ECHAD: embedding-based change detection from multivariate time series in smart grids

Journal:	*IEEE Access*
Manuscript ID	Draft
Manuscript Type:	Regular Manuscript
Date Submitted by the Author:	n/a
Complete List of Authors:	Ceci, Michelangelo; Universita degli Studi di Bari Aldo Moro, Dept. of Informatics Corizzo, Roberto; American University, Japkowicz, Nathalie; Univ Ottawa, ; American University Mignone, Paolo; University of Bari, Computer Science Pio, Gianvito; University of Bari, Computer Science
Keywords: Please choose keywords carefully as they help us find the most suitable Editor to review:	Change detection algorithms, Smart grids, Neural networks
Subject Category Please select at least two subject categories that best reflect the scope of your manuscript:	Computational and artificial intelligence, Science - general
Additional Manuscript Keywords:	one-class learning, embedding

Original Manuscript ID: Access-2020-33391

Original Article Title: "ECHAD: embedding-based change detection from multivariate time series in smart grids"

To: IEEE Access Editor

Re: Response to reviewers

Dear Editor,

Thank you for allowing a resubmission of our manuscript, with an opportunity to address the reviewers' comments.

We are uploading (a) our point-by-point response to the comments (below) (response to reviewers), (b) an updated manuscript with yellow highlighting indicating changes, and (c) a clean updated manuscript without highlights (PDF main document).

Best regards,

Michelangelo Ceci, Roberto Corizzo, Nathalie Japkowicz, Paolo Mignone and Gianvito Pio

Reviewer#1, Concern # 1:

The Topic of this paper is interesting. However, the main contribution of the paper should be Clearer, e.g. Compare the results with existing techniques, in terms of both accuracy/sensitivity and computation complexity. The Author should try to clearly demonstrate the contributions of their work and the effectiveness of the proposed method.

Author response: In Section 1 we have now clarified the contribution of the paper. As for the comparison with existing state-of-the-art techniques (namely, Isolation Forest, Local Outlier Factor and One Class SVM), it is already reported on the paper, in terms of Accuracy, Precision, Recall (a.k.a. sensitivity) and F1-Score in Figures 5-8 for synthetic datasets. As detailed on the paper, computing such measures on the real dataset is not feasible since the ground truth is not available. However, in Section IV.C and Figures 9-14, we carry out a qualitative analysis, emphasizing the capability of our method to avoid false positives and to predict clearly correct changes, not detected by competitor systems. We agree with the reviewer about the lack of a comparison in terms of running times in the previous version of the paper.

Author action: We updated the manuscript by clarifying the contribution of the paper In Section 1. Moreover, we reported a comparison between ECHAD and all the competitors also in terms of the running times (see Section IV.C). For the sake of completeness, we also reported some comments about the theoretical computational complexity of ECHAD at the end of Section III.

Reviewer#1, Concern # 2:

what is the meaning of this abbreviation ECHAD.

Author response: We thank the reviewer for pointing out that this information was missing on the paper. ECHAD stands for “Embedding-based CHAnge Detection”.

Author action: We updated the manuscript by reporting the meaning of the acronym ECHAD in the first occurrence in the Abstract (page 1) and in the Introduction (page 2).

Reviewer#1, Concern # 3:

What is the meaning of MPAP
Author response: We thank the reviewer for pointing out that this information was missing on the paper. The time series shown in Figure 1 are examples of typical features observed in a power grid. MPAP is actually the acronym of “Media Potenza Apparente Trifase” (in Italian), that corresponds to “Average Three-Phase Electric Power” (in English). The real-world dataset was provided by e-distribuzione S.p.A. (an Italian company), that reported the (abbreviated) name of such a variable in Italian.

Author action: We updated the manuscript