set of sequences (or atoms) of the dictionary $\mathbf{D}$. The formation of the dictionary is achieved through a specific learning procedure which tries to fit the dictionary members-atoms to the input-training data by means of linear combinations of few of them. These unsupervised learning methods evaluate the cost function over the reconstruction error with remarkable results on several machine learning disciplines. The dictionary learning is defined as a joint optimization problem with respect to dictionary $\mathbf{D} = \{\mathbf{d}^{(j)}\}, j = 1 : K$ as well as the coefficients $\mathbf{A} = \{\boldsymbol{\alpha}^{(k)}\}, k = 1 : N$ and it is usually expressed as follows:

$$\min_{\mathbf{D}\in C, \mathbf{A}\in\mathbb{R}^{K\times N}}\left(\frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2 + \lambda\psi(\mathbf{A})\right) \tag{3}$$

$$\min_{\mathbf{D}\in C, \mathbf{A}\in R^{K\times N}}\left(\frac{1}{2}\|\mathbf{X} - \mathbf{D}\mathbf{A}\|_F^2\right) \quad \text{s.t: } \psi(\mathbf{A}) \leq \rho \tag{4}$$

Again, the embedded $\psi(\cdot)$ denotes the sparsity-inducing term. The set C of the dictionary atoms is usually defined to be the convex set of matrices that covers the following constraint:

$$C \triangleq \{\mathbf{D} \in \mathbb{R}^{d\times K} \quad \text{s.t: } \forall_{j=1:K}, (\mathbf{d}^{(j)})^T\mathbf{d}^{(j)} \leq 1\} \tag{5}$$

Equation (5) enables the use of normalized atoms in contrast to arbitrary ones in order to evade the cases of evaluating small coefficient values. In this paper we also expand the dictionary learning stage, in addition to the typical embedding $\psi(\cdot)$ norms and C constraint of the dictionary $\mathbf{D}$, by a) addressing the idea of embedding the atoms of the dictionary in a directed tree and b) by providing an archetypal oriented solution in which the dictionary $\mathbf{D}$ of archetypes is subject to two dual geometrical constraints: The first one, states that any $\mathbf{x}^{(i)}$ vector must be approximated by a convex combination of some archetypes $\mathbf{d}^{(j)}$, while the other, ensures that each

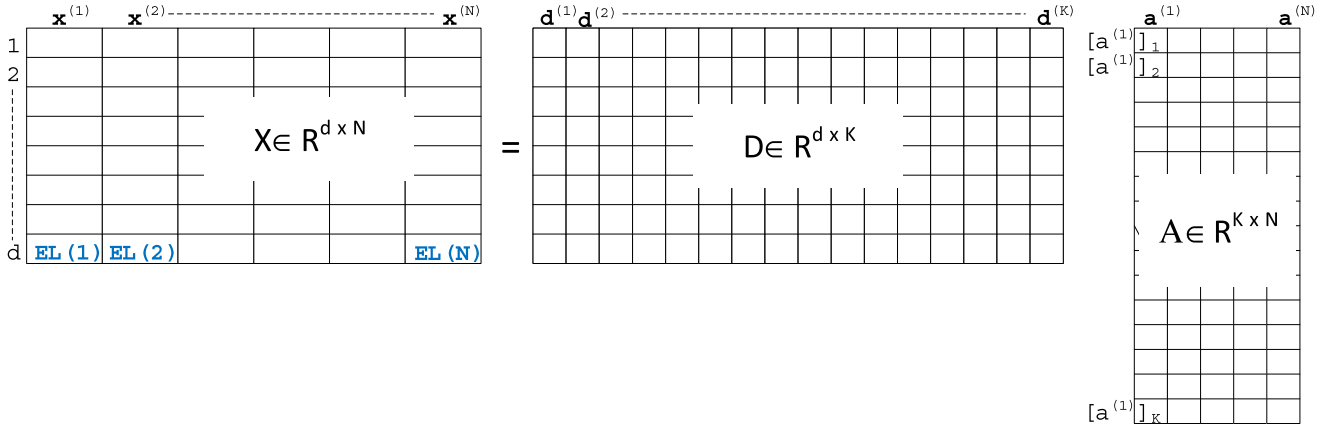

FIGURE 2. Graphical depiction of the dictionary learning process D from a training frame X of N-length historical electric load consumptions. In this example, the frame X consists of N eleven-dimensional sequences (d = 11). The dictionary D consists of K > 11 atoms and is learned throughout the dictionary learning algorithm by eqs. (3)-(5). The dictionary matrix D will be used for sparse representation in the testing stage.

j-archetype must be approximated by a convex combination of the $\mathbf{x}^{(i)}$ vectors.

Summarizing, the dictionary $\mathbf{D} \in \mathbb{R}^{d\times K}$ is learned by using (3)-(4) along with a) the typical dictionary constraints of (5) as well as the additional above-mentioned ones and b) appropriate forms for the embedding norms $\psi(\cdot)$. The input to the algorithms that implement (3)-(4) is the set $\mathbf{X} \in \mathbb{R}^{d\times N}$ of N-sized, d-dimensional sequences of electricity load for any substation. In order to provide an estimate (i.e. forecast) for the electricity load of one specific moment indexed at time k, $EL\,(k)$ we create a query input sequence $\mathbf{q} \in \mathbb{R}^{d-1}$ due to the fact that the desired electricity load $EL\,(k)$ now is an unknown which needs to be predicted. For this reason, we solve (1) or (2) by introducing a truncated dictionary $\mathbf{D}_{tr} \in \mathbb{R}^{(d-1)\times K}$ whose values are the first (d-1) rows of the original $\mathbf{D} \in \mathbb{R}^{d\times K}$ dictionary. Having in mind that the solution of (1) or (2) still provides a k-dimensional coefficient sparse vector $\boldsymbol{\alpha}_{tr} \in \mathbb{R}^{K}$, we provide a prediction for the electric load demand at the unknown time k, $EL\,(k)$ by applying the following reconstruction term:

$$EL\,(k) = \mathbf{D}\alpha_{tr} \tag{6}$$

Note that in order to apply (6), the atoms $\mathbf{d}_{tr}^{(j)}$ of the truncated dictionary $\mathbf{D}_{tr}$ must be re-normalized so that they lie on the unit ℓ_2 ball. Fig. 2 provides the general concept of sparse representation via a graphical depiction of the dictionary learning $\mathbf{D} \in \mathbb{R}^{d\times K}$ procedure which can be seen as a matrix factorization problem with some additional constraints. In addition, Fig. 3 depicts the proposed method for predicting the $EL\,(k)$ electric load by presenting a $\mathbf{q} \in \mathbb{R}^{d-1}$ vector with previous load values, the $\mathbf{D}_{tr} \in \mathbb{R}^{(d-1)\times K}$ which is one of the outcomes of Fig. 2, and the corresponding $\boldsymbol{\alpha}_{tr} \in \mathbb{R}^{K}$ sparse coefficients.

