CHAPTER 9

Time-Series Classification Methods: Review and Applications to Power Systems Data

Gian Antonio Susto, Angelo Cenedese, Matteo Terzi
University of Padova, Padova, Italy

CHAPTER OVERVIEW

The diffusion in power systems of distributed renewable energy resources, electric vehicles, and controllable loads has made advanced monitoring systems fundamental to cope with the consequent disturbances in power flows; advanced monitoring systems can be employed for anomaly detection, root cause analysis, and control purposes. Several machine learning-based approaches have been developed in the past recent years to detect if a power system is running under anomalous conditions and, eventually, to classify such situation with respect to known problems. One of the aspects, which makes Power Systems challenging to be tackled, is that the monitoring has to be performed on streams of data that have a time-series evolution; this issue is generally tackled by performing a features' extraction procedure before the classification phase. The features' extraction phase consists of translating the informative content of time-series data into scalar quantities: such procedure may be a time-consuming step that requires the involvement of process experts to avoid loss of information in the making; moreover, extracted features designed to capture certain behaviors of the system, may not be informative under unseen conditions leading to poor monitoring performances. A different type of data-driven approaches, which will be reviewed in this chapter, allows to perform classification directly on the raw time-series data, avoiding the features' extraction phase: among these approaches, dynamic time warping and symbolic-based methodologies have been widely applied in many application areas. In the following, pros and cons of each approach will be discussed and practical implementation guidelines will be provided.

1 INTRODUCTION

The modern trends in energy generation, transmission, and distribution follow the paradigm of *smart infrastructures* to gain in service flexibility, reliability, and autonomy while not compromising the overall system performance and control. Related for example to the decentralized and/or distributed exploitation of renewable energy resources, the employment of electric vehicle fleets, the management of controllable loads, and these policies have made advanced monitoring (AM) systems fundamental to assess line conditions and utilities' behaviors, in order to grant the requested quality of service to the final user, cope with the presence of disturbances in power flows, and push current generation of power systems toward their limit.

Big Data Application in Power Systems. https://doi.org/10.1016/B978-0-12-811968-6.00009-7

Pervasive measurement of such a complex and interconnected system of systems can provide (and do provide) a huge amount of data to be employed for a variety of purposes ranging from failure and anomaly detection [3] to the demand/response analysis and optimization [4–6], from the root cause analysis [7] to the service provider control [8], and from the predictive/preventive maintenance [9] to the physical or cyber-attack prevention [10]. In particular, the development of phasor measurement units (PMUs) [11], frequency disturbance recorder [12], and advanced metering infrastructure (AMI) [13] have allowed the continuous monitoring of the transmission line and the connected power systems, and can be complemented with utility monitoring devices, smart meters, insulation monitoring units, to build a thorough picture of the whole grid structure, health, and dynamic behavior.

Nonetheless, to unleash the full value of these complex data sets, algorithms need to be developed to transform these massive dumb data flows into synoptic smart information and drive the way to manage the energy and power systems [14]. Indeed, these solutions typically constitute the core of energy management systems (EMS), which can be specifically translated toward the final application in factories (FEMS), buildings (BEMS), and home (HEMS) [15]. In this sense, several machine learning (ML)-based approaches [16, 17] have been developed in the past recent years to *characterize* the behavior of smart grid systems and power lines, to *profile* user demand and exploitation of resources and services, to *detect* if a power system is running under anomalous conditions, to *classify* such situations with respect to known problems.

1.1 Contribution

In this chapter, we will try to provide an overview of the main ML techniques that are used in the context of power systems. Without aiming at being exhaustive, the main goal of this contribution is to highlight the differences among the approaches in terms of information they can provide and issues in their usage. In particular, we will use the term *classification* to indicate the subfield of ML in the realm of *supervised learning* where "supervised" indicates that the output is known: given a signal $\mathbf{x}$ belonging to some domain $\mathbb{X}$ as an input and a finite set $\mathbb{Y}$ of different classes (the output), the problem of supervised learning consists on finding a rule that associates $\mathbf{x}$ to one $\mathbf{y} \in \mathbb{Y}$. In general, the set of output classes (also called dictionary) is obtained according to a training procedure where a training input dataset is used to characterize both $\mathbb{Y}$ and the learning rule.

In the context of power systems, we are facing classification problems when dealing with fault detection and isolation (FDI), predictive maintenance, AM, user profiling, and cyber-security applications. Just to provide an example

in the case of FDI, one output class can be related to the normal behavior of the power system, while additional classes can be referred to known problems like voltage sags, voltage swells, fault currents, voltage oscillations, and frequency oscillations, while the input signals are time-series generated from multiple continuous data flows such as PMUs data, currents, or voltages: the task of an FDI algorithm is to interpret heterogeneous signals coming from different measurement units in order to discern and understand the state of the overall system [2].

1.2 Notation

Throughout this chapter, we consider a time-series $\mathbf{z}_\bullet$ as a (finite-length) sequence of n ordered real values at time instants $t_{\bullet,1}$, …, $t_{\bullet,n}$. For the sake of simplicity, and without loss of generality, we assume that the time series is obtained through a preprocessing phase that may include sampling and windowing of the continuous data flow coming from a measurement unit. The time series is then characterized by p input descriptors x (whose meaning will be clearer in the following), hence the input space $\mathbb{X}$ is p-dimensional, and a training set composed by N signals $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$ allows to define the class set $\mathbb{Y}$. These basic definitions and notation are shown in Fig. 1 and summarized in Table 1.

In Fig. 2 the data flow from sensors (PMUs or other types) to classification is represented with the notations adopted in this chapter to indicate all the related quantities. For the sake of clarity, Table 2 provides the list of acronyms adopted throughout this chapter.

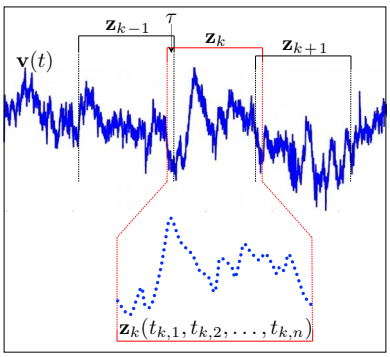

FIG. 1

Windowing procedure. Finite-length windows $\mathbf{z}_k$ are extracted from the raw data stream v to obtain time series.

Table 1 Summary Table of the Main Adopted Notation

Symbol	Description
$t \in \mathbb{R}$	Time
$\overline{t}$	Time of interest
R	Number of nodes in the power system cluster
$S \geq R$	Number of data sources generating signals
$\overline{q} \geq R$	Cardinality of raw signal
$\mathbf{v}(t) \in \mathbb{R}^{\overline{q}} \times \mathbb{R}$	Raw signal
$k \in \mathbb{N}$	Window index
$\tau \in [0, 1)$	Window overlap parameter
n	Cardinality of samples per window
$q \leq \overline{q}$	Cardinality of preprocessed signal
$\mathbf{z}_k(t_{k,1}, \ldots, t_{k,n}) \in \mathbb{R}^{n \times q}$	Preprocessed signal
$p \leq n$	Number of signal descriptors
$\mathbb{X} \subseteq \mathbb{R}^p$	Domain of signal descriptors
$\mathbf{x} = x_1, \ldots, x_p \in \mathbb{X}$	Signal descriptors
N	Number of observations available for training
$\{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$	Set of input data for training
$\{y^1, \ldots, y^N\}$	Set of output labels for training
$D \in \mathbb{R}^{N \times (p+1)}$	Design matrix for training
$\widetilde{M}$	Number of observations of reduced dataset (e.g., dictionary learning)
M	Number of classes
$\mathbb{Y}$	Class dictionary
$y \in \mathbb{Y}$	Class label
$f(\bullet) : \mathbb{X} \rightarrow \mathbb{Y}$	Classifier/association rule

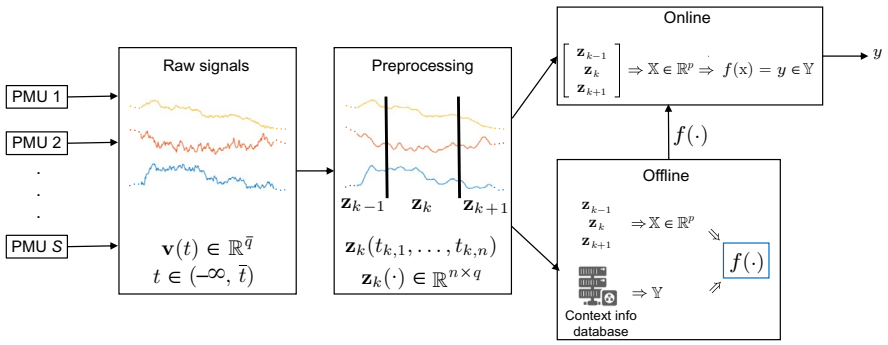

FIG. 2

Data flow. Scheme of the data flow from sensors (PMUs or other types) to classification.

Table 2 Summary Table of the Main Acronyms Used in the Text

Acronyms	Descriptions
1-NN	1 Nearest neighbor
1-NN-DTW	1-NN with DTW distance
AR	Auto regressive
ARMA	Auto regressive moving average
ARIMA	Auto regressive integrated moving average
BEMS	Building energy management system
BoF	Bag of features
BoSS	Bag-of-SFA symbols
BoSS-VS	BoSS-vector space
BoW	Bag of words
DB	Distance based
DBA	DTW Barycenter averaging
DFT	Discrete Fourier transform
DR	Dimensionality reduction
DDTW	Derivative dynamic time warping
DT	Decision tree
DTW	Dynamic time warping
DTWUDC	DTW under dynamic constraints
DWT	Discrete wavelet transform
DNN	Deep neural network
EBC	Ensemble of bundle classifier
EMS	Energy management system
ERP	Edit distance with real penalty
ESN	Echo state network
FB	Feature based
FDI	Fault detection and isolation
FEMS	Factory energy management system
GP	Gaussian process
HEMS	Home energy management system
HMM	Hidden Markov model
ICA	Independent component analysis
IF	Interval feature
k-NN	*k*-Nearest neighbors
LDA	Linear discriminant analysis
LR	Logistic regression
LSM	Liquid state machine
MCB	Multiple coefficient binning
MDS	Multidimensional scaling
ML	Machine learning
MMCL	Model metric colearning

Continued

Table 2 Summary Table of the Main Acronyms Used in the Text *Continued*

Acronyms	Descriptions
mRmR	Minimum redundancy maximum relevance
NR	Numerosity reduction
PAA	Piecewise aggregate approximation
PCA	Principal component analysis
PDC	Phasor data concentrator
PMU	Phasor measurement unit
RF	Random forest
RVM	Relevance vector machine
SAX	Symbolic aggregate approximation
SFA	Symbolic Fourier approximation
SIFT	Scale invariant feature transform
SMTS	Symbolic multivariate time series
SVM	Support vector machine
TWED	Time warping with edit distance
VAR	Vector autoregressive model
VSM	Vector space model
WDTW	Weighted dynamic time warping

2 THE CLASSIFICATION PROBLEM

The research on classification of time series has been of certain interest for some decades and in various fields, from speech recognition [18] to financial analysis [19], from manufacturing [20] to, of course, power systems [2, 21–23], and it is even more of key importance in this era of big data and pervasive information flow. Specifically, two cornerstone issues need to be addressed:

- How do we compare different time series? In particular, how do we compare time series with different lengths?
- How can we recognize that different time series are *realizations* of a common (unknown) process which represents a certain class?

The last question is particularly relevant in AM applications: if a database of known failures is available, detection of current failures could be performed and exploited in predictive maintenance [24]/fault detection (FD) and FDI solutions. Some works in AM of power systems formalize FD and FDI problems as semisupervised ones, where particular classifiers (like One-Class-SVM) are built on a single group of data: such data are usually associated with normality conditions [25]; the goal of this classifiers is to create a solution that define a "normality space": when a new observation is available, it will be classified as anomaly if it lies outside the boundaries of the normality space. Such problem

formulation can also be tackled with some of the methodologies presented in this chapter.

2.1 Classification Methods Taxonomy

For the sake of clarity, we provide a brief introduction to the different methodologies treated in this work that can be employed to solve the classification task with power system data. Time-series classification techniques can be essentially divided into two main branches:

- *Feature-based (FB)*. FB methods perform a *feature extraction* procedure before the classification phase. In general, from the original signal $\mathbf{v}(t)$ a moving window k of fixed length n is considered to obtain a time-series $\mathbf{z}_k$ and a set $\mathbf{x}$ of p *features* is calculated over it: to give some examples, commonly chosen features are mean, variance, maximum, minimum, entropy, all related to the time series extracted from the signal.
  The idea underlying these methods is to capture signal statistics that identify a certain class of signals. In theory, if a process is weakly stationary then a second-order statistic is sufficient to characterize that signal; however, signals obtained from real-world scenarios are not stationary due to several nuisance factors and many more features may be necessary to summarize the informative content.
  In this respect, some observations are in order: unfortunately, nonautomatic feature extraction procedures may be a time-consuming step that requires the involvement of process experts to avoid loss of information; moreover, extracted features designed to capture certain behaviors of the system, may not be informative under unseen conditions leading to poor monitoring performances. Finally, the tuning of n is far to be trivial for optimal results: normally, it is estimated through a cross-validation procedure [26]. When dealing with the learning phase in FB methods, the learning rule is based on the definition of a dataset of N observations and of a *design matrix* as

$$D = \begin{bmatrix} \mathbf{x}^{(1)} & y^{(1)} \\ \mathbf{x}^{(2)} & y^{(2)} \\ \vdots & \vdots \\ \mathbf{x}^{(N)} & y^{(N)} \end{bmatrix} \in \mathbb{R}^{N\times(p+1)}. \tag{1}$$

- *Distance-based (DB)*. DB methods avoid the feature extraction phase in favor of the definition of suitable *distances*, among which the most common is dynamic time warping (DTW) [27]. Then, the classification phase is carried out through metric classifiers: one simple and popular choice and, surprisingly, one of the most effective is 1-nearest neighbor classifier (1-NN) [26].

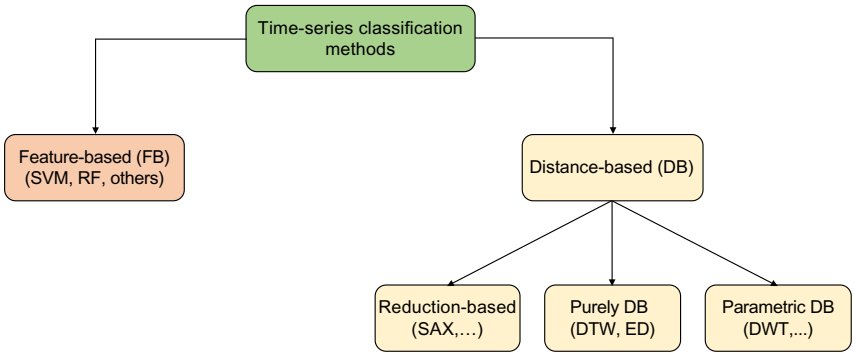

FIG. 3

Classification taxonomy. Time-series classification methodologies' tree highlighting the two families of feature based and distance based.

This strategy is motivated by the fact that the feature extraction phase could be time consuming and may cause significant loss of information about the original signal [28]. Conversely, though, due to nuisance factors, the DB direct comparison of time series (e.g., by exploiting the Euclidean distance) may lead to ill-posed problems and unsatisfactory performances, thus calling for a careful selection of the distance metrics that trades off between complexity (of the measure) and accuracy (in the classification).

This main categorization is also summarized in Fig. 3.

2.2 Computational Issues

The ML techniques to be employed in big data-related applications strongly depend on the architecture of the EMS infrastructure that delivers the task. In the context of power systems, the architecture can be represented as a main "parent" system unit that monitors and controls the connected "child" nodes. In turn, each node is a smaller unit that processes the measurements derived from multiple PMUs. A typical scheme of an EMS architecture is shown in Fig. 4.

Given this structure, the main unit is provided with powerful hardware in terms of computational and memory resources, while, conversely, the nodes are equipped with resource-limited hardware. The described network architecture must to be taken into account in the classification algorithms design and the algorithmic burden must be distributed over the system; the main system unit will be able to run demanding algorithms in a centralized fashion while nodes will constitute a computational grid with the parallel computation of parsimonious local procedures. In this respect, a further premise is needed to allow a better understanding of the remainder of this chapter. A learning algorithm

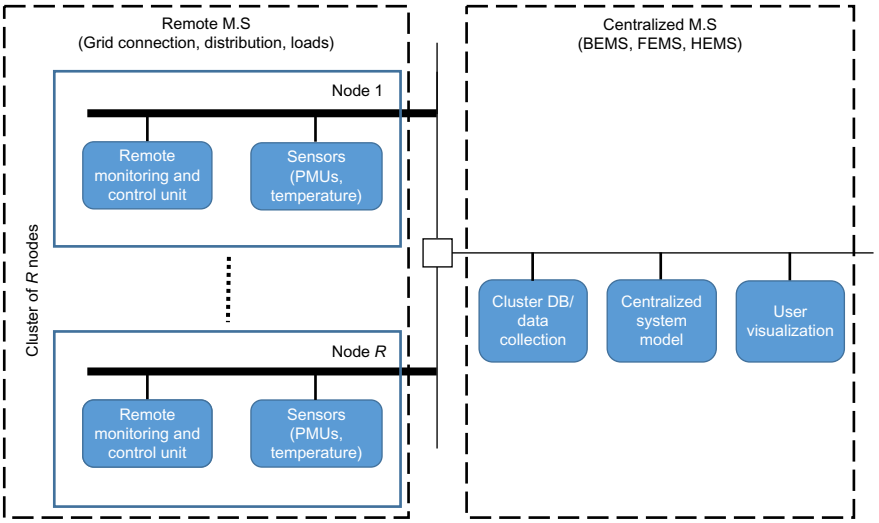

FIG. 4

Energy management system (EMS). Logical block diagram of EMS with the two main parts of remote monitoring and centralized control. These solutions are then specialized into HEMS, BEMS, FEMS, according to their employment within the residential, building, factory environments.

can be characterized in terms of complexity according to two different points of view, namely *space complexity* and *time complexity*: clearly, when dealing with resource-constrained systems, it is crucial to take into account both these aspects. For example, time-complexity is composed by two terms, respectively, related to training complexity and classification complexity: in most applications the training phase is run on systems with high computation and memory capabilities, while the classification complexity can be reduced so that the algorithmic solution can be implemented in the nodes.

There is also another point that forces nodes to be equipped with parsimonious algorithms: generally, in the nodes an *online* monitoring action is required in order to detect anomalies as soon as possible; in these settings, thus, classification must be executed almost in real time. Conversely, this does not necessarily apply for central units, where, generally, *offline* analyses are performed.

An important example of algorithms that are suited only for central units, but not for nodes, is *lazy-learning* approaches. Lazy-learning algorithms are techniques, like nearest neighbors (NN), where all the computational burden is in the evaluation of the classifier and not in its creation: such methods generally exploit comparisons with historical data to perform the classification of a new observations; given these premises, it is apparent that lazy-learning approaches cannot be adopted in nodes since: (i) the evaluation there need to be performed as quick as

possible; (ii) nodes do not have access to the whole network data, and therefore the comparisons on that level can be made only on a local, smaller database (not always available) leading to suboptimal classification performances.

Considering this complex scenario, we provide here some general guidelines on which types of algorithms are suited for remote (nodes) and central (main) units. As we will see throughout this chapter, among the two sets of techniques we highlighted before, there are better choices per se and with respect to the specific application; indeed, both FB and DB methods can be simple or cumbersome depending on their formulations. In fact, as FB algorithms with an high number p of features may be prohibitive, the same applies to DB algorithms where an high number N of training examples are considered.

Nonetheless, various *data reduction* procedures may be applied in order to reduce the complexity of learning algorithms. For example, reduction techniques such as symbolic aggregate approximation (SAX) or discrete Fourier transform (DFT) are computationally simple and induce an approximation that may be considered as acceptable for most real applications in power systems. Given the crucial importance of parsimony in power systems, the following section is devoted to discuss data reduction techniques.

3 DATA SOURCES

One of the main practical issues in modern time-series classification is the problem of *time and space complexity* of data. In general, dealing with huge datasets is computationally expensive and, under some conditions, even unfeasible, especially with resource-constrained hardware. On the other hand, in most real-world problems, the informative content of a dataset is generally *sparse* (i.e., the useful information size is much "smaller" than the size of original dataset). For these reasons, research in big data classification has been focused on developing suitable techniques to optimize and reduce the data representation. In the related literature, the procedures related to translating data into simplified and informative representation are called *data reduction*.

In the context of power systems, this problem arises due to huge amount of data coming from PMUs and other sensors. As represented in Fig. 5, PMUs (and other sensors) are connected to local phasor data concentrators (PDCs) and, jointly, to a corporate PDC, receives data from different PDCs. To give insights on how this structure generates an amount of data in the realm of big data, only one PCD collecting data from 100 PMUs of 20 measurements each at 30-Hz sampling rate generates over 50 GB of data 1 day [29]. Hence, data reduction is fundamental to reduce data storage and to best capture the interaction between different PMU locations.

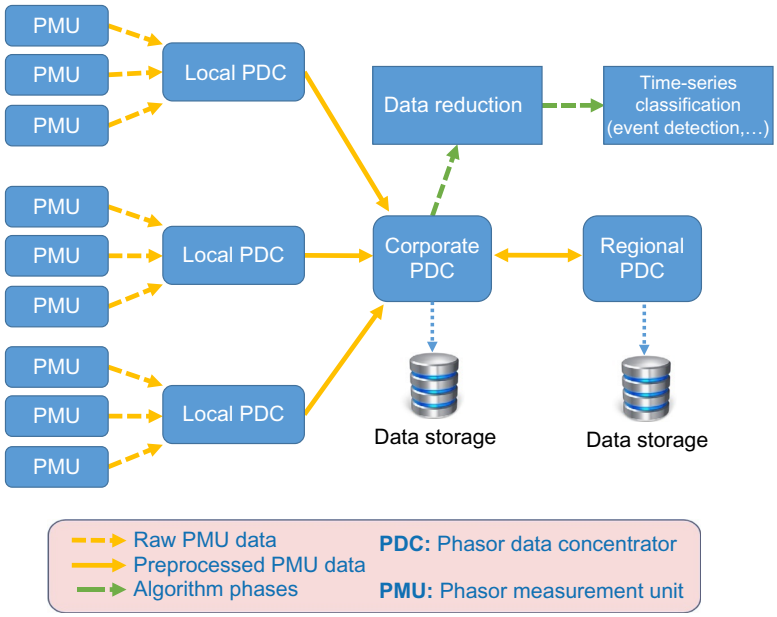

FIG. 5

Typical power system scheme data flow. Detailed structure of a EMS where the data streams and the procedure units are represented (note that other data sources than PMUs may be in place).

For the sake of clarity, and to avoid confusion due to different notations used in related literature, we distinguish two data reduction techniques in which we are interested in: *dimensionality reduction* and *numerosity reduction*. In this section we will present the most important time-series representations for these two types of reduction; for a more exhaustive review, we refer the interested reader to [30, 31].

3.1 Dimensionality Reduction

We refer to dimensionality reduction (DR) when dealing with FB techniques. More in detail, let us consider a design matrix $D \in R^{N \times p}$, with a high-dimensional feature space (e.g., when p is very large; generally $p > 1000$); DR aims at finding a subset of informative features (*feature selection*) or, more generally, informative lower-dimensional structures through linear (e.g., principal/independent component analysis) and nonlinear (e.g., manifold learning) data transformation approaches.

In this context, although the space complexity can be overwhelming, the main issue is the renowned *curse of dimensionality* [26]. It manifest itself in various ways, which all causes high variance and high bias of classifiers resulting in poor classification performance. In fact, when p is high, all feasible training samples sparsely populate the feature space and the concept of locality vanish.

This problem can be easily seen with the k-NN classifier and considering features uniformly distributed in a p-dimensional unit hypercube: when $p = 1000$, in order to capture for example 10% of data to evaluate local average, it is necessary to consider the 99.7% of the range of each feature.

3.1.1 DR Techniques Review

As already stated earlier, the informative content of a signal is often embedded in a low-dimensional space that can be isolated through DR techniques. Among the most popular methodologies, we mention here generalized discriminant analysis [32], independent component analysis (ICA) [33, 34], kernel PCA [35, 36], linear discriminant analysis (LDA) [37], multidimensional scaling (MDS) [38], and principal component analysis (PCA) [39, 40]. In the realm of nonlinear approaches, manifold learning, whose objective is to learn the hidden manifold described by the data [41–44], has been gaining lot of attention in the past recent years.

A simpler, but often equally efficient, DR approach is to remove redundant (correlated) features, selecting a subset of relevant features, instead of finding the underlying low-rank structure of the data at hand. Well-known feature selection techniques are backward feature elimination, forward feature construction, minimum redundancy maximum relevance (mRmR), just to provide some examples. Interestingly, random forests (RFs), besides being among the most effective classifiers, are also powerful instruments for feature selection [45].

For an exhaustive description, we refer the interested readers to [46]. In the context of power systems, recently, Zhou et al. [2, 23] adopted a data-driven feature-based approach combining mRmR reduction technique with an ensemble of bundle classifier (EBC), which combines individual classifiers in order to handle the heterogeneity of the PMU data.