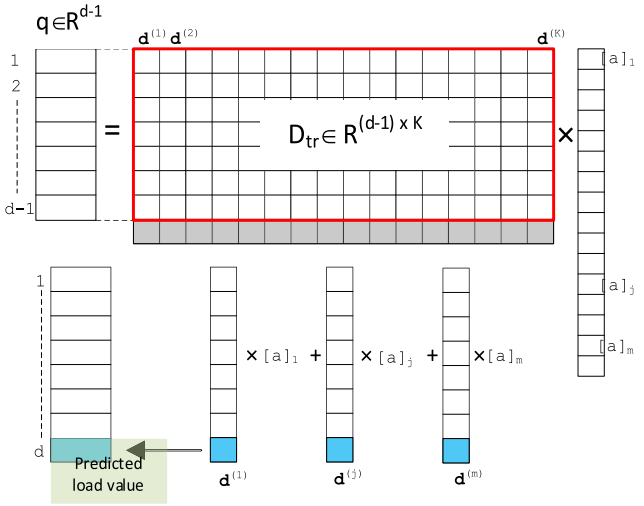

FIGURE 3. Proposed method for prediction of the electric load by a) utilizing a truncated dictionary $\mathbf{D}_{tr}$ for sparse coefficient evaluation and b) applying them to the initial dictionary D.

B. PROPOSED METHODS OVERVIEW

Table 2 summarizes in a glance, a bouquet of the methods which will be explored in this work for electricity load prediction. Specifically, we are exploring the following method-cost functions for sparse representation and dictionary learning, which will be used in eqs (3)-(5) for dictionary learning and (1)-(2) for sparse coding: a) the ℓ_0 norm implemented with the K-SVD and orthogonal matching pursuit (OMP) algorithms, b) the ℓ_1 norm implemented with the online and least-angle regression-least absolute shrinkage and selection operator (LARS-LASSO) algorithms, c) the ℓ_2 norm accompanied with simplex constraints on the atoms by means of archetypal analysis and d) an ℓ_0^{tree} norm with the atoms embedded in a tree structured group form $g(T)$. The following sections provide the necessary short descriptions of them.

1) THE ℓ_0 NORM: K-SVD/OMP

It is common ground that the use of the ℓ_0 norm leads to a combinatorial NP-hard optimization problem, whose solution can be achieved only through approximation methods. To this end, greedy based policies are employed in

TABLE 2. List of explored dictionary learning (and corresponding sparse representation) methods for electricity load prediction.

Norm	Key types of embedding function $\psi(\cdot)$, dictionary set C and additional properties of $\boldsymbol{\alpha}$
ℓ_0	$\min_{\mathbf{D}\in C,\mathbf{A}\in\mathbb{R}^{K\times N}}(0.5\|\mathbf{X}-\mathbf{DA}\|_F^2)$, s.t: $\|\mathbf{A}\|_0 \le \rho$
	$C \triangleq \{\mathbf{D}\in\mathbb{R}^{d\times K}$ s.t: $\forall_{j=1:K}$, $(\mathbf{d}^{(j)})^T\mathbf{d}^{(j)}=1\}$
ℓ_1	$\min_{\mathbf{D}\in C,\mathbf{A}\in\mathbb{R}^{K\times N}}(0.5\|\mathbf{X}-\mathbf{DA}\|_F^2+\lambda\|\mathbf{A}\|_1)$
	$C \triangleq \{\mathbf{D}\in\mathbb{R}^{d\times K}$ s.t: $\forall_{j=1:K}$, $(\mathbf{d}^{(j)})^T\mathbf{d}^{(j)}\le 1\}$
ℓ_2	$\min(\|\mathbf{X}-\mathbf{DA}\|_F^2=\|\mathbf{X}-\mathbf{XBA}\|_F^2)$
	$s.t.\ \mathbf{a}^{(i)}\in\Delta_K, \mathbf{b}^{(j)}\in\Delta_N$
	$\Delta_K \triangleq \left\{\mathbf{a}^{(i)}\in\mathbb{R}^K\ s.t.\ \mathbf{a}^{(i)}\ge 0\ and\ \sum_{j=1}^{K}[\mathbf{a}^{(i)}]_j=1\right\}$
	$\Delta_N \triangleq \left\{\mathbf{b}^{(j)}\in\mathbb{R}^N\ s.t.\ \mathbf{b}^{(j)}\ge 0\ and\ \sum_{i=1}^{N}[\mathbf{b}^{(j)}]_i=1\right\}$
	$\forall i\in[1:N],\ \&\ \forall j\in[1:K]$
ℓ_0^{tree}	$\min_{\mathbf{D}\in C,\mathbf{A}\in\mathbb{R}^{K\times N}}(0.5\|\mathbf{X}-\mathbf{DA}\|_F^2+\lambda\ell_o^{tree}(\mathbf{A}))$
	$C \triangleq \{\mathbf{D}\in\mathbb{R}^{d\times K}$ s.t: $\forall_{j=1:K}$, $(\mathbf{d}^{(j)})^T\mathbf{d}^{(j)}\le 1\}$

Eqs. (1)-(2) are used for the determination of $\boldsymbol{\alpha}$ while Eqs. (3)-(5) for dictionary learning

order to find and provide an optimized solution vector. The strategy is to pursuit the atom who has the strongest relation input sample, in order to minimize the reconstruction error in the least-squares sense. For this case, we are using the popular method under the name of the orthogonal OMP [44]. Given a dictionary $\mathbf{D}$ and any sample $\mathbf{x}^{(i)}$, OMP sequentially selects the atoms with the highest correlation to the respective sample's residual. At a step s: $0 < s \le \rho$, with ρ being the sparsity level, the algorithm for the selection of the most efficient atom is provided by $k_s = \arg\max_j \left|(\mathbf{d}^{(j)})^T\mathbf{r}_{s-1}\right|$, where $\mathbf{r}_{s-1}$ is the current residual. Then, upon the selection of an atom, the signal $\mathbf{x}^{(i)}$ is projected onto the span of currently selected atoms as: $\hat{\mathbf{a}}_s = \left(\mathbf{D}_{V_s}\right)^+\mathbf{x}^{(i)}$, in which $V_s = V_{s-1}\cup k_s$ represents the set of indices pointing at the currently selected dictionary atoms and $\mathbf{D}_{V_s}$ is the subset of dictionary indexed by V_s. Concluding, the value of the new residual is now provided by $\mathbf{r}_s = \mathbf{x}^{(i)} - \mathbf{D}_{V_s}\hat{\mathbf{a}}_s$ and the process now recaps until either ρ atoms are selected, or the residual magnitude minimizes. In this work we use the batch-OMP implementation [45], with the Cholesky decomposition. At the dictionary learning stage, ℓ_0-norm regularized DL is addressed with the KSVD/OMP algorithm. Given a set of training samples $\mathbf{X}$ and the corresponding set of sparse coefficients $\mathbf{A}^{(t)}$ evaluated with the use of OMP for the dictionary $\mathbf{D}^{(t)} = \left[\mathbf{d}_1^{(t)}, \mathbf{d}_2^{(t)}, \ldots, \mathbf{d}_K^{(t)}\right]$ at iteration t, our target is to derive an updated dictionary $\mathbf{D}^{(t+1)}$:

$$\min_{\mathbf{D}^{(t+1)}}\left\{\left\|\mathbf{X}-\mathbf{D}^{(t+1)}\mathbf{A}^{(t)}\right\|_F^2\right\} \quad \text{s.t.} \quad \left\|\mathbf{a}^{(i),(t)}\right\|_0 \le \rho \tag{7}$$

We define ω_k to be the group of indices which are pointing to those training samples that use the atom $d^{(k),(t)}$: $\omega_k = \left\{i|1\le i\le N, [\mathbf{A}]_{k,i}\ne 0\right\}$, such that $[\mathbf{A}]_{(k,:)}$ denotes the k^{th} row of the coefficient matrix $\mathbf{A}\in\mathbb{R}^{d\times N}$. Then, let $\Omega_k \in R^{K\times|\omega_k|}$ be the matrix of ones on the $(\omega_k(i), i)$ entries and zeros elsewhere. Following, a globally oriented error matrix $\mathbf{E}_k = \mathbf{X} - \sum_{j\ne k}\mathbf{d}^{(j),(t)}[\mathbf{A}]_{j,:}$ represents the information that cannot be explained with this representation. Therefore, the k^{th} atom can be optimized so as to better represent this information. Summarizing, K-SVD updates the k^{th} atom by applying singular value decomposition (SVD) on the matrix $\mathbf{E}_k^{\omega}$, where $\mathbf{E}_k^{\omega} = \mathbf{E}_k\Omega_k$, by searching for the closest rank-1 matrix so that $\mathbf{E}_k^{\omega} = \mathbf{U}\Delta\mathbf{V}^T$. The updated value $\mathbf{d}_k^{(t+1)}$ of the k^{th} atom is defined as the first column of $\mathbf{U}$, and the new coefficient vector $\mathbf{a}_{\omega}^{(k)}$ as the first column of $\mathbf{V}$ multiplied by $[\Delta]_{1,1}$. This process is repeated for each atom, producing an updated dictionary. The new dictionary is now utilized by the OMP algorithm in order to derive the updated sparse coefficients $\mathbf{A}^{(t+1)}$ etc.

2) THE ℓ_1 NORM: ONLINE/LARS-LASSO

The ℓ_1 norm, is used extensively for solving sparse representation problems given the fact that it can provide an analytical solution in polynomial time. For example, popular solvers include the basis pursuit as well as the LASSO [46]. Contrary to the coordinate descent methods, which assume that the dictionary atoms exhibit low correlation, we apply homotopy-based methods like the LARS-LASSO algorithm [47]. The LARS-LASSO algorithm calculates the solution path by repeatedly decreasing the value of λ and using as a warm-restart the previously calculated solution. It has been reported that the uniqueness of the solution for a specific value of the parameter λ, is ensured and thus it can be proved that the solution path is piecewise linear [12]. This property is very important since the algorithm follows the direction of each segment until it reaches a critical point, i.e. where either a non-zero element becomes zero (so it is removed from the active set of coefficients), or a new non-zero element is added to the active set of coefficients. As a result, the homotopy method initializes with an empty set of coefficients, and iteratively updates it by one variable at a time. The complexity of the method relies in reversing the covariance matrix of the selected atoms at each critical point in order to update the active set of coefficients, which is performed by the Cholesky decomposition, or the Woodbury formula. In our proposal we have adopted the LARS-LASSO algorithm. For this specific ℓ_1 oriented convex relaxation approach, the online learning method along with the LARS-LASSO algorithm have been employed for the corresponding tasks of dictionary learning.

3) THE ℓ_2 NORM: ARCHETYPAL ANALYSIS

Archetypal analysis is an innovative unsupervised learning method related to generative data analysis methods such as sparse coding [48]. Archetypes are a special case of dictionary elements, due to the fact that they are mainly placed in

the convex hull of the ambient space. Archetypal analysis is all about learning a factorial representation of $\mathbf{X}$ by addressing a corresponding archetypal problem. It searches for a set $\mathbf{D}$ of archetypes in which each $\mathbf{d}^{(j)}$ vector also belongs to the ambient space, but their formation is constrained under two dual geometrical restrictions. The first one, states that any $\mathbf{x}^{(i)}$-sequence must be approximated adequately by a convex combination of some archetypes $\mathbf{d}^{(j)}$, while the second one states that each $\mathbf{d}^{(j)}$ archetype must also be approximated by a convex combination of the $\mathbf{x}^{(i)}$ sequence vector. Thus, given a set of archetypes $\mathbf{D}$, each $\mathbf{x}^{(i)}$ sequence should be approximated by $\mathbf{x}^{(i)}_{apprx} = \mathbf{D}\mathbf{a}^{(i)}$, where $\mathbf{a}^{(i)} \in \mathbb{R}^K$, is a coefficient column vector which resides in the simplex Δ_K [49]:

$$\Delta_K \triangleq \left\{ \mathbf{a}^{(i)} \in \mathbb{R}^K \; s.t. \; \mathbf{a}^{(i)} \geq 0 \; and \; \sum\nolimits_{j=1}^{K} [\mathbf{a}^{(i)}]_j = 1 \right\} \quad (8)$$

and each archetype $\mathbf{d}^{(j)}$ must be approximated by the product $\mathbf{X}\mathbf{b}^{(j)}$, with $\mathbf{b}^{(j)} \in \mathbb{R}^N$ is another coefficient column vector which resides in the simplex Δ_N:

$$\Delta_N \triangleq \left\{ \mathbf{b}^{(j)} \in \mathbb{R}^N \; s.t. \; \mathbf{b}^{(j)} \geq 0 \; and \; \sum\nolimits_{i=1}^{N} [\mathbf{b}^{(j)}]_i = 1 \right\} \quad (9)$$

Then the above problem is expressed by the following minimization of the residual sum of squares (RSS):

$$\min_{\mathbf{a}_i \in \Delta_p \text{ for } i=1:N} \sum_{i=1}^{N} \left\| \mathbf{x}^{(i)} - \mathbf{D}\mathbf{a}^{(i)} \right\|_2^2 \quad (10)$$

where $\forall j, \mathbf{d}^{(j)} = \mathbf{X}\mathbf{b}^{(j)}$, which is equivalent to:

$$\min_{\substack{\mathbf{a}^{(i)} \in \Delta_K \text{ for } i=1:N \\ \mathbf{b}^{(j)} \in \Delta_N \text{ for } j=1:K}} \|\mathbf{X} - \mathbf{X}\mathbf{B}\mathbf{A}\|_F^2 \quad (11)$$

with $\mathbf{A} \in \mathbb{R}^{K \times N}$, $\mathbf{B} \in \mathbb{R}^{N \times K}$ and $\mathbf{D} = \mathbf{X}\mathbf{B}$. Equation (10) is a non-convex optimization problem, but we observe that it is convex with respect to one of the variables $\mathbf{A}$ or $\mathbf{B}$, when the other variable is kept fixed. This allows us to enable a block-coordinate descent scheme which guarantees in an asymptotic way a stationary point of the problem. It should be noted here that the main difference between sparse coding and archetypal analysis, aside from the non- negativity of $\mathbf{a}^{(i)}$, is that the archetypes must be convex mixtures of the data points $\mathbf{X}$, and in a similar way, $\mathbf{b}^{(j)}$ must be also constrained to exist in the simplex Δ_N, facts that inherently attribute them a sparsified nature. The solution of the problem in Eq. (10) can be addressed efficiently by an active set algorithm.