3.2 Numerosity Reduction

We refer as numerosity reduction (NR) when aiming at reducing data volume by choosing alternative, smaller forms of data representation of the signals at hand. It differs from DR in the sense that it aims at finding a different representation of time series and/or reducing the number N of training examples needed for classification without reducing accuracy. For example, consider a collection of N input univariate time-series $\{\mathbf{x} \in \mathbb{R}^n\}_{i=1}^{N}$. Data reduction techniques aim at reducing n and/or N.

3.2.1 NR Techniques Review

In this framework, to reduce both n and N, parametric and nonparametric approaches may be employed for NR. Parametric approaches model the time series using a parametric model such as discrete wavelet transform (DWT), DFT,

and log-linear models to cite the most common examples. Then the complexity space reduces from $O(n)$ to $O(p)$ where $\mathbf{x} \in \mathbb{R}^p$ is p-dimensional vector of parameters of the model. Example of nonparametric approach is histogram or, simply, sampling.

Another family of NR approaches is *symbolic representation*, for which SAX [47, 48] is the most know technique. SAX technique mainly consists of three phases:

- signals standardization in order to obtain a zero mean and unit variance signal;
- piecewise aggregate approximation (PAA) [49] described in the following; and
- symbolic mapping through discretization on amplitude domain.

After normalization, in the PAA phase, a signal $\mathbf{z} = z_1, \ldots, z_n$ let $\overline{\mathbf{z}} = \overline{z}_1, \ldots, \overline{z}_p$ of length s is discretized on time in p frames in order to obtain a vector $\overline{\mathbf{z}} = \overline{z}_1, \ldots, \overline{z}_p \in \mathbb{R}^p$. Formally, the resulting ith element $\overline{z}_i$ is defined by the mean of ith interval:

$$\overline{z}_i = \frac{p}{s} \sum_{j=\frac{s}{p}(i-1)+1}^{\frac{s}{p}i} z_j \tag{2}$$

Then, the SAX representation procedure (i.e., the discretization on amplitude domain) can be summarized as follows. Let a_i denote the ith element of the alphabet $\mathcal{A}$, with $|\mathcal{A}| = \alpha$. The mapping from the PAA approximation to the correspondent word $\mathbf{x} = x_1, \ldots, x_p$ of length p is obtained as follow:

$$x_i = a_j \quad \text{iif } \beta_{j-1} \leq \overline{z}_i < \beta_j \tag{3}$$

where $\left\{\beta_j\right\}_{j=1}^{\alpha-1}$ are breakpoints tuned to have symbols with equiprobable occurrence. One of the advantages of introducing the *SAX representation* is that a new distance measure—which is a lower bound of Euclidean distance—can be immediately defined. Let $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ be two time series of same length n and $\mathbf{x}^{(1)} = x_1^{(1)}, \ldots, x_p^{(1)}$ and $\mathbf{x}^{(2)} = x_1^{(2)}, \ldots, x_p^{(2)}$ be their *SAX symbolic representation*; the SAX distance is defined as:

$$D_{\text{SAX}}(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) = \sqrt{\frac{n}{p} \sum_{i=1}^{p} \text{dist}\left(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}\right)^2} \tag{4}$$

Another popular symbolic approach is the symbolic Fourier approximation (SFA) [50]. The SFA accepts the same parameters p and α as SAX. In this case, p represents the number of Fourier coefficients (real and imaginary) used. Naturally, each sliding window is normalized to have a standard deviation of one

to obtain amplitude invariance, before applying SFA. Provided the parameters, the SFA symbolization is carried out in due main steps:

1. preprocessing phase called multiple coefficient binning (MCB) discretization, and
2. SFA transformation.

In the phase (1) the p coefficients (real and imaginary) c_i are extracted for all the training time series and histogram is built for each c_i, where each bin corresponds. Then each histogram is used to infer the breakpoints β_{ij}, $i = 1, \ldots, wp$, $j = 1, \ldots, \alpha + 1$ in order to make symbols equiprobable. In the phase (2) each coefficient is extracted and it mapped to a symbol according to breakpoints found in the MCB phase. Then the string representing the time series is formed by the sequence of coefficients. Thus, to find a second-order resolution (two coefficients) there are two real plus two imaginary coefficients resulting in a word of length four. SFA presents some important difference from SAX: first of all, the time complexity (to transform a single time-series $\mathbf{z}$) is $O(n \log n)$ while SAX time complexity is $O(n)$. However, provided the same word length, SFA best represents the raw signal as it does not apply any piece-wise discretization, but expresses a linear combination of continuous Fourier functions through the learned coefficients. Moreover, in opposite to SAX, if we wanted a finer resolution increasing p, it would not be necessary to recalculate all DFT coefficients as the symbols of a smaller word length are always a prefix of the larger word lengths.

Regarding the reduction of N, the most popular approaches are clustering and dictionary learning. Time-series clustering aims at finding groups (clusters) in which data can be divided. A way to speed up clustering approaches (that are generally in the realm of *lazy-learning* approaches [26]), "mean" (or centroid) time series for each cluster is usually taken as representative of each group: this generally requires $\widetilde{M}$ comparisons instead of N, and, since $N \gg \widetilde{M}$, this decreases considerably the time to perform the clustering. Similarly, dictionary learning techniques find a sort of base of $\widetilde{M}$ signals, from which a given signal can be represented as a linear combination. Indeed, in the classification context, the *supervised* dictionary learning aims at learning a dictionary containing the elements which best represent the classes and thus they find a *discriminative* representation.

4 CLASSIFICATION METHODS

4.1 Feature-Based Methods

Assuming a dataset of N time-series $\mathbf{z}_k$, $k = 1, \ldots, N$, and M possible target classes $\mathbb{Y} = \{y_1, \ldots, y_M\}$ that jointly describe the classification problem of interest,

FB methods focus on finding a compact description $\mathbf{x} = [x_1, \ldots, x_p] \in \mathbb{X}$ of the time-series $\mathbf{z}_k$ such that $p \leq n$ (and typically $p \ll n$); all these N observations are collected into a matrix $D \in \mathbb{R}^{N \times (p+1)}$, called *design matrix*, as defined in Eq. (1). In practice, D represents the supervised learning phase of the procedure and it is exploited to define the rule $f(\bullet) : \mathbb{X} \rightarrow \mathbb{Y}$ according to a chosen classification method [26] as better detailed in the following.

In this category, the most employed classification algorithms are k-nearest neighbor (k-NN) [26], support vector machines (SVMs) [1, 51], relevance vector machine (RVM) [52], decision trees (DT) [53], RF [54], logistic regression (LR) [55], Gaussian processes (GP) [56], and deep neural networks (DNN) [57]. We refer the interested reader to the literature references for further details on the specific methods and the general textbooks [26, 58].

4.1.1 Metrics-Based Approaches

The techniques earlier are not straightforwardly applicable to continuous time series since they need input vectors of fixed length, and this justifies their inclusion in the FB methods.[1] As a matter of fact, even if the time series in input to the classifier were discrete time and of fixed length, the accuracy performance would be poor due to two main reasons: first, it is common to consider long sequences of samples of $n > 100$ or even $n > 1000$; in these cases the space spanned by the time series is too large and sparse incurring in the aforementioned "curse of dimensionality" problem. Second, considering the discrete values as independent features would be not reasonable, since they do not provide any information per se about the characteristics of the signal, since time-series values are strongly highly correlated on time and the feature extraction phase is exactly designed so as to highlight this correlation.

The flowchart of the FB classification procedure is given in Fig. 6: after a preprocessing phase on the raw signal characterized by the presence of a low-pass filtering operation to reduce measurement noise and the windowing procedure, the proper feature extraction task is performed on the time series,

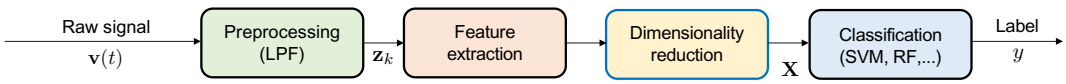

FIG. 6
FB methods. Operation flow of FB classification procedures.

[1] This is partly true for deep learning approaches: recent deep neural networks schemes can avoid the feature extraction phase.

resulting in a compact set of signal descriptors, which undergo the classification phase.

The main advantages of the FB methods clearly reside in the compactness in the representation able to characterize the signal. Typical examples of features are follows.

- Sample features: sample variance/mean/RMS value of the time series.
- Energy/power features: energy value from the DFT coefficients, power spectrum bands.
- Correlation features: number/location/width of (prominent) peaks in the autocorrelation function (repetitive and periodic signals present a peak in the autocorrelation function); correlation parameters among different signal dimensions.

Unfortunately, though, the FB approach presents also several drawbacks:

- features must be defined ad hoc depending to the task[2];
- high dimensionality;
- nonstationarity of time series; and
- time structures are not considered.

It clearly appears from these lists that most of feature-based ML techniques are not adapted to exploit time structures (i.e., patterns), which are an *intrinsic* and *distinctive* characteristic of each time series. In this regard, the first attempts to exploit patterns can be found in Refs. [59–61]. In particular, in Kadous [60] parameterized events are extracted from multivariate time series: these events are clustered in the parameters space and the resulting prototypes are used as basis to build classifiers. Instead, Kudo et al. [61] maps multivariate time series into binary vectors: the space value-time is represented by a grid and each element (of the vector) is associated with one cell of the grid count: if the signal passes through the corresponding cell, the element is true (1) otherwise it is false (0). Then, these converted binary vectors are used as the basis for the classification.

More in general, a limitation to these methods stem from the fact that classification rules are extracted taking into account absolute time values, leading to the inability of handling situations where particular behaviors happen at different time values: Geurts [62] argues that many time-series classification problems can be solved by detecting and combining local properties (patterns) on time series, and proposes a procedure that captures the information of shift-invariant patterns using DTs over piece-wise constant time series.

[2]This is one of the main reasons that favors the usage of deep learning in complex problem such as natural language processing and computer vision: in these fields, the definition of informative features has required at least 20 years of research.

More recently, *interval features* (IFs) have been introduced in order to capture temporal information [63–66]. These features are common statistics such as mean, variance, slope but they are calculated over random *intervals* exploiting a boosting procedure. The first idea on IFs was presented in Rodríguez et al. [63], later expanded by Rodríguez and Alonso [64] and Rodríguez et al. [65] using classifiers such as DTs and SVMs applied on the features extracted from binary ensembles. However, as discussed in Deng et al. [66], the number of candidate splits is generally large and thus there can be multiple splits having the same ability of separating the classes. To cope with these issues, Deng et al. [66] introduce an additional measure able to better distinguish among IFs. Another problem in boosting IFs is the size of the relative space that is $O(n^2)$, where n is the length of a time series. In Rodríguez et al. [63] the feature space is reduced to $O(n \log n)$ using only intervals of length equal to powers of two. Deng et al. [66] consider the same approach of random sampling strategy used on RF [54] further reducing the feature space at each node to $O(n)$.

4.1.2 Occurrence Counting Approaches

Another type of approaches to classify time series is the so-called bag-of-words (BoW), also bag-of-features (BoF), nowadays used in image classification and document classification and classically developed in the context of natural language processing. BoW consists in representing data (images in computer vision or documents in natural language processing) using a histogram of word occurrences, where a *word* is a task-dependent element [67], namely a proper textual word in language processing or the image description through intensity local gradients in computer vision. After this encoding, the classification task is reduced to computing a histogram-based similarity (typically using Euclidean distance).

With respect to the two steps of (i) conversion of the time series into a BoW representation (i.e., an histogram of the word occurrences) and (ii) training of a classifier (such as SVM, RF, kNN) upon BoW features, we particularly focus on step (i) in the following, since step (ii) is similarly performed by all methods using RF, SVM, or some other common classifier over the word histogram. Indeed, several BoW-inspired techniques have been recently investigated in order to extract local and global features [68–76]. Hereafter, a brief overview of the main contributions is given.

In the computer vision field, the BoW technique is used for image classification [77] often in combination with the scale invariant feature transform (SIFT) technique. SIFT is a *covariant* detector, which extracts local features (keypoints) that are robust to noise (e.g., changes in illumination) and invariant to affine transformations and scale. As a general note, BoW methods ignore temporal ordering, which may cause that patterns in observed time series or images

are not identified. Nonetheless, some BoW-based works try to indirectly remove this lack, although to a limited extent. For example, in Bailly et al. [73, 76] a variant of SIFT for time series is applied and local features (keypoints' descriptors) are extracted with a procedure very similar to SIFT and BoW approach is used over the SIFT descriptors. This choice is motivated by the fact that SIFT captures local structures while BoW allows to describe the global behavior of the time series. Furthermore, in Bailly et al. [76], the same authors adopt dense-SIFT-like descriptor: the main difference with the previous work is that keypoints no longer correspond to extrema but are rather extracted at all scales every time step on Gaussian-filtered time series. This approach in general leads to more robust global descriptors, especially when local extrema can be found (when signal are very smooth).