4) THE ℓ_0^{tree} NORM: HIERARHICAL SPARSITY

In the proposed work we extent the typical sparse representation framework by additionally introducing a sparsity-inducing norm $\psi(\cdot)$ whose objective is to embed the atoms of $\mathbf{D}$ and consequently the solution coefficients $\mathbf{a}$ into a fixed hierarchical tree structure. Following [39] let us consider a tree T with K nodes: $T = \{j\}, j = \{1, \ldots, K\}$. We are interested in specific patterns of the non-zero coefficients of $\boldsymbol{\alpha}$ which are constrained under the assumption that they are part of a connected and rooted subtree of a tree structure T. We provide an example in order to make clear the above-mentioned statement in Fig. 4. Given a solution $\mathbf{a} \in \mathbb{R}^8$, we define the ancestors(j) of a node j to be formed by the subset of indices which correspond to the ancestors of the node j. Then, the solution vector $\mathbf{a}$ is subjected to the condition: $\mathbf{a}_j \neq 0, \Rightarrow [\mathbf{a}_k \neq 0]$ for all k in ancestors (j), which impose the following rule: the contribution of the atom $\mathbf{d}^{(j)} \in \mathbb{R}^d$ in the reconstruction of the signal $\mathbf{x}$, is allowed only if it's ancestors (j) are also part of the contributing subgroup. Jennaton *et al.* propose a complementary description by declaring the descendants(j) as follows: $\mathbf{a}_j = 0, \Rightarrow [\mathbf{a} = 0]$, for all k in descendants(j), which intuitively states that if an atom $\mathbf{d}^{(j)} \in \mathbb{R}^d$ does not contribute actively to the representation of $\mathbf{x}$, then neither should its descendants within tree T [50]. Now, we denote as $T \triangleq \{\text{descendants}(j)\}$, with $j = 1 : p$, the set of elements with the descendants (j) of each node. Each element g of T is referred as a group. Then, we can penalize the number of groups g in $\mathbf{G}$, that contribute to the representation of $\mathbf{x}$, by using a tree norm $\psi(\boldsymbol{\alpha}) l_0^{tree} \triangleq (\boldsymbol{\alpha})$, which records at least one non-zero coefficient of $\boldsymbol{\alpha}$ as:

$$\psi \triangleq (\boldsymbol{\alpha}) l_0^{tree}(\boldsymbol{\alpha}) = \sum_{g \in T} \delta^g(\mathbf{a}), \text{ with}$$

$$\delta^g = \begin{cases} 1 & \text{if there exists } j \in g \text{ such that } \boldsymbol{\alpha}_j \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

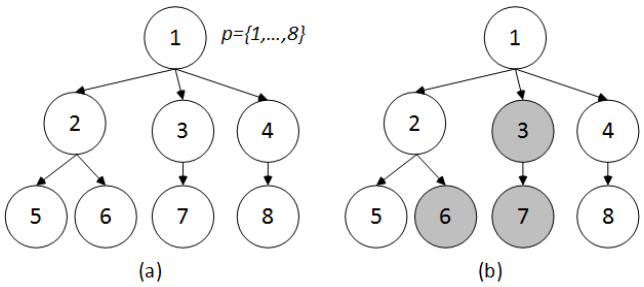

FIGURE 4. (a) An example of a solution vector that is embedded within a tree structure T. (b) Solution with non-zero valued (non-shadowed) nodes in which the ancestor property applies: $\alpha_8 \neq 0 \Rightarrow \alpha_{(\text{ancestors}(8))} \neq 0$.

Then, (1) combined with the constraint of (12), results to the following nonconvex formulation:

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^K} \left(\frac{1}{2} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda \sum_{g \in T} \delta^g \right) \quad (13)$$

Equation (13) is solved by approximation from dyadic partitions which provide a solution to:

$$\min_{\boldsymbol{\alpha} \in R^K} \left(\frac{1}{2} \|\mathbf{u} - \mathbf{a}\|_2^2 + \lambda \sum_{g \in T} \delta^g(\mathbf{a}) \right) \quad (14)$$

where $\mathbf{u} \in \mathbb{R}^K$ is a fixed signal, and the other parameters retain their previous notions. This problem is addressed with the introduction of a proximal operator for the nonconvex regularization term $\lambda \cdot \ell_0^{tree}(\boldsymbol{\alpha})$. It has been shown in the literature that (14) can be solved efficiently by performing a sequence

of thresholding operations on the variable **a** with the iterative shrinkage thresholding algorithm (ISTA) [39]. Dictionary learning, in the context of the proposed hierarchical sparse coding framework, is articulated with the use of:

$$\min_{\mathbf{D}\in C,\mathbf{A}\in\mathbb{R}^{K\times N}} \left(0.5\left\|\mathbf{X}-\mathbf{D}\mathbf{A}\right\|_F^2+\lambda\ell_o^{tree}\left(\mathbf{A}\right)\right) = \min_{\mathbf{D}\in C,\mathbf{A}\in\mathbb{R}^{K\times N}} \frac{1}{N}\sum_{i=1}^{N}\left(0.5\left\|\mathbf{x}^{(i)}-\mathbf{D}\mathbf{a}^{(i)}\right\|_2^2+\lambda\ell_o^{tree}\left(\mathbf{a}^{(i)}\right)\right) \tag{15}$$

in which C is the convex set of matrices $C \triangleq \mathbf{D} \in \mathbb{R}^{N\times K}$ s.t. $\forall_{j=1:K}, (\mathbf{d}^{(j)})^T\mathbf{d}^{(j)} \leq 1$.

C. EXPERIMENTAL PROTOCOL – IMPLEMENTATION DETAILS

This section deals with the implementation of the aforementioned sparse coding methods for the construction of hourly net load prediction models. More specifically, we analyze the choice of each methodology and provide all the relevant technical details.

Starting this task, we first encounter the algorithm K-SVD. As previously described in detail, K-SVD is an iterative method that alternates between sparse coding of the examples based on the current dictionary, and a process of updating the dictionary atoms to better fit the data. Since its introduction, it has been employed successfully in several image processing-oriented applications, such as restoration, denoising and face recognition. Therefore, being a well-known and tested algorithm, it was chosen for electricity load signal forecasting. With respect to the overcomplete dictionary property, the number of atoms is set to 60, six times greater than the dimension of input space. Another term to be defined during dictionary learning is the number of non-zero elements contained in the atoms. As a rule of thumb, this value usually does not exceed 10% of the number of atoms; the so-called sparsity level is selected to be equal to 6, following the aforementioned rule. As the K-SVD algorithm is flexible with regards to its compatibility with the sparse decomposition technique, OMP is used here. Finally, the update of the dictionary is performed with a maximum number of 100 iterations.

The online dictionary learning algorithm described in [38], is then applied in combination with LARS-LASSO for decomposition purposes. In this strategy, a sparsity penalty based on ℓ_1 norm, is imposed as shown in Eq. (3). The regularization parameter λ was set to 0.1. For completeness purposes we also provide results with the elastic-net [51], which combines the ℓ_1 and the ℓ_2 norms. For this case, the regularization coefficients λ_1 and λ_2 were set to 0.1 and 0.01, respectively. Finally, an alternative model is developed, featuring dictionary learning with the online algorithm and OMP for sparse coding [52] instead of LASSO. As previously, the value of λ was set to 0.1, following a trial-and-error procedure. For comparison reasons, the dictionary size remains the same for all experiments that take place throughout this study. Another parameter that affects the performance of the method is the execution time of the algorithm, which was set to 20 seconds, as it was observed that there is no improvement beyond this limit.