In Wang et al. [69], DWT is applied on sliding windows of the time series and the resulting DWT coefficients form a word for each window (segment). In the training phase all the DWT segments are clustered through k-means in order to obtain a word dictionary $\mathbf{D}$. In the classification phase each DWT window is assigned to the nearest word in $\mathbf{D}$, to build a histogram that is used to computed the similarity; the classification is finally carried out using 1-NN.

A BoF framework is proposed also in Bailly et al. [70], which combines IFs and BoW. Here, there are extracted IFs and start/end time points over random subsequences of random length, and a supervised codebook of class probability estimate (CPE) histogram is built in the training phase: for each sequence (of the time series) a CPE is found using an RF classifier. Then, all the CPEs are quantized in order to form a different histogram for each class, which are concatenated into a single histogram of each time series and are used as features in combination with other global features. Finally, the employed classifier is RF.

A symbolic multivariate time series (SMTS) method is discussed in Baydogan et al. [72]: each time series is represented by a feature space, which contains the time instants, the time-series values, and the first difference values, all collected in a design matrix D. Then D is input to a symbolic discretization which is obtained using tree-based classifiers (supervised discretization). Then the classification is performed using a common BoW approach based on histograms of symbols. Its total computational complexity is due to the number of trees, the number of training instances, and the number of time-series subsequences extracted. Then, multivariate time series are mapped into a feature matrix, where features are vectors containing a time index $\overline{t}$, the values, and the gradient of time series at t for all dimensions. The feature space is finally partitioned into regions (i.e., symbols) by an RF classifier. An appreciable property of SMTS is that it does not require tuning parameters, while one main drawback of this

representation is the possibly high dimensionality, which limits its application for large datasets.

In Schäfer [74] the bag-of-SFA-symbols (BoSS) is introduced. An univariate time-series $\mathbf{z}$ is represented by SFA words and then an histogram is built. However, since this approach is $O(N^2n^2)$ for training, $O(Nn)$ for classification, in Schäfer [75] it is presented a "scalable version" named BoSS-VS that uses vector space models (VSMs). In this case, once the BoSS histogram is obtained, for each SFA word w, the word (called *term*) frequency *tf* of w in a certain class c_i is computed, together with the ratio *idf* given by the total number of classes divided by the number of classes in which w appears. Then, the *tf-idf* measure is obtained by the product $tf \cdot idf$: this measure is used to weigh the word frequencies in the vector to give a higher weight to representative words of a class. The motivation behind this choice is that an high *tf-idf* for a word w means that w appears with an high frequency in a specific class c_i, while low *tf-idf* values means that w in common in all classes. When a new observation $\mathbf{z}_{\text{new}}$ arrives, the BoSS histogram and the relative *tf* vector are computed. Then the *cosine similarity*[3] is obtained in order to predict the nearest class. Using this model (named "term frequency inverse document frequency," *tf-idf*, model), the complexities of training and classification reduce to $O(Nn^{3/2})$ and $O(n)$, respectively.

In Lin et al. [68] and Senin and Malinchik [71] time series are mapped into SAX words through a sliding windows partitioning, which are used to build histograms of n-grams: for each time series an histogram counts the frequency of occurrences of each SAX word and thus each time series is represented by its histogram. In particular, in Senin and Malinchik [71], the authors combine SAX and VSM [78] exploiting the *tf-idf* model [79, 80], weighing bags, and cosine similarities as metrics. This technique has a parameter space of dimension $O(n^2)$ and needs to recompute all SAX coefficients for each new choice of parameters p (number of frames in the PAA representation) and α (cardinality of the alphabet); moreover, the training time is $O(Nn^3)$ where N is the number of training instances.

4.1.3 Dynamics-Based Approaches

The last set of FB techniques we present in this review explicitly takes into account the *dynamics* of the signals and comes from *dynamical systems* and *signal processing* theory. In the context of dynamical systems and identification theory, in the past decades much attention has been conveyed on modeling stochastic processes (whose realizations are time series) in order to *predict* their future trends

[3] Given two vectors $\mathbf{x}_1$ and $\mathbf{x}_2$, both in , the cosine similarity is defined as $\cos\beta = \frac{\langle \mathbf{x}_1, \mathbf{x}_2 \rangle}{\|\mathbf{x}_1\| \cdot \|\mathbf{x}_2\|}$, where $\langle \bullet, \bullet \rangle$ is the inner product.

and values. The most common models are Auto-Regressive (AR), Auto-Regressive-Moving-Average (ARMA), and Auto-Regressive-Integrated-Moving-Average (ARIMA) models [81], just to give some examples. With such approaches, the coefficients themselves of the fitted model are used as features for a suitable classifier [82, 83] or are used to build a more complex generative model [84].

In more detail, Roberts [84] takes a Bayesian point of view and proposes a hierarchical model that consists of a feature extraction stage and a generative classifier, probabilistically linked by a latent feature space. The classifier is implemented as an hidden Markov model (HMM) with Gaussian and multinomial observation distributions defined on a representation of AR models. The HMM is used to model the correlation between adjacent windows (subsequences), that is, this model assumes that time series are consecutively extracted from an unique flow.

In a similar way, signal-processing transformations such as DFT or DWT are applied to the raw signals to obtain coefficients that can be exploited in training suitable classifiers [85–87]. Interestingly, DWT results to be more suitable for nonstationary time series and, conversely w.r.t. DFT, is ideal for identifying highly discriminant local time and scale features [88]. We note that DWT and DFT and dynamical models can be seen as dimensionality reduction procedures: in the next section, we will revise these concepts from the point of view of distance-based methods.

Eruhimov et al. [89] gathered the most known features deriving from the presented methods such as statistical moments, wavelets coefficients, PCA coefficients, Chebyshev coefficients, and the original values of time series and built a classifier from them. However, this method can be accurate at the cost of complexity and a feature selection procedure is needed to reduce the dimensionality.

Although time-pattern (dynamic) information has been considered in literature, most of feature-based models present common limitations due to the nature of time series. In fact, the presence of variability in the time series causes these methods to be ineffective to cope with common issues. The variability arises because of the stochasticity of the process generating the time series, nonstationarity of time series, and nuisance factors. To give a practical example, the most effective FB methods that exploit time-patterns presented in this chapter are not able to deal with variable time-series lengths and the other methods which can handle this issue exploits only global statistics making them ineffective with nonstationarity.

Finally, perhaps the most limiting issue of feature-based methods is that the feature extraction phase can be demanding both in terms of memory and computational burden. These factors could make feature-based methods not

Table 3 Main characteristics of feature-based methods which have been reviewed in this work

Methods	Characteristics
Time-pattern features	First attempt to capture local structures
	Capture temporal information
	Feature space is big ($O(n)$)
Interval features	Feature extraction can be onerous
	Local and global structures are captured
	Histogram extraction is onerous
Bag-of-features	Time-patterns not considered
Dynamic features	Encode information about dynamics
	Capture behavior in the frequency domain
Frequency domain features	FFT/DWT are efficiently implemented
	Good accuracy
Ensemble of features	Feature extraction is very onerous

suitable for resource-constrained devices. In Table 3 the main peculiarities of each FB methods are summarized.

4.2 Distance-Based Methods

DB methods can be clustered into three groups:

- *Purely distance-based*: These methods are based on the direct computation of ad hoc defined distances over raw time series.
- *Reduction distance-based*: Such methods are based on the computation of opportunely defined distances over a reduced representation of raw time series.
- *Parametric distance-based*: With this type of DB approaches, raw signals are represented with a combination (generally linear) of basis signals (e.g., sine functions in the Fourier series representation). The coefficients of different representations (parameters) are used for the computation of ad hoc defined distances.

A general picture of the dataflow for DB methods is given in Fig. 7 and the three groups will be discussed in detail in the following of this section.

4.2.1 Purely Distance-Based Methods

Purely DB methods performs the classification task by adopting a classifier that exploits an opportunely defined distance applied to the raw time-series $\mathbf{z}$. Thus, in this case, we have $p \equiv n$ and we refer to $\mathbf{x}$ as the time-series $\mathbf{z}$ (the map $\mathbf{z} \rightarrow \mathbf{x}$ can be seen as the identity map). Here, we consider a set of variable-length

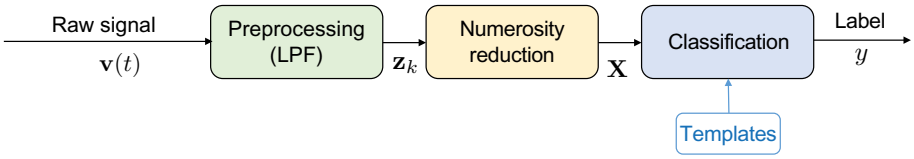

FIG. 7
DB methods. Operation flow of DB classification procedures.

training time series and the corresponding label $D = \{(\mathbf{x}_i, y_i), i = 1, \ldots, N\}$. As mentioned earlier, purely DB methods are based on the computation of a distance over the raw time series. In choosing a distance, the most straightforward approach is to adopt an *Euclidean distance*; however, this choice has many drawbacks: Euclidean distance to be computed requires time series of equal lengths; moreover, even when comparing two series of equal length, Euclidean distance can be an unfortunate choice since it does not considers common nuisance factors such as warping [90].

For the previous reasons, a more popular approach for distances is DTW [91]. DTW measures the similarity between two time series with, possibly, different lengths by warping the time axis of one (or both) sequences to achieve alignment between the two. DTW provides a *similarity score*, an index on how similar two time series are: in order to define the similarity score, let us consider two time series $\mathbf{x}^{(1)} = \{x_1^{(1)}, \ldots, x_n^{(1)}\}$ and $\mathbf{x}^{(2)} = \{x_1^{(2)}, \ldots, x_m^{(2)}\}$ and let us define a grid $\mathcal{G} = [n] \times [m]$. A *warping path* wp in $\mathcal{G}$ is a sequence $wp = (\mathbf{p}_1, \ldots, \mathbf{p}_l)$ with points $\mathbf{p}_k = (i_k, j_k) \in \mathcal{G}$ s.t.:

$$\begin{array}{ll} \mathbf{p}_1 = (1,1) \text{ and } \mathbf{p}_l = (n,m) & \text{(boundary conditions)} \\ \mathbf{p}_{k+1} - \mathbf{p}_k \in \{(1,0),(0,1),(1,1)\} & \text{(warping conditions)} \end{array}$$

$\forall k | 1 \leq k < l$. The cost of "warping" $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ along the warping path s is given by

$$d_s(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \sum_{(i,j) \in s} \left(x_i^{(1)} - x_j^{(2)}\right)^2 \tag{5}$$

where $(x_i^{(1)} - x_j^{(2)})^2$ is called *local transformation cost*. Then, the DTW similarity score is defined by

$$d\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) = \min_s d_s\left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}\right) \tag{6}$$

Regarding the choice of the classifier, the most common choice is 1-NN combined with DTW (NN-DTW). Notably, even if NN is generally considered as one of the simplest approaches to classification, in many papers NN-DTW outperforms more sophisticated approaches when dealing with time-series

classification [90, 92, 93]. One of the issues of NN-DTW is its computational cost: the DTW is $O(n^2)$ and it has to be evaluated for each training example in order find the NN. In order to alleviate the aforementioned issue, various approximations of DTW [94–98] have been introduced: the most promising are FastDTW [95] and SDTW [97]. FastDTW adopts a multiscale approach that recursively projects a solution from a coarse resolution and refines the projected solution. FastDTW time and space complexity are $O(n)$; SDTW, instead, extracts keypoint descriptors (similarly to SIFT [67], a popular approach in computer vision) and uses them to reduce complexity.

Beside approximations, several other extensions and modifications of DTW have been proposed: Keogh and Pazzani [99] proposed the derivative dynamic time warping (DDTW), which transforms the original time series into a first-order differences time series, in order is to avoid ill-conditioning; ill-conditioning is a common issue in DTW when dealing with noisy and long signals due to the fact that single points of one of the compared time series could be mapped onto a large subset of the other time series leading to poor alignments. Jeong et al. [100] proposed a penalty-based DTW (WDTW), which adds a multiplicative weight penalty in order to penalize points with higher phase difference between a reference point and a testing point. This has the aim to prevent minimum distance distortion caused by outliers. Other used similarity measures are edit distance with real penalty (ERP) proposed by Chen and Ng [101] and Chen et al. [102] and time warp edit distance (TWED) proposed by Marteau [103]. ERP is a variant of L1-norm, which can support local time shifting. It can also be viewed as a variant of EDR [101] and DTW, but it is a metric distance function. TWE distance is an elastic distance measure (efficiently implemented using dynamic programming) which, unlike DTW, is also a distance. It allows warping in the time axis and combines the edit distance (defined for time series) with L_p-norms. Marteau [103] also provides a lower bound for the TWED measure which allows to operate into down-sampled representation spaces in order to fasten the algorithm.