The next learning method that was implemented in the context of the present study employed the archetypal analysis. In this case, the dictionary identification consists of the non-convex optimization problem in (10) and (11), which is solved by the active set algorithm proposed in [49]. As it can be observed and differently to the approaches that were mentioned so far, sparsity is induced by the ℓ_2 norm by placing atoms at the vicinity of the convex hull of the ambient data space. Therefore, in addition to highlighting the most appropriate sparse coding methodology for our problem in hand, the effect of the various regularization norms is also being considered.

Towards this direction, an additional concept, that of hierarchical sparsity, is incorporated into the net load prediction model. This is done by introducing the idea of embedding the atoms of a dictionary in a directed tree in both dictionary learning and sparse representation stages. The conducted experiments with the use of hierarchical sparsity for STLF employ a ℓ_0^{tree} tree-structured sparse regularization norm, which has been found to be useful in a number of cases. The number of atoms depends on the tree configuration. Here, we select the branching structure of $\left[\begin{array}{ccc}4 & 4 & 2\end{array}\right]$, that conceptualizes a shallow, three-level depth tree, where the first two levels have four branches and the last one has two. This results to a 53-dimension dictionary, so that there is an agreement with the rest of the methods, as well as a reasonable execution time. As stated above, the sparsity level of the dictionary is regulated by the parameter λ. After trial and error, λ was set to 0.01 as the minimum value which produces valid dictionary elements, in the sense of crisp atoms.

In order to highlight the superiority of the proposed sparse coding methodology, standard linear or non-linear machine learning methods, including simple linear regression, support vector regression and MLP neural networks, are also tested. To ensure a fair comparison between the methods, all the comparative models use exactly the same input variables and training dataset with the proposed one. Furthermore, the hyper-parameters of each model are optimized using trial and error, in conjunction with suggestions in literature, as described below.

Probably the most common and simple modelling method is linear regression [53]. Thus, a least squares regression model was built and used as baseline; it should be noted that in this case, there are no hyper-parameters to be optimized. Subsequently, an SVR model with gaussian kernel function was employed [54]. Sequential minimal optimization (SMO) was selected as training algorithm, while Bayesian optimization was used to optimize the model's hyperparameters, namely epsilon, kernel scale and box constraint [55]. The most representative class of non-linear load forecasting methods is MLP neural networks [56]; to provide a comparison with similar approaches, a two-layered MLP network was developed and

trained with the Levenberg-Marquardt backpropagation algorithm [57]. The structure of the MLP networks constitutes an important set of hyper-parameters, which in this case was selected following an exhaustive search procedure; to be more specific a two-hidden layer structure was selected and all possible combinations of neurons between 5 and 20 in each hidden layer were tested. An architecture with 15 and 5 neurons in each layer was finally selected. It should be noted that, due to the fact that MLP training methods are very sensitive to initialization, the training procedure was repeated 20 times, each one using a different random initialization for the NN weights.

IV. RESULTS

The main goal of this work is to investigate the use of sparse representation techniques for net load forecasting with 1 hour prediction horizon. In the context of the case study, several experiments were conducted, implementing different methodologies. The simulation results are presented in a comparative form in this section.

A very important point in evaluating a forecasting model is first of all the accuracy it achieves in estimating the actual measurements. In order to be able to compare and evaluate the experimental results, it was considered appropriate to use a summary table. Table 3 contains the values of the performance indices for the proposed approach, alongside with the rest sparse-based methods. For reasons of better readability, each method of the table is accompanied by the norm it incorporates. As the hierarchical model proves to be the most efficient among all sparce coding approaches, its performance is compared to that of known linear and non-linear methods in Table 4. As common and representative metrics, R^2, MAE and RMSE are employed to measure the models' efficiency. In the context of investigating the predictive accuracy of the proposed model, the performance indices were also calculated individually for daylight and nighttime scenarios, and included in Table 5.

The prediction performance of the hierarchical structured model is presented graphically in Figs. 5a, 5b, and 6. The data shown in all the following figures are normalized in the range $[-1, 1]$ for confidentiality reasons. A snapshot of the hourly actual and predicted values spanning a randomly chosen interval of 24 hours, is depicted in Fig. 5a; it should be reminded that the predictions for the next hour are provided every 15 minutes. As a complementary graph, the respective forecasting errors are presented in Fig. 5b as the residuals between normalized actual and forecasting values. Finally, the actual AP values against predictions are presented using a scatterplot (Fig. 6).

V. DISCUSSION

Following are the remarks and conclusions drawn from the experimental results. At first, the reliability of the models is examined through their predicting performance, which is determined with R^2, MAE and RMSE. As can be seen from Table 3 the hierarchical sparsity-based model seems to outperform the other sparse representation techniques, establishing its superiority. Attempting to contrast the sparse methodologies with each other, we observe that archetypal analysis is not a competent approach in comparison with the proposed one. This is most likely due to the convexity constraints under which the dictionary and coefficients are extracted, making it impossible to adequately represent the signal, thus leading to inaccurate predictions. Beyond that, the underlying l_2 norm may not be conducive to creating an efficient forecasting model. However, a significant improvement in performance is noted with the use of K-SVD/OMP, Online-OMP/LARS-LASSO and Online/LARS-LASSO algorithms, which incorporate ℓ_0, ℓ_1 or $\ell_1 + \ell_2$ norm induced sparsity. In this case, finding the dictionary atoms is not subject to any convexity or non-negativity constraints, which probably make these models superior to archetype analysis. In addition, the small difference of 6% for the MAE, in favor of the ONLINE/LARS-LASSO, leads us to the conclusion that the ℓ_1 norm is more appropriate for the problem in hand.

One point worth commenting on, concerns the method by which the atoms are obtained, in each case. Despite the fairly large difference in their performance, the ℓ_1 and ℓ_0^{tree} norm sparse models share the same dictionary learning algorithm, that of online matrix factorization. This remark leads us to the conclusion that the superiority of the hierarchical model stems from its operating principle which imposes the selection of non-zero coefficients that necessarily belong to a subset of the original tree. The subordinate predictive behavior of the models incorporating K-SVD/OMP and archetypes

TABLE 3. Performance of sparse methods for 24 hours scenario.

Sparse Method	Norm	R^2	MAE	RMSE
K-SVD/OMP	ℓ_0	0.8867	0.8048	1.1641
Online / LARS-LASSO	ℓ_1	0.8817	0.7396	1.0111
Elastic Net	$\ell_1 + \ell_2$	0.8965	0.8570	1.1869
Online-OMP / LARS-LASSO	ℓ_0	0.9191	0.8993	1.2248
Archetypal Analysis	ℓ_2	0.7896	1.0236	1.6655
Hierarchical Sparsity	ℓ_0^{tree}	**0.9406**	**0.6058**	**0.9426**

TABLE 4. Performance of proposed and comparison methods for 24 hours scenario.

Method	R^2	MAE	RMSE
LR	0.9180	0.7754	1.1074
SVR	0.9255	0.7332	1.0557
*MLP**	0.9163 (0.8868 ± 0.1085)	0.7929 (0.8146 ± 0.1620)	1.2002 (1.2261 ± 0.1534)
Hierarchical Sparsity	**0.9406**	**0.6058**	**0.9426**

*For the MLP approach, the best network performance is shown along with the mean and standard deviation of the 20 runs within parentheses.

TABLE 5. Performance of proposed and comparison methods for daylight and nighttime scenario.

Method	Daylight			Nighttime		
	R^2	MAE	RMSE	R^2	MAE	RMSE
LR	0.8912	0.9715	1.3190	0.9395	0.5798	0.8454
SVR	0.8978	0.9366	1.2783	0.9495	0.5304	0.7723
MLP	0.8910 (0.8603 ± 0.1332)	0.9918 (1.0215 ± 0.1971)	1.4108 (1.4325 ± 0.1677)	0.9399 (0.9085 ± 0.0745)	0.5935 (0.6069 ± 0.1261)	1.0751 (1.1248 ± 0.1353)
Hierarchical Sparsity	**0.9025**	**0.8758**	**1.2488**	**0.9814**	**0.3366**	**0.4685**

*For the MLP approach, the best network performance is shown along with the mean and standard deviation of the 20 runs within parentheses.

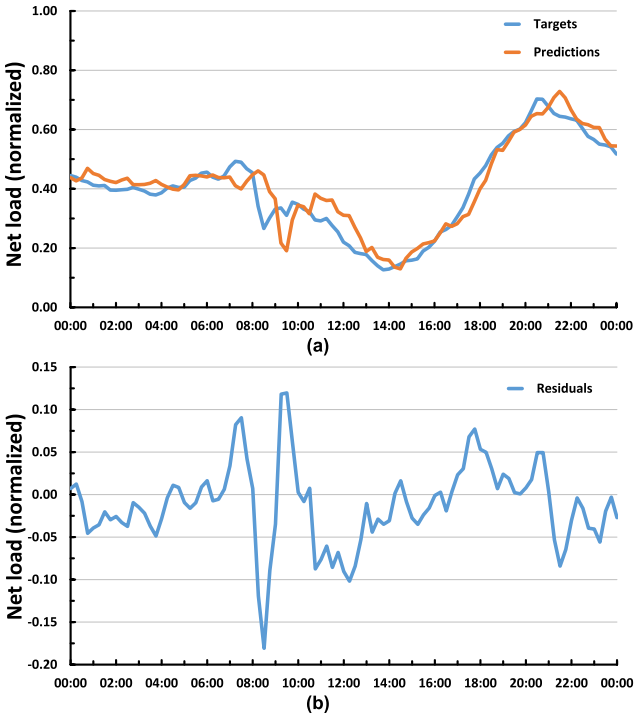

FIGURE 5. Hierarchical sparse model (a) real and predicted net load values, and (b) residuals between real and predicted net load values.

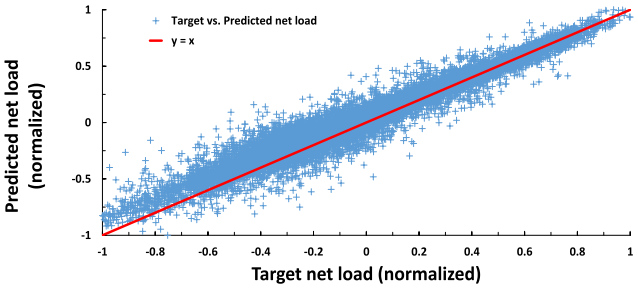

FIGURE 6. Scatterplot of actual net load values against predictions from the hierarchical sparse model.

can be interpreted similarly. An additional advantage of this methodology lies in the flexibility it provides in the selection of the atoms, during the dictionary learning process, through different branching configurations. As already has been stated, net load time series despite its volatility, is characterized by seasonal effects, that stem from calendar factors and consumers' profile. Taking advantage of the properties of group sparsity, we built a model that is inherently capable of extracting specific features of the net load with strong correlation, and thus providing high-accuracy predictions. It should be noted that hierarchical sparse coding has already been proven to be an effective approach for modeling time-series presenting structured patterns, according to the literature, where a sparsity-controlled vector autoregressive model is established in [58] and tested upon several datasets. Finally, group sparsity has also been found beneficial in different types of time-series, e.g. microgrid reconfiguration [59] and representative selection in choreographic time-series [60].

The results so far can be visualized through Fig. 5a, where the high predictive ability of the hierarchical model is confirmed, especially during the evening hours, according to Table 5. It is easy to see that the predictions follow with quite high accuracy the time series of the actual net load. These results are perfectly reasonable if we take into account the high penetration of photovoltaics that characterizes the data of the substation. Obviously, their operation during daylight, makes it difficult to make accurate predictions. Additionally, Fig. 5b shows that the forecast error for the given 24-hour period ranges from -0.15 to 0.135, which is in line with the value of the total MAE concerning forecasts for a period of one year. In order to further assess the reliability of the hierarchical model, the scatterplot of Fig. 6 is illustrated. Based on this graph we conclude that the methodology generally manages to provide valid forecasts. However, as indicated by the reference line, the predictions are less accurate for small values of the actual net load.

Moreover, the proposed method prevails not only against sparse competing models, but also against some established forecasting methodologies. Looking at Table 4, it is obvious that the tree-structured approach has a positive effect on the net load forecasts, as the achieved MAE and RMSE are 17% and 6% lower, respectively, while the R^2 coefficient is 2.3% higher than that of LR. The SVR model appears similarly inferior, but with a slight improvement in relation with LR. This is justified, as we expect higher performance from non-linear methodologies, taking into account the non-linear behavior of the electric load. However, this

is not the case with neural networks. The weak performance of MLP, compared to LR and SVR, evidenced by the metric values is probably due to overfitting. In conclusion, the hierarchical model achieves a remarkable supremacy against both linear and non-linear methods. This outcome is particularly important for the evaluation of the proposed method and enhances its predictive reliability. As stated above, the tree-based dictionary learning constitutes a strong and robust training tool, leading to a high-performance model, avoiding the risk of data noise sensitivity which characterizes the non-linear methods.

VI. CONCLUSION

The prediction of the net load with a time horizon of one hour is studied in this article, presenting a new method based on sparse representation. The proposed model adopts a hierarchical tree structure approach in order to derive the coefficients, while the sparsity is induced by ℓ_0^{tree} norm. The underlying load forecasting problem is also confronted by a number of sparse-based techniques. Through the experimental simulations that are performed, these methodologies are compared with each other and their strengths and weaknesses are highlighted. Finally, the obtained results testify to the superior performance of the hierarchical sparsity model not only in relation to the other sparse approaches but also to common and established methods, such as linear regression, support vector regression and MLP neural networks.