As we already states, DTW is not a distance measure and this implies that it cannot be employed with kernel methods [104], where kernel must be positive definite. Moreover, time series of different length cannot be compared. In this context, Cuturi et al. [105] propose a new family of kernels between variable-length time series, called alignment kernels, which consider the soft-max of the score of all possible DTW-based alignments to consider the three of the scores of all possible alignments. However, the computation of such kernels can be performed in quadratically, and motivated by this limitation, Cuturi [106] provides an efficient version of it.

Although the usage of kernel methods combined with global alignment allows to consider variable-length time series and disturbances which cause time

warping, these do not consider the *dynamics* or patterns. Indeed, albeit the term "dynamic" (deriving form dynamic programming), DTW has nothing which considers the *dynamics* of time series we want to classify. Soatto [107], among other contributes on defining distances for nonstationary time series, also introduces the DTW under dynamic constraints (DTWUDC), which constrains the DTW to follow a dynamical system. More in detail, in Soatto [107] it is assumed that the data (of two time series) are outputs of dynamical models driven by inputs that are warped versions of some common function. Thus, given two univariate time series $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ $(i = 1, 2)$, he assumes that there exists the dynamical model (linear in the parameters)

$$\begin{cases} \dot{h}_i(t) & = Ah_i(t) + Bu(w_i(t)) \\ \mathbf{x}^{(i)}(t) & = Ch_i(t) + n_i(t) \end{cases}$$

where A, B, C are suitable matrices, h_i are the state functions, $n_i(t)$ are noise processes, $w_i(t)$ are warping functions, u is a common input, and $\mathbf{x}^{(i)}$ is the time series i. Then, the distance can be evaluated in two stages to fit $u_i(t)$ and then $w_i(t)$. In the section dedicated to parametric distance-based methods we will see other methods, mainly kernel methods, on dynamical systems which capture the dynamic essentials of the time series.

4.2.2 Reduction Distance-Based Methods

As we detailed earlier, although the 1-NN-DTW classifier is remarkably difficult to beat, it presents computational issues which prevent its usage in resource-constrained systems. Moreover, shape-based methods typically fail to provide satisfactory results for long time series, as the weight of discriminative "local" structures decreases. In this context, albeit various approximation of DTW was presented, the 1-NN still remains a bottleneck, as it requires the comparison with all the training time series, mostly when the length n and the number N of the time series are large. Moreover, it also requires space to store the entire dataset, which is unfeasible for most resource-constrained devices.

As we have seen in Section 3, data reduction techniques aim at reducing n and/or N. The main idea of reduction distance-based methods is to reduce the time series in a parsimonious in a new representation space and compute a suitable distance in this space. In this part, we will discuss distance-based methods which reduce n or N or both. Before reviewing these techniques, we notice that some of the techniques we have previously exposed may also fall into this category. For example, this is the case of VS-BOSS, in which the time-series $\mathbf{x}$ is mapped into histograms and the histogram-similarity is computed in order to classify $\mathbf{x}$.

As we saw previously, symbolic representations such as SAX are very useful for NR. In this context, the simplest classifier is the 1-NN-SAX classifier, that is the 1-NN on the space of symbolic representation endowed with the metric d_{SAX}.

However, as the Euclidean distance, it is not robust to time distortions or more simply time shifts. Moreover, as we pointed out, when N is large, using SAX approximation may not be enough. Thus, it has been crucial to find some sparse representation of the space of training examples.

The rationale under the reduction of N is to find a subset of canonical $k \ll N$ examples (*templates*) which best describe training set without loss of information (in the sense that they are *sufficient*). Fundamentally there are two directions to find this templates, that is, unsupervised and supervised. In the unsupervised approach, *templates* are found trough clustering techniques regardless the task at hand. On the other hand, supervised approaches aim at finding also the most discriminative templates for the classification tasks, that is, the templates which best represent each class. Naturally, supervised methods are most suited for the classification task. In literature, there are two (at least) different definitions of template, that is *shapelet* and *dictionary*. They rely on the same idea, but is tackled with different approaches.

The concept on *shapelet* has been developed in the recent literature [108–114]. Shapelets are subsequences of time series which are maximally (in some sense) representative of a certain class and thus are useful to classify unlabeled time series; shapelets are maximally representative in the sense of the *information gain* criteria also used to train decision trees and RFs [53].

The main advantages of using shapelets are that 1-NN with all the training instances is avoided in favor of the computation of the distance to the shapelets, which represent each class and it is phase-invariant contrary to simpler techniques such as 1-NN with Euclidean Distance (or SAX distance). On real problems, the speed difference of classification can be greater than three orders of magnitude [108]. However, despite the fast classification, the training of the shapelets is onerous. In the first work where shapelets for classification was introduced, the worst-case scenario for the training time was $O(N^2n^3)$ where N is the number of time series in the dataset and n is the length of the longest time series in the dataset. In order to reduce the training complexity, various extensions have been proposed. Among all, in [112], SAX is used to find suboptimal shapelets reducing training complexity to $O(Nn^2)$: in this case, time series are mapped to a low-dimensional space of SAX words and shapelets are found directly on this space. Then the distance used for classification is the d_{MAX} defined earlier.

4.2.3 Dictionary Learning

The other approach we find in the literature is called *dictionary learning*, whose aim is to learn a sparse representation of the time series in terms of a basis of signals and express the input signals as a linear combination of basic elements belonging to a set called dictionary. Before continuing we notice that we refer to $\mathbf{x}$ as the time-series $\mathbf{z}$, that is $p = n$. Dictionary learning can be categorized in two

approaches: unsupervised and supervised. In order to understand their difference, in the following we briefly present the main concepts of these frameworks which are detailed for example in Refs. [115–117]. Suppose of having N univariate fixed-length training time-series $\{\mathbf{x}^{(i)} \in \mathbb{R}^n\}_{i=1}^N$ associated with binary labels $\{y_i \in -1, +1\}_{i=1}^N$. In order to find an optimal (in the sense of mean square error) and sparse representation, through dictionary $\mathbf{D} = \left[\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_{\tilde{M}}\right]$ of signal $\mathbf{x}$ we can solve the convex optimization problem

$$\min_{\boldsymbol{\alpha}, \mathbf{D}} \sum_{i=1}^{N} \| \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i \|_2^2 + \lambda \| \boldsymbol{\alpha}_i \|_1, \quad \text{s.t. } \| \boldsymbol{d}_i \|_2 = 1 \tag{7}$$

where ℓ_1-norm is used for $\boldsymbol{\alpha}$ since encourages sparsity [26] inducing noninformative $\boldsymbol{\alpha}_i$ to be zero. Once obtain optimal $\boldsymbol{\alpha}^*$, $\mathbf{D}^*$, we can solve the classification task solving

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i, \boldsymbol{\alpha}^*, \mathbf{D}^*, \boldsymbol{\theta})) + \lambda_2 \| \boldsymbol{\theta} \|_2^2 \tag{8}$$

where $L(\cdot, \cdot)$ and f are an opportune loss function and the predicting function, respectively, which together define a classifier and $\boldsymbol{\theta}$ parameterizes the model f. Common choices of f are

1. linear models in $\boldsymbol{\alpha}$: $f(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\theta}) = \boldsymbol{w}^T\boldsymbol{\alpha} + b$ with $\boldsymbol{\theta} = \boldsymbol{w} \in \mathbb{R}^k, b \in \mathbb{R}$.
2. bilinear models in $\mathbf{x}$ and $\boldsymbol{\alpha}$: $\mathbf{x}^T\boldsymbol{W}\boldsymbol{\alpha} + b$ where $\boldsymbol{\theta} = \boldsymbol{W} \in \mathbb{R}^{n \times k}, b \in \mathbb{R}$.

This approach is called unsupervised since the dictionary $\mathbf{D}^*$ is obtained to find a sparse representation of N training time series *independently* of the classification task. However, as pointed by Mairal et al. [116], the dictionary found with this procedure is optimal in the sense of *reconstructive* tasks but not for *discriminative* one, that is, classification.

In order to tackle this issues, supervised dictionary learning has been introduced in order to learn a discriminative dictionary exploiting the class label information. In [116], the authors propose an approximation solution of formulation (Eq. 9) which learn jointly $\mathbf{D}$ and $\boldsymbol{\theta}$:

$$\min_{\mathbf{D}, \boldsymbol{\theta}} \sum_{i=1}^{N} L(S^*(\mathbf{x}_i, \mathbf{D}, \boldsymbol{\theta}, -y_i) - S^*(\mathbf{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i)) + \lambda_2 \| \boldsymbol{\theta} \|_2^2 \tag{9}$$

where $S^*(\mathbf{x}_i, \mathbf{D}, \boldsymbol{\theta}, y_i) = \min_{\boldsymbol{\alpha}} L(y_i, f(\mathbf{x}_i, \boldsymbol{\alpha}_i, \mathbf{D}, \boldsymbol{\theta})) + \lambda_0 \| \mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i \|^2 + \lambda_1 \| \boldsymbol{\alpha}_i \|_1$. Then the classified label $\hat{y}$ of a new time series $\mathbf{x}_{\text{new}}$ is given by

$$\hat{y} = \arg\min_{y \in \{-1, +1\}} S^*(\mathbf{x}_{\text{new}}, \mathbf{D}, \boldsymbol{\theta}, y)$$

Other supervised approaches are discriminative KSVD [118], task-driven dictionary learning [117], Fisher discrimination dictionary learning [119], and label-consistent KSVD (LC-KSVD) [120, 121]. However, all these approaches are not robust to time-shifts or general deformations as it uses the Euclidean distance. In order to overcome this issue, in [122] a family of Gaussian elastic matching kernels was introduced. They use DTW, ERP, and TWED distances to compute the Gaussian kernel

$$K(\mathbf{x}^{(1)},\mathbf{x}^{(2)}) = \exp -\frac{\|\mathbf{x}^{(1)}-\mathbf{x}^{(2)}\|^2}{2}$$

However, the Gaussian elastic matching kernel cannot be guaranteed to be a positive definite symmetric (PDS) kernel. Thus, proper modifications have to be applied in order to remove the non-PDS part. Although the attempts to embed DTW distance to dictionary learning, several issues such as nonpositive semidefiniteness of "DTW Gaussian kernel" can compromise the robustness of results. Moreover, these approach is computationally expensive.

Recently, in view of these considerations, another sparse representation approach of a dictionary has been considered. It relies on the well-known notion of *centroid* for clustering algorithms. Each centroid is considered as the class representative and thus, if the classification problem involves M classes, M representative time series will be selected. However, DTW does not induce a proper definition of mean and thus the literature attempted to find a definition which is consistent with DTW. The most promising definition was given by Petitjean et al. [123, 124] and it was called DTW Barycenter averaging (DBA). Roughly speaking, it is based on an expectation-maximization scheme and multiple sequence alignment (commonly used in computational biology). This method is very effective as it allows to apply NR reducing N and DTW approximation to speed-up the single comparison. The difference between this approach and dictionary learning is that the "centroid" could not appertain to the dataset. Moreover, this method is supervised in some sense as it exploits the class label information by evaluating the centroid for each class.

4.2.4 Parametric Distance-Based Methods

Parametric distance-based methods compute the distance onto a reduced parametric representation of the signal. In this case, each time-series $\mathbf{z}$ is represented by a representation $\mathbf{x} \in \mathbb{R}^p$.

The most common procedure for the training phase is as follows:

- Find a parametric representation of all the training time series. The most used techniques are DWT, DFT stopped at a given coefficient order.
- Find a "centroid" or more generally a representative (template) of each class.

Once obtained the templates for each class, then the classification task is just given by an 1-NN classifier on the representatives $\mathbf{x}^{(\bullet)}$.