Starting from the superiority of the hierarchical sparsity model for net load forecasting, a proposed future research direction would be to further exploit the way the method works, such as by assigning the prediction of different load patterns to different branches. Given the promising results obtained in the current work when applying STLF at the substation level using sparse coding for the first time, the development of similar prediction models is strongly recommended. An upcoming challenge of the current power grids, is the integration of energy storage systems, in order to meet the ever-growing energy needs. For this reason, the authors are convinced that such models, can be exploited for congestion management and fault detection, significantly enhancing the stability and security of future distribution grids. Also, taking into account the limited scope of application of sparse coding methods in the electricity sector, the extension of their use to additional areas, such as RES generation forecast, as well as long-term net load and reactive power forecasting, is considered to be a fruitful direction.

Another appealing pillar which can be further exploited has to do with the potential extension of sparse coding principals to deep learning methods. In order to proceed to this extent, we need to incorporate two relative parametric and generative models for managing signals: a) Convolutional Sparse Coding (CSC), and b) multi-layered Convolutional Sparse Coding ML-CSC [15]. Specifically, the ML-CSC addresses a number of common deep learning architectures which makes ML-SCS an interesting candidate for a new line of research for developing deep-learning models.

REFERENCES

[1] C. Kuster, Y. Rezgui, and M. Mourshed, "Electrical load forecasting models: A critical systematic review," *Sustain. Cities Soc.*, vol. 35, pp. 257–270, Nov. 2017.
[2] M. Papadimitrakis, N. Giamarelos, M. Stogiannos, E. N. Zois, N. A.-I. Livanos, and A. Alexandridis, "Metaheuristic search in smart grid: A review with emphasis on planning, scheduling and power flow optimization applications," *Renew. Sustain. Energy Rev.*, vol. 145, Jul. 2021, Art. no. 111072.
[3] I. K. Nti, M. Teimeh, O. Nyarko-Boateng, and A. F. Adekoya, "Electricity load forecasting: A systematic review," *J. Electr. Syst. Inf. Technol.*, vol. 7, no. 1, p. 19, Dec. 2020.
[4] L. Wu, M. Shahidehpour, and T. Li, "Stochastic security-constrained unit commitment," *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 800–811, May 2007.
[5] J. Xie, Y. Chen, T. Hong, and T. D. Laing, "Relative humidity for load forecasting models," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 191–198, Jan. 2018.
[6] Y. Yang, S. Li, W. Li, and M. Qu, "Power load probability density forecasting using Gaussian process quantile regression," *Appl. Energy*, vol. 213, pp. 499–509, Mar. 2018.
[7] E. Ceperic, V. Ceperic, and A. Baric, "A strategy for short-term load forecasting by support vector regression machines," *IEEE Trans. Power Syst.*, vol. 28, no. 4, pp. 4356–4364, Nov. 2013.
[8] Y. Li, J. Che, and Y. Yang, "Subsampled support vector regression ensemble for short term electric load forecasting," *Energy*, vol. 164, pp. 160–170, Dec. 2018.
[9] A. K. Srivastava, A. S. Pandey, and D. Singh, "Short-term load forecasting methods: A review," in *Proc. Int. Conf. Emerg. Trends Elect., Electron. Sustain. Energy Syst. (ICETEESES)*, Sultanpur, India, 2016.
[10] K. Zor, O. Timur, and A. Teke, "A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting," in *Proc. 6th Int. Youth Conf. Energy (IYCE)*, Jun. 2017, pp. 1–7.
[11] P. Zeng, M. Jin, and M. F. Elahe, "Short-term power load forecasting based on cross multi-model and second decision mechanism," *IEEE Access*, vol. 8, pp. 184061–184072, 2020.
[12] M. Elad, *Sparse and Redundant Representations*. New York, NY, USA: Springer, 2010.
[13] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.
[14] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.
[15] V. Papyan, Y. Romano, J. Sulam, and M. Elad, "Theoretical foundations of deep learning via sparse representations: A multilayer sparse model and its connection to convolutional neural networks," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 72–89, Jul. 2018.
[16] J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad, "Trainlets: Dictionary learning in high dimensions," *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 3180–3193, Jun. 2016.
[17] E. Elhamifar, G. Sapiro, and S. S. Sastry, "Dissimilarity-based sparse subset selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 11, pp. 2182–2197, Nov. 2016.
[18] C.-N. Yu, P. Mirowski, and T. K. Ho, "A sparse coding approach to household electricity demand forecasting in smart grids," *IEEE Trans. Smart Grid*, vol. 8, no. 2, pp. 738–748, Mar. 2017.
[19] D. Yang, L. Xu, S. Gong, H. Li, G. D. Peterson, and Z. Zhang, "Joint electrical load modeling and forecasting based on sparse Bayesian learning for the smart grid," in *Proc. 45th Annu. Conf. Inf. Sci. Syst.*, 2011, pp. 1–6.
[20] X. Sun, X. Wang, J. Wu, and Y. Liu, "Hierarchical sparse learning for load forecasting in cyber-physical energy systems," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf.*, May 2013, pp. 533–538.
[21] Q. Duan, W. X. Sheng, Y. Ma, and K. Ma, "Sparse Bayesian learning using combined kernels for medium term load forecasting," in *Proc. 2nd IET Renew. Power Gener. Conf. (RPG)*, Beijing, China, 2013.
[22] Y. Zhang, S. J. Kim, and G. B. Giannakis, "Short-term wind power forecasting using nonnegative sparse coding," in *Proc. 49th Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, 2015.
[23] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[24] S. Singh and A. Majumdar, "Deep sparse coding for non–intrusive load monitoring," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4669–4678, Sep. 2018.

[25] Y. Wang, Q. Chen, C. Kang, Q. Xia, and M. Luo, "Sparse and redundant representation-based smart meter data compression and pattern extraction," *IEEE Trans. Power Syst.*, vol. 32, no. 3, pp. 2142–2151, May 2017.
[26] H. Chen, S. Wang, S. Wang, and Y. Li, "Day-ahead aggregated load forecasting based on two-terminal sparse coding and deep neural network fusion," *Electr. Power Syst. Res.*, vol. 177, Dec. 2019, Art. no. 105987.
[27] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, Jun. 1996.
[28] D. L. Donoho, "Neighborly polytopes and sparse solution of underdetermined linear equations," Stanford Univ., Stanford, CA, USA, Tech. Rep. 2005-04, 2004.
[29] N. J. Johannesen, M. Kolhe, and M. Goodwin, "Relative evaluation of regression tools for urban area electrical energy demand forecasting," *J. Cleaner Prod.*, vol. 218, pp. 555–564, May 2019.
[30] S. R. Salkuti, "Day-ahead thermal and renewable power generation scheduling considering uncertainty," *Renew. Energy*, vol. 131, pp. 956–965, Feb. 2019.
[31] N. Zhang, C. Kang, Q. Xia, Y. Ding, Y. Huang, R. Sun, J. Huang, and J. Bai, "A convex model of risk-based unit commitment for day-ahead market clearing considering wind power uncertainty," *IEEE Trans. Power Syst.*, vol. 30, no. 3, pp. 1582–1592, May 2015.
[32] D. Csercsik, A. Sleisz, and P. M. Sores, "Increasing the flexibility of European type electricity auctions via a novel bid class," in *Proc. 16th Int. Conf. Eur. Energy Market (EEM)*, Sep. 2019, pp. 1–4.
[33] L. Sun, K. Zhou, X. Zhang, and S. Yang, "Outlier data treatment methods toward smart grid applications," *IEEE Access*, vol. 6, pp. 39849–39859, 2018.
[34] M. Q. Raza, Z. Baharudin, B.-U.-I. Badar-Ul-Islam, M. A. Zakariya, and M. H. M. Khir, "Neural network based STLF model to study the seasonal impact of weather and exogenous variables," *Res. J. Appl. Sci., Eng. Technol.*, vol. 6, no. 20, pp. 3729–3735, Nov. 2013.
[35] M. H. Amini, A. Kargarian, and O. Karabasoglu, "ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation," *Electr. Power Syst. Res.*, vol. 140, pp. 378–390, Nov. 2016.
[36] K. B. Sahay and M. M. Tripathi, "Day ahead hourly load forecast of PJM electricity market and iso new england market by using artificial neural network," in *Proc. ISGT*, Feb. 2014, pp. 1–5.
[37] D. H. Vu, K. M. Muttaqi, and A. P. Agalgaonkar, "Assessing the influence of climatic variables on electricity demand," in *Proc. IEEE PES Gen. Meeting | Conf. Expo.*, Jul. 2014, pp. 1–5.
[38] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, no. 1, pp. 19–60, Jan. 2010.
[39] R. Jenatton, I. J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *Proc. 27th Int. Conf. Mach. Learn.*, Haifa, Israel, 2010.
[40] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Supervised dictionary learning," INRIA, Paris, France, Tech. Rep. 6652-2008, 2009.
[41] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.
[42] E. N. Zois, D. Tsourounis, I. Theodorakopoulos, A. L. Kesidis, and G. Economou, "A comprehensive study of sparse representation techniques for offline signature verification," *IEEE Trans. Biometrics, Behav., Identity Sci.*, vol. 1, no. 1, pp. 68–81, Jan. 2019.
[43] F. Bach, "Optimization with sparsity-inducing penalties," *Found. Trends Mach. Learn.*, vol. 4, no. 1, pp. 1–106, Aug. 2011.
[44] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.
[45] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," Israel Inst. Technol., Haifa, Israel, Tech. Rep. CS-2008-08, 2008.
[46] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc. B, Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.
[47] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, Apr. 2004.
[48] A. Cutler and L. Breiman, "Archetypal analysis," *Technometrics*, vol. 36, no. 4, p. 338, Nov. 1994.
[49] Y. Chen, J. Mairal, and Z. Harchaoui, "Fast and robust archetypal analysis for representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, May 2014, pp. 1478–1485.
[50] R. Jenatton, I. J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *J. Mach. Learn. Res.*, vol. 12, pp. 2297–2334, 2011.
[51] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Stat. Soc. B*, vol. 67, no. 2, pp. 301–320, 2005.
[52] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.
[53] B. Dhaval and A. Deshpande, "Short-term load forecasting with using multiple linear regression," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 4, pp. 3911–3917, 2020.
[54] Z. Tan, J. Zhang, Y. He, Y. Zhang, G. Xiong, and Y. Liu, "Short-term load forecasting based on integration of SVR and stacking," *IEEE Access*, vol. 8, pp. 227719–227728, 2020.
[55] M. S. Alam, N. Sultana, and S. M. Z. Hossain, "Bayesian optimization algorithm based support vector regression analysis for estimation of shear capacity of FRP reinforced concrete members," *Appl. Soft Comput.*, vol. 105, Jul. 2021, Art. no. 107281.
[56] G. Dudek, "Multilayer perceptron for short-term load forecasting: From global to local approach," *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3695–3707, Apr. 2020.
[57] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 989–993, Nov. 1994.
[58] E. Carrizosa, A. V. Olivares-Nadal, and P. Ramírez-Cobo, "A sparsity-controlled vector autoregressive model," *Biostatistics*, vol. 18, no. 2, pp. 244–259, Apr. 2017.
[59] E. Dall Anese and G. B. Giannakis, "Risk-constrained microgrid reconfiguration using group sparsity," *IEEE Trans. Sustain. Energy*, vol. 5, no. 4, pp. 1415–1425, Oct. 2014.
[60] I. Rallis, N. Doulamis, A. Voulodimos, and A. Doulamis, "Hierarchical sparse modeling for representative selection in choreographic time series," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 1023–1027.

NIKOLAOS GIAMARELOS received the Diploma degree in applied mathematics and physics and the master's degree in automation systems from the National Technical University of Athens, Greece, in 2016 and 2018, respectively. He is currently pursuing the Ph.D. degree in the field of computational intelligence with the University of West Attica, Athens.

His current research interests include modeling and optimization of dynamic systems, machine learning, and smart grid engineering applications.

ELIAS N. ZOIS received the bachelor's degree in physics, the M.Sc. degree in electronic engineering, and the Ph.D. degree from the University of Patras (UP), Patras, Greece, in 1994, 1996, and 2000, respectively.

From 2000 to 2008, he was working as an Adjunct Professor with the Technological Educational Institute of Athens. He is currently an Assistant Professor with the University of West Attica. His research interests include among other, computer vision, image processing, machine learning, and biometrics.

MYRON PAPADIMITRAKIS (Student Member, IEEE) was born in Athens, Greece, in 1994. He received the Diploma degree in mechanical engineering from the University of Thessaly (UTH), in 2018. He is currently pursuing the Ph.D. degree in the domain of control systems and machine learning with the Department of Electrical and Electronics Engineering, University of West Attica.

His research interests include nonlinear system modeling and control, with emphasis on smart grid applications. He is a member of the Technical Chamber of Greece.

MARIOS STOGIANNOS received the B.Sc. degree in electronic engineering and the M.Sc. degree in the field of design and development of advanced electronic systems from the Technological Educational Institute of Athens (TEIA), Greece, in 2013 and 2016, respectively. He is currently pursuing the Ph.D. degree in the field of process analysis and plant design with the National Technical University of Athens (NTUA), Greece.

His research interests include computational intelligence, metaheuristic search, system modeling, and intelligent control.

NIKOLAOS-ANTONIOS I. LIVANOS was born in Athens, Greece, in 1973. He received the B.Sc. and M.Sc. degrees in physics and electronics from the University of Patras, in 1995 and 1997, respectively. He is currently pursuing the Ph.D. degree with the University of Bolton, U.K.

He is also the Founder of EMTECH Group (three micro companies in Greece, Germany, and Cyprus). He is the Managing Director and the Technical Manager at EMTECH SPACE P.C., Greece. He is a member of the Experts Team at Skolkovo Foundation, Russia, for the space, energy, and medical clusters. His scientific interests include real-time multidisciplinary decision support systems based on co-simulations, in the domains of space simulation, energy automation, and health-care systems.

ALEX ALEXANDRIDIS (Member, IEEE) received the Diploma degree in chemical engineering and the Ph.D. degree in computational intelligence and control from the National Technical University of Athens (NTUA), Greece, in 2000 and 2003, respectively.

Since 2018, he has been a Full Professor with the Department of Electrical and Electronic Engineering, University of West Attica, Greece. He serves as the principal investigator for a number of research projects. He has authored or coauthored more than 80 original research works. His research interests include computational intelligence, optimization, nonlinear system modeling and control, with emphasis on model predictive control methods, and applications to energy systems, process engineering, materials science, geoscience, and environmental science.