These simple methods suffer from various issues as they do not care of the intrinsic "dynamics" information of the signal. This issue has been tackled, mainly, by the computer vision literature [125, 126] and by Cuturi and Doucet [127] and Chen et al. [128] in a general context. Bissacco et al. [126] proposes family of kernels for dynamical systems based on the Binet-Cauchy kernel [129] for recognizing dynamic textures. Bissacco et al. [126] extends the work of Vishwanathan et al. [125] considering phase information, inputs or initial conditions. Essentially, these two works rely on a probabilistic modeling of the time series to define a kernel: in order to compare two time series, first, the dynamic behavior of each time series is learned by learning the parameters of a given state space dynamical systems, and then, the kernel is defined as a kernel between these two sets of parameters. In other words, the distance is computed over the parameters of the fitted dynamical systems. We will see later that a similar approach is followed by other classification methods. The work of Cuturi and Doucet [127] introduced the *Autoregressive Kernels* that are based on the vector autoregressive model (VAR): every multivariate (q-dimensional) time-series $\mathbf{z} \in \mathbb{R}^{q \times n}$ is represented by the feature $L(\theta;\mathbf{z}) = p_\theta(\mathbf{z})$, which is the likelihood function (it is a function of θ for a fixed sample $\mathbf{z}$), modeled by a VAR model. Given a measurable space $\mathcal{X}$ and a model (i.e., a parameterized family of distribution on $\mathcal{X}$ of the form $\{p_\theta,\ \theta \in \Theta\}$), the kernel K of two time-series $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$ is defined by

$$K(\mathbf{z}^{(1)}, \mathbf{z}^{(2)}) = \int_{\theta \in \Theta} p_\theta(\mathbf{z}^{(1)}) p_\theta(\mathbf{z}^{(2)}) \omega(d\theta)$$

where, in this case, $\omega(d\theta)$ is the matrix-normal inverse-Wishart prior. Moreover, Cuturi and Doucet [127] have shown that this kernel can be easily computed even when $q \gg n$ due to the fact that it does not resort to the actual estimation of a density. Indeed, all the kernels defined in [125–127] rely on a probabilistic parametric modeling of time series, but the computation of the Autoregressive Kernel avoids the two-step approach presented previously. Finally, Chen et al. [128] presents a model-metric colearning (MMCL) methodology, which differently from the works on [125–127], present a kernel based on nonlinear dynamical systems, named echo state networks (ESN) [130, 131]. For each time series, an ESN-model is trained and the model parameters θ are used to compute an opportune distance, also using kernel methods. For other recent application of ESN and its extension using liquid state machines (LSM), see [132–134]. We can notice that all the methods presented in this section can be seen as feature-based models. Indeed, in [127] the profile likelihood and in [125, 126, 128, 133, 134] the parameters of the dynamical systems can be seen as features.

Although accounting the "dynamics" information using the kernel methods may lead to superior accuracy with respect to the simple parametric distance-based methods, they are more computationally expensive (as they requires the feature extraction phase and the computation of the kernel). This fact would favor the usage of simpler methods which avoid the computation of the kernel. In the context of power systems, simple distance-based methods could be sufficient to capture the difference of signals, to identify, for example, anomalies. Finally, in Table 4 a summary of the most important characteristics of DB methods is presented.

Table 4 Main characteristics of distance-based methods that have been reviewed in this work

Methods		Characteristics
Purely DB	DTW	Invariant to time-warpings Complexity is $O(n^2)$
	DTW approximations	Complexity is $O(n)$
	DDTW	DTW on the first-derivative signal Reduces pathological alignments
	WDTW	Filtering with logistic weight function Favor matching points located in a neighborhood Reduces pathological alignments
	ERP	Supports local time shifting Metric distance function
	TWED	Elastic distance measure Edit distance + L_p norm
	Alignment kernels	DTW-based alignment kernel
	DTWUDC	DTW + dynamic constraints
Red-DB	SAX	Reduce into symbolic space
	1-NN-SAX	SAX representation + SAX distance Suitable for simple signals α and p must be tuned Prediction is $O(N)$
	1-NN-SAX with k templates	Prediction is $O(\widetilde{M})$
	1-NN-SFA	Fourier representation More adherence to the shape w.r.t. SAX Finer resolution with total recomputation
	Shapelets	Shift-invariant templates Training is $O(N^2n^3)$
	Dictionary learning	Learn a sparse representation In general use Euclidean distance Chen et al. [122] add kernel representation

Continued

Table 4 Main characteristics of distance-based methods that have been reviewed in this work *Continued*

	Methods	Characteristics
	DBA	DTW centroids are defined and used as templates
Par-DB	1-NN-DWT	Better than DFT to handle nonstationarity Good frequency and temporal resolution
	Binet-Cauchy kernels	Kernel to embed dynamic behavior Rely on a probabilistic parametric modeling of time series
	Autoregressive kernels	Easily computed even when $q \gg n$ Rely on a probabilistic parametric modeling of time series AR model
	MMCL	Kernel based on nonlinear dynamical systems (ESN)
	LSM	Distance on LSM parameters

4.3 Methods Comparison

In Table 5 we summarize the general characteristics of FB and DB methods.

Moreover, in Table 6 we present a general overview of the complexity of the methods reviewed in this work and we provide an indication of whether they

Table 5 Main Characteristics of Feature and Distance-based Time-Series Classification Approaches

Class of Methods	Characteristics
	Feature extraction can be onerous
	Difficult to define a "distance" between classes
FB	Easy interpretation of results
Purely DB	Complexity grows at least linearly with n
	Kernel-based methods can be onerous
	Distance has to be chosen tailored to the problem
	Comparisons with all N training examples (best case: $O(Nn)$)
	Distance computed in the reduce space
Red-DB	Reduce numerosity of time series
Par-DB	Distance is computed over the space of parameters
	Kernel-based methods can be onerous

Table 6 Categorization of time-series methods

	Methods	Complexity	Type
FB	Time-pattern features	$O(n)$, low if p and n low	Remote
	Interval features	$O(n) - O(n^2)$, low if p and n low	Remote
	Bag of features	High ($O(Nn) - O(n)$)	Centralized
	Dynamic features	High, low with simple AR models	Remote
	Frequency domain features	Low	Remote
Purely DB	DTW	High ($O(Nn^2)$)	Centralized
	DTW approximations	High (lower bound $O(Nn)$)	Centralized
	DDTW	High ($O(Nn^2)$)	Centralized
	WDTW	High ($O(Nn^2)$)	Centralized
	ERP	High ($O(Nn^2)$)	Centralized
	TWED	High ($O(Nn^2)$)	Centralized
	Alignment kernels	High	Centralized
Red-DB	DTWUDC	High (DTW + dynamic model)	Centralized
	1-NN-SAX	High ($O(Nn)$)	Centralized
	1-NN-SAX with k templates	Low ($O(kn)$)	Remote
	1-NN-SFA	High $O(Nn)$	Centralized (RWT)
	Shapelets	Low (reduction methods)	Remote
	Dictionary learning	$O(k)$ (cardinality of dictionary)	Remote
	DBA	Low ($O(n^2) - O(n)$), DTW approx	Remote
Par-DB	1-NN-DWT	High, low *with* templates	Centralized (RWT)
	Binet-Cauchy kernels	High	Centralized
	Autoregressive kernels	High ($O(N(n^2p^3))$), p: AR order	Centralized
	MMCL	High	Centralized
	LSM	High	Centralized

Notes: *Methods as DTW which admits variable-length timeseries has complexity which depends on the lengths (*n *and* m *of two timeseries). However, for the sake of simplicity with consider only* n *assuming that* m *is very similar to* n *(very common in real scenarios).* RWT, *remote with template.*

are suitable for *remote* or *centralized* tasks. With "remote" we refer to classification tasks than can be delivered without any kind of information coming from other nodes or locations, while with "centralized" we refer to algorithms or framework that can be executed in a central system that is aware of all nodes data. Notice that in doing the distinction remote/centralized, we assume that the training phase is carried out offline and thus in Table 6 only classification complexity is concerned.

Table 7 Categorization of time-series classification methods

	Methods	References
FB	Time-pattern features	[60–62]
	Interval features	[63–66]
	Bag of features	[68–76]
	Dynamic features	[82–84]
	Frequency domain features	[85–87]
	Ensemble of features	[89]
Purely DB	Miscellaneous	[90, 92, 93]
	DTW	[91]
	DTW approximations	[94–98]
	DDTW	[99]
	WDTW	[100]
	ERP	[101, 102]
	TWED	[103]
	Alignment kernels	[105, 106]
	DTWUDC	[107]
Red-DB	1-NN-SAX	[47, 48]
	1-NN-SAX with templates	[135]
	1-NN-SFA	[50]
	Shapelets	[108–114]
	Dictionary learning	[115–117]
	DBA	[123, 124]
Par-DB	1-NN-DWT	[136]
	Binet-Cauchy kernels	[125, 126, 129]
	Autoregressive kernels	[105, 127]
	MMCL	[128]
	LSM	[132–134]

Finally in Table 7, we summarize the references that correspond to each group of methods.

5 APPLICATIONS

In Table 8 we provide a list of some power-system applications whose issues have been addressed adopting some of the methodologies presented in this chapter. As can be seen, most of the applications concern event, anomaly and FD problems, exploiting FB techniques, which appear of an immediate application.

Table 8 Power-system applications

Year	Refs.	Method	Methodology	Data	Application
2016	[2]	Kernel PCA Partial SVM	FB	PMU	FDI
2016	[23]	mRmR Ensemble of bundle classifier SVM	FB	PMU	FDI
2016	[137]	DFT, DWT, FDST, PCA, Shapelet 1-NN, SVM	FB	PMU	FDI
2016	[13]	DTW, MDTW	Purely DB	AMI AM	FD
2016	[138]	Decision tree, SVM	FB	AMI	TD
2016	[139]	DWT (for preprocessing) Gaussian mixture models Parzen density estimator *k*-means clustering, *k*-NN Standard SVDD SVDD with negative examples	FB	PMU	ND
2016	[140]	Semisupervised SVM Adaboost, Multiple kernel learning	FB	PMU	AD
2015	[141]	Wavelet	Par-DB	PMU	AD/ED
2015	[142]	DWT, neural networks	FB	PMU	Event/FDI
2015	[143]	Adaptive neuro-fuzzy inference system Neural networks, SVM	FB	PMU	FD
2015	[144]	One class classifier	DB	Smart sensors	FDI
2014	[12]	Hidden Markov Models (HMM)	FB	PMU	FD
2014	[145]	Clustering, outlier detection Recursive feature elimination Multiple linear regression, ARIMA Support vector regression, *k*-NN RF, Boosting tree MARS Ensemble of methods	FB	Meteo	Prediction
2014	[146]	Naive Bayes Rule induction (OneR, NNge, JRipper) Decision tree learning (RFs) Binary classification (SVM) Boosting (Adaboost)	FB	PMU	FD AD
2013	[147]	Iterative Hilbert Huang Transform SAX	Red-DB	PMU	Monitoring

Continued

Table 8 Power-system applications *Continued*

Year	Refs.	Method	Methodology	Data	Application
2012	[148]	SVM One class SVM (semisupervised)	FB	PMU	FDI
2009	[17]	LS-SVM, DWT	FB	PMU	FDI
2008	[149]	DWT (for feature extraction)	Par-DB	PMU	FD
2006	[16]	SVM, neural networks	FB	PMU	FDI
2006	[22]	LR, neural networks	FB	Fault logs Meteo	FDI
2001	[21]	DWT	Par-DB	PMU	FDI

Notes: *A list of applications in the field of power systems is given, together with the main adopted time-series classification techniques. In "Application" column, the acronyms AD, TD, ND stand for anomaly, theft, and novelty detections.*

6 CONCLUDING REMARKS

Surfing through these methodologies applied to the power systems time series highlights the synergy between big data analysis and cyber-physical systems within the *smart paradigm*. These methodologies have paved a golden way toward new frontiers in scientific innovation and quality of service, leveraging continuous technological advances. Nonetheless, still some issues remains to be solved and to gain a main role in industrial and academic research.

A first fundamental aspect regard the control and architecture at all scales, meaning that through the pervasive monitoring of the systems the aim is to reach a full knowledge of the whole power system pipeline (from the first energy transformation to the final user) and at all levels (from the wide area grid network to the residential installations): on the one side this means even higher volumes and complexity of the data, while on the other it calls for the interoperability of the systems, the buzzword in the world of Internet of Things.

Also, it appears how the information and communication technologies, the cyber part, have gained a predominant role with respect to the physical counterpart, leveraging big data so as to offer new services and an increased performance of the systems: conversely, this ICT-mediated technology opens new scenarios in the field of faults, malicious behaviors, and more in general *cybersecurity* maintenance [150]. Concurrently, the governance of data and the privacy concerns need to be taken into account in this context, so as to guarantee the accurate and continuous knowledge of plants and behaviors without being invasive through the definition of policies of information management [151].

References

[1] B.E. Boser, I.M. Guyon, V.N. Vapnik, A training algorithm for optimal margin classifiers, in: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, ACM, 1992, pp. 144–152.

[2] Y. Zhou, R. Arghandeh, I. Konstantakopoulos, S. Abdullah, A. von Meier, C.J. Spanos, Abnormal event detection with high resolution micro-PMU data, in: Power Systems Computation Conference (PSCC), 2016, pp. 1–7.

[3] J. Valenzuela, J. Wang, N. Bissinger, Real-time intrusion detection in power system operations, IEEE Trans. Power Syst. 28 (2) (2013) 1052–1062.

[4] A. Guerini, G. De Nicolao, Long-term electric load forecasting: a torus-based approach, in: 2015 European Control Conference (ECC), IEEE, 2015, pp. 2768–2773.

[5] F. Javed, N. Arshad, F. Wallin, I. Vassileva, E. Dahlquist, Forecasting for demand response in smart grids: an analysis on use of anthropologic and structural data and short term multiple loads forecasting, Appl. Energy 96 (2012) 150–160.

[6] P. Siano, Demand response and smart grids—a survey, Renew. Sust. Energ. Rev. 30 (2014) 461–478.

[7] K. Fischer, T. Stalin, H. Ramberg, J. Wenske, G. Wetter, R. Karlsson, T. Thiringer, Field-experience based root-cause analysis of power-converter failure in wind turbines, IEEE Trans. Power Electron. 30 (5) (2015) 2481–2492.

[8] D. Karlsson, M. Hemmingsson, S. Lindahl, Wide area system monitoring and control—terminology, phenomena, and solution implementation strategies, IEEE Power Energy Mag. 2 (5) (2004) 68–76.

[9] S. Yang, D. Xiang, A. Bryant, P. Mawby, L. Ran, P. Tavner, Condition monitoring for device reliability in power electronic converters: a review, IEEE Trans. Power Electron. 25 (11) (2010) 2734–2752.

[10] A.G. Phadke, P. Wall, L. Ding, V. Terzija, Improving the performance of power system protection using wide area monitoring systems, J. Mod. Power Syst. Clean Energy 4 (3) (2016) 319–331.

[11] A.G. Phadke, J.S. Thorp, Synchronized Phasor Measurements and Their Applications, Springer US, 2008. ISBN: 9780387765372.

[12] H. Jiang, J.J. Zhang, W. Gao, Z. Wu, Fault detection, identification, and location in smart grid based on data-driven computational methods, IEEE Trans. Smart Grid 5 (6) (2014) 2947–2956.

[13] N. Zhou, J. Wang, Q. Wang, A novel estimation method of metering errors of electric energy based on membership cloud and dynamic time warping, IEEE Trans. Smart Grid 8 (3) (2016) 1318–1329.

[14] N. Yu, S. Shah, R. Johnson, R. Sherick, M. Hong, K. Loparo, Big data analytics in power distribution systems, in: 2015 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT), 2015, pp. 1–5, https://doi.org/10.1109/ISGT.2015.7131868.

[15] M. Manic, D. Wijayasekara, K. Amarasinghe, J.J. Rodriguez-Andina, Building energy management systems: the age of intelligent and adaptive buildings, IEEE Ind. Electron. Mag. 10 (1) (2016) 25–39.

[16] P. Janik, T. Lobos, Automated classification of power-quality disturbances using SVM and RBF networks, IEEE Trans. Power Delivery 21 (3) (2006) 1663–1669.

[17] Q.-M. Zhang, H.-J. Liu, Application of LS-SVM in classification of power quality disturbances, Proc. Chinese Soc. Electr. Eng. 28 (1) (2008) 106.

[18] L. Rabiner, B.-H. Juang, Fundamentals of Speech Recognition, Prentice Hall, Upper Saddle River, NJ, 1993.

[19] R.S. Tsay, Analysis of Financial Time Series, vol. 543, John Wiley & Sons, London, 2005.

[20] G.A. Susto, A. Beghi, Dealing with time-series data in predictive maintenance problems, in: 2016 IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2016, pp. 1–4.

[21] O.A.S. Youssef, Fault classification based on wavelet transforms, in: 2001 IEEE/PES Transmission and Distribution Conference and Exposition, vol. 1, IEEE, 2001, pp. 531–536.

[22] L. Xu, M.-Y. Chow, A classification approach for power distribution systems fault cause identification, IEEE Trans. Power Syst. 21 (1) (2006) 53–60.

[23] Y. Zhou, R. Arghandeh, I.C. Konstantakopoulos, S. Abdullah, A. von Meier, C. J. Spanos, Distribution Network Event Detection with Ensembles of Bundle Classifiers, in: IEEE PES General Meeting 2016, 2016.

[24] G.A. Susto, A. Schirru, S. Pampuri, D. Pagano, S. McLoone, A. Beghi, A predictive maintenance system for integral type faults based on support vector machines: an application to ion implantation, in: 2013 IEEE International Conference on Automation Science and Engineering (CASE), IEEE, 2013, pp. 195–200.

[25] A. Beghi, L. Cecchinato, C. Corazzol, M. Rampazzo, F. Simmini, G.A. Susto, A one-class SVM based tool for machine learning novelty detection in HVAC chiller systems, IFAC Proc. 47 (3) (2014) 1953–1958.

[26] J. Friedman, T. Hastie, R. Tibshirani, The Elements of Statistical Learning, Springer Series in Statistics, vol. 1, Springer, Berlin, 2009.

[27] M. Müller, Dynamic time warping, Inf. Retr. Music Motion 2007, pp. 69–84.

[28] A. Schirru, G.A. Susto, S. Pampuri, S. McLoone, Learning from time series: supervised aggregative feature extraction, in: 2012 IEEE 51st Annual Conference on Decision and Control (CDC), IEEE, 2012, pp. 5254–5259.

[29] L. Xie, Y. Chen, P.R. Kumar, Dimensionality reduction of synchrophasor data for early event detection: linearized analysis, IEEE Trans. Power Syst. 29 (6) (2014) 2784–2794.

[30] L. Van Der Maaten, E. Postma, J. Van den Herik, Dimensionality reduction: a comparative, J. Mach. Learn. Res. 10 (2009) 66–71.

[31] T.-C. Fu, A review on time series data mining, Eng. Appl. Artif. Intel. 24 (1) (2011) 164–181.

[32] B. Scholkopft, K.-R. Mullert, Fisher discriminant analysis with kernels, in: Neural Networks for Signal Processing IX, vol. 1, 1999, p. 1.

[33] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis, vol. 46, John Wiley & Sons, New York, 2004.

[34] A. Subasi, M.I. Gursoy, EEG signal classification using PCA, ICA, LDA and support vector machines, Expert Syst. Appl. 37 (12) (2010) 8659–8666.

[35] B. Schölkopf, A. Smola, K.-R. Müller, Kernel principal component analysis, in: International Conference on Artificial Neural Networks, Springer, 1997, pp. 583–588.

[36] S. Mika, B. Schölkopf, A.J. Smola, K.-R. Müller, M. Scholz, G. Rätsch, Kernel PCA and De-Noising in Feature Spaces, in: NIPS, vol. 11, 1998, pp. 536–542.

[37] G. McLachlan, Discriminant analysis and statistical pattern recognition, 544, John Wiley & Sons, New York, 2004.

[38] J.B. Kruskal, M. Wish, Multidimensional Scaling, vol. 11, SAGE, Thousand Oaks, CA, 1978.

[39] H. Hotelling, Analysis of a complex of statistical variables into principal components, J. Educ. Psychol. 24 (6) (1933) 417.

[40] I. Jolliffe, Principal Component Analysis, Wiley Online Library, New York, 2002.

[41] J.B. Tenenbaum, V. De Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.

[42] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (5500) (2000) 2323–2326.

[43] L. Cayton, Algorithms for manifold learning, Univ. of California at San Diego Tech. Rep. (2005) 1–17.

[44] H. Narayanan, S. Mitter, Sample complexity of testing the manifold hypothesis, in: Adv. Neural Inf. Process. Syst., 2010, pp. 1786–1794.

[45] G. Biau, Analysis of a random forests model, J. Mach. Learn. Res. 13 (Apr) (2012) 1063–1095.

[46] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: a review, in: Data Classification: Algorithms and Applications, CRC Press, Boca Raton, FL, 2014, p. 37.

[47] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, in: Proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, ACM, 2003, pp. 2–11.

[48] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing SAX: a novel symbolic representation of time series, Data Min. Knowl. Disc. 15 (2) (2007) 107–144.

[49] E. Keogh, K. Chakrabarti, M. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, Knowl. Inf. Syst. 3 (3) (2001) 263–286.

[50] P. Schäfer, M. Högqvist, SFA: a symbolic Fourier approximation and index for similarity search in high dimensional datasets, in: Proceedings of the 15th International Conference on Extending Database Technology, ACM, 2012, pp. 516–527.

[51] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[52] M.E. Tipping, Sparse Bayesian learning and the relevance vector machine, J. Mach. Learn. Res. 1 (Jun) (2001) 211–244.

[53] L. Breiman, J. Friedman, C.J. Stone, R.A. Olshen, Classification and Regression Trees, CRC Press, Boca Raton, FL, 1984.

[54] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[55] D.R. Cox, The regression analysis of binary sequences, J. R. Stat. Soc. Ser. B Methodol. 20 (1958) 215–242.

[56] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes for Machine Learning, vol. 1, MIT Press, Cambridge, 2006.

[57] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016.

[58] C.M. Bishop, Pattern Recognition, Mach. Learn. 128 (2006) 1–58.

[59] S. Manganaris, Supervised Classification With Temporal Data, Vanderbilt University, Nashville, TN, 1997.

[60] M.W. Kadous, Learning comprehensible descriptions of multivariate time series, in: ICML, 1999, pp. 454–463.

[61] M. Kudo, J. Toyama, M. Shimbo, Multidimensional curve classification using passing-through regions, Pattern Recogn. Lett. 20 (11) (1999) 1103–1111.

[62] P. Geurts, Pattern extraction for time series classification, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2001, pp. 115–127.

[63] J.J. Rodríguez, C.J. Alonso, H. Boström, Boosting interval based literals, Intell. Data Anal. 5 (3) (2001) 245–262.

[64] J.J. Rodríguez, C.J. Alonso, Interval and dynamic time warping-based decision trees, in: Proceedings of the 2004 ACM Symposium on Applied Computing, ACM, 2004, pp. 548–552.

[65] J.J. Rodríguez, C.J. Alonso, J.A. Maestro, Support vector machines of interval-based features for time series classification, Knowl.-Based Syst. 18 (4) (2005) 171–178.

[66] H. Deng, G. Runger, E. Tuv, M. Vladimir, A time series forest for classification and feature extraction, Inf. Sci. 239 (2013) 142–153.

[67] D.G. Lowe, Object recognition from local scale-invariant features, in: The proceedings of the Seventh IEEE International Conference on Computer vision, vol. 2, IEEE, 1999, pp. 1150–1157.

[68] J. Lin, R. Khade, Y. Li, Rotation-invariant similarity in time series using bag-of-patterns representation, J. Intell. Inf. Syst. 39 (2) (2012) 287–315.

[69] J. Wang, P. Liu, M.F.H. She, S. Nahavandi, A. Kouzani, Bag-of-words representation for biomedical time series classification, Biomed. Signal Process. Control 8 (6) (2013) 634–644.

[70] M.G. Baydogan, G. Runger, E. Tuv, A bag-of-features framework to classify time series, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2796–2802.

[71] P. Senin, S. Malinchik, SAX-VSM: interpretable time series classification using SAX and vector space model, in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013, pp. 1175–1180.

[72] M.G. Baydogan, G. Runger, Learning a symbolic representation for multivariate time series classification, Data Min. Knowl. Disc. 29 (2) (2015) 400–422.

[73] A. Bailly, S. Malinowski, R. Tavenard, T. Guyet, L. Chapel, Bag-of-temporal-SIFT-Words for time series classification, in: ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data, 2015.

[74] P. Schäfer, The BOSS is concerned with time series classification in the presence of noise, Data Min. Knowl. Disc. 29 (6) (2015) 1505–1530.

[75] P. Schäfer, Scalable time series classification, Data Min. Knowl. Disc. volume 30 (2016) 1273–1298.

[76] A. Bailly, S. Malinowski, R. Tavenard, L. Chapel, T. Guyet, Dense bag-of-temporal-SIFT-words for time series classification, in: International Workshop on Advanced Analytics and Learning on Temporal Data, Springer International Publishing, September, 2015, pp. 17–30.

[77] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: ECCV Workshop on Statistical Learning in Computer Vision, vol. 1, Prague, 2004, pp. 1–2.

[78] G. Salton, A. Wong, C.-S. Yang, A vector space model for automatic indexing, Commun. ACM 18 (11) (1975) 613–620.

[79] H.P. Luhn, A statistical approach to mechanized encoding and searching of literary information, IBM J. Res. Dev. 1 (4) (1957) 309–317.

[80] K.S. Jones, A statistical interpretation of term specificity and its application in retrieval, J. Doc. 28 (1) (1972) 11–21.

[81] L. Ljung, System identification, in: Signal Analysis and Prediction, Springer, New York, 1998, pp. 163–173.

[82] D. Garrett, D.A. Peterson, C.W. Anderson, M.H. Thaut, Comparison of linear, nonlinear, and feature selection methods for EEG signal classification, IEEE Trans. Neural Syst. Rehabil. Eng. 11 (2) (2003) 141–144.

[83] Z.-Y. He, L.-W. Jin, Activity recognition from acceleration data using AR model representation and SVM, in: 2008 International Conference on Machine Learning and Cybernetics, vol. 4, IEEE, 2008, pp. 2245–2250.

[84] P.S.S. Roberts, Bayesian time series classification, Adv. Neural Inf. Process. Syst. 14 (2002) 937.

[85] P. Jahankhani, V. Kodogiannis, K. Revett, EEG signal classification using wavelet feature extraction and neural networks, in: IEEE John Vincent Atanasoff 2006 International Symposium on Modern Computing (JVA'06), IEEE, 2006, pp. 120–124.

[86] A. Subasi, EEG signal classification using wavelet feature extraction and a mixture of expert model, Expert Syst. Appl. 32 (4) (2007) 1084–1093.

[87] E.D. Übeyli, Combined neural network model employing wavelet coefficients for EEG signals classification, Digital Signal Process. 19 (2) (2009) 297–308.

[88] N.E. Huang, Z. Shen, S.R. Long, M.C. Wu, H.H. Shih, Q. Zheng, N.-C. Yen, C.C. Tung, H. H. Liu, The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, in: Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, vol. 454, The Royal Society, 1998, pp. 903–995.

[89] V. Eruhimov, V. Martyanov, E. Tuv, Constructing high dimensional feature space for time series classification, in: European Conference on Principles of Data Mining and Knowledge Discovery, Springer, 2007, pp. 414–421.

[90] G.E. Batista, X. Wang, E.J. Keogh, A complexity-invariant distance measure for time series, in: SDM, vol. 11, SIAM, 2011, pp. 699–710.

[91] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Trans. Acoust. Speech Signal Process. 26 (1) (1978) 43–49.

[92] X. Xi, E. Keogh, C. Shelton, L. Wei, C.A. Ratanamahatana, Fast time series classification using numerosity reduction, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 1033–1040.

[93] J. Lines, A. Bagnall, Time series classification with ensembles of elastic distance measures, Data Min. Knowl. Disc. 29 (3) (2015) 565–592.

[94] E. Keogh, C.A. Ratanamahatana, Exact indexing of dynamic time warping, Knowl. Inf. Syst. 7 (3) (2005) 358–386.

[95] S. Salvador, P. Chan, Toward accurate dynamic time warping in linear time and space, Intell. Data Anal. 11 (5) (2007) 561–580.

[96] G. Al-Naymat, S. Chawla, J. Taheri, SparseDTW: a novel approach to speed up dynamic time warping, in: Proceedings of the Eighth Australasian Data Mining Conference, vol. 101, Australian Computer Society, Inc., 2009, pp. 117–127.

[97] K.S. Candan, R. Rossini, X. Wang, M.L. Sapino, sDTW: computing DTW distances using locally relevant constraints based on salient feature alignments, Proc. VLDB Endowment 5 (11) (2012) 1519–1530.

[98] D.F. Silva, G.E. Batista, Speeding up all-pairwise dynamic time warping matrix calculation, in: Proceedings of the 2016 SIAM International Conference on Data Mining, SIAM, 2016, pp. 837–845.

[99] E.J. Keogh, M.J. Pazzani, Derivative dynamic time warping, in: SDM, vol. 1, SIAM, 2001, pp. 5–7.

[100] Y.-S. Jeong, M.K. Jeong, O.A. Omitaomu, Weighted dynamic time warping for time series classification, Pattern Recogn. 44 (9) (2011) 2231–2240.

[101] L. Chen, R. Ng, On the marriage of LP-norms and edit distance, in: Proceedings of the Thirtieth International Conference on Very Large Data Bases, vol. 30, VLDB Endowment, 2004, pp. 792–803.

[102] L. Chen, M.T. Özsu, V. Oria, Robust and fast similarity search for moving object trajectories, in: Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, ACM, 2005, pp. 491–502.

[103] P.-F. Marteau, Time warp edit distance with stiffness adjustment for time series matching, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 306–318.

[104] B. Scholkopf, A.J. Smola, Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT Press, Cambridge, MA, 2001.

[105] M. Cuturi, J.-P. Vert, O. Birkenes, T. Matsui, A kernel for time series based on global alignments, in: 2007 IEEE International Conference on Acoustics, Speech and Signal Processing—ICASSP'07, vol. 2, IEEE, 2007, pp. 413.

[106] M. Cuturi, Fast global alignment kernels, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 929–936.

[107] S. Soatto, On the distance between non-stationary time series, in: Modeling, Estimation and Control, Springer, 2007, pp. 285–299.

[108] L. Ye, E. Keogh, Time series shapelets: a new primitive for data mining, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 947–956.

[109] L. Ye, E. Keogh, Time series shapelets: a novel technique that allows accurate, interpretable and fast classification, Data Min. Knowl. Disc. 22 (1–2) (2011) 149–182.

[110] A. Mueen, E. Keogh, N. Young, Logical-shapelets: an expressive primitive for time series classification, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2011, pp. 1154–1162.

[111] J. Lines, L.M. Davis, J. Hills, A. Bagnall, A shapelet transform for time series classification, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 289–297.

[112] T. Rakthanmanon, E. Keogh, Fast shapelets: a scalable algorithm for discovering time series shapelets, in: Proceedings of the 13th SIAM International Conference on Data Mining, SIAM, 2013, pp. 668–676.

[113] J. Hills, J. Lines, E. Baranauskas, J. Mapp, A. Bagnall, Classification of time series by shapelet transformation, Data Min. Knowl. Disc. 28 (4) (2014) 851–881.

[114] J. Grabocka, N. Schilling, M. Wistuba, L. Schmidt-Thieme, Learning time-series shapelets, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 392–401.

[115] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Discriminative learned dictionaries for local image analysis, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008, IEEE, 2008, pp. 1–8.

[116] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, F.R. Bach, Supervised dictionary learning, in: Adv. Neural Inf. Process. Syst., 2009, pp. 1033–1040.

[117] J. Mairal, F. Bach, J. Ponce, Task-driven dictionary learning, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2012) 791–804.

[118] Q. Zhang, B. Li, Discriminative K-SVD for dictionary learning in face recognition, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2010, pp. 2691–2698.

[119] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: 2011 IEEE International Conference on Computer Vision (ICCV), IEEE, 2011, pp. 543–550.

[120] Z. Jiang, Z. Lin, L.S. Davis, Learning a discriminative dictionary for sparse coding via label consistent K-SVD, in: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2011, pp. 1697–1704.

[121] Z. Jiang, Z. Lin, L.S. Davis, Label consistent K-SVD: learning a discriminative dictionary for recognition, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2651–2664.

[122] Z. Chen, W. Zuo, Q. Hu, L. Lin, Kernel sparse representation for time series classification, Inf. Sci. 292 (2015) 15–26.

[123] F. Petitjean, A. Ketterlin, P. Gançarski, A global averaging method for dynamic time warping, with applications to clustering, Pattern Recogn. 44 (3) (2011) 678–693.

[124] F. Petitjean, G. Forestier, G.I. Webb, A.E. Nicholson, Y. Chen, E. Keogh, Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm, Knowl. Inf. Syst. 47 (1) (2016) 1–26.

[125] S.V.N. Vishwanathan, A.J. Smola, R. Vidal, Binet-Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes, Int. J. Comput. Vis. 73 (1) (2007) 95–119.

[126] A. Bissacco, A. Chiuso, S. Soatto, Classification and recognition of dynamical models: the role of phase, independent components, kernels and optimal transport, IEEE Trans. Pattern Anal. Mach. Intell. 29 (11) (2007) 1958–1972.

[127] M. Cuturi, A. Doucet, Autoregressive kernels for time series, 2011 (arXiv preprint arXiv:1101.0673).

[128] H. Chen, F. Tang, P. Tino, A.G. Cohn, X. Yao, Model metric co-learning for time series classification, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, AAAI Press, 2015, pp. 3387–3394.

[129] S.V.N. Vishwanathan, A.J. Smola, et al., Binet-Cauchy kernels, in: NIPS, 2004, pp. 1441–1448.

[130] H. Jaeger, The "echo state" approach to analysing and training recurrent neural networks-with an erratum note, vol. 148, German National Research Center for Information Technology GMD Technical Report, Bonn, Germany, 2001, p.34.

[131] H. Jaeger, Adaptive nonlinear system identification with echo state networks, in: Adv. Neural Inf. Process. Syst., 2002, pp. 593–600.

[132] W. Aswolinskiy, R.F. Reinhart, J. Steil, Time series classification in reservoir-and model-space: a comparison, in: IAPR Workshop on Artificial Neural Networks in Pattern Recognition, Springer, 2016, pp. 197–208.

[133] Q. Ma, L. Shen, W. Chen, J. Wang, J. Wei, Z. Yu, Functional echo state network for time series classification, Inf. Sci. 373 (2016) 1–20.

[134] Y. Li, J. Hong, H. Chen, Sequential data classification in the space of liquid state machines, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2016, pp. 313–328.

[135] P. Siirtola, H. Koskimäki, V. Huikari, P. Laurinen, J. Röning, Improving the classification accuracy of streaming data using SAX similarity features, Pattern Recogn. Lett. 32 (13) (2011) 1659–1668.

[136] P. Fryzlewicz, H. Ombao, Consistent classification of nonstationary time series using stochastic wavelet representations, J. Am. Stat. Assoc. 104 (2012) 299–312.

[137] S. Brahma, R. Kavasseri, H. Cao, N.R. Chaudhuri, T. Alexopoulos, Y. Cui, Real time identification of dynamic events in power systems using PMU data, and potential applications—models, promises, and challenges, IEEE Trans. Power Delivery 32 (2017) 294–301.

[138] A. Jindal, A. Dua, K. Kaur, M. Singh, N. Kumar, S. Mishra, Decision tree and SVM-based data analytics for theft detection in smart grid, IEEE Trans. Ind. Inf. 12 (3) (2016) 1005–1016.

[139] A.E. Lazzaretti, D.M.J. Tax, H.V. Neto, V.H. Ferreira, Novelty detection and multi-class classification in power distribution voltage waveforms, Expert Syst. Appl. 45 (2016) 322–330.

[140] M. Ozay, I. Esnaola, F.T.Y. Vural, S.R. Kulkarni, H.V. Poor, Machine learning methods for attack detection in the smart grid, IEEE Trans. Neural Networks Learn. Syst. 27 (8) (2016) 1773–1786.

[141] D.-I. Kim, T.Y. Chun, S.-H. Yoon, G. Lee, Y.-J. Shin, Wavelet-based event detection method using PMU data, IEEE Trans. Smart Grid 8 (2017) 1154–1162.

[142] S. Alshahrani, M. Abbod, B. Alamri, Detection and classification of power quality events based on wavelet transform and artificial neural networks for smart grids, in: Smart Grid (SASG), 2015 Saudi Arabia, IEEE, 2015, pp. 1–6.

[143] P. Gopakumar, J.B. Reddy, D.K. Mohanta, Adaptive fault identification and classification methodology for smart power grids using synchronous phasor angle measurements, IET Gener. Transm. Distrib. 9 (2) (2015) 133–145.

[144] E. De Santis, L. Livi, A. Sadeghian, A. Rizzi, Modeling and recognition of smart grid faults by a combined approach of dissimilarity learning and one-class classification, Neurocomputing 170 (2015) 368–383.

[145] C. Fan, F. Xiao, S. Wang, Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques, Appl. Energy 127 (2014) 1–10.

[146] R.C.B. Hink, J.M. Beaver, M.A. Buckner, T. Morris, U. Adhikari, S. Pan, Machine learning for power system disturbance and cyber-attack discrimination, in: 2014 7th International Symposium on Resilient Control Systems (ISRCS), IEEE, 2014, pp. 1–8.

[147] M.J. Afroni, D. Sutanto, D. Stirling, Analysis of nonstationary power-quality waveforms using iterative Hilbert Huang transform and SAX algorithm, IEEE Trans. Power Delivery 28 (4) (2013) 2134–2144.

[148] N. Shahid, S.A. Aleem, I.H. Naqvi, N. Zaffar, Support vector machine based fault detection & classification in smart grids, in: 2012 IEEE Globecom Workshops (GC Wkshps), IEEE, 2012, pp. 1526–1531.

[149] N.I. Elkalashy, M. Lehtonen, H.A. Darwish, A.-M.I. Taalab, M.A. Izzularab, DWT-based detection and transient power direction-based location of high-impedance faults due to leaning trees in unearthed MV networks, IEEE Trans. Power Delivery 23 (1) (2008) 94–101.

[150] L. Langer, F. Skopik, P. Smith, M. Kammerstetter, From old to new: assessing cybersecurity risks for an evolving smart grid, Comput. Secur. 62 (2016) 165–176.

[151] M. Buchmann, Governance of data and information management in smart distribution grids: increase efficiency by balancing coordination and competition, Util. Policy (2017), https://doi.org/10.1016/j.jup.2017.01.003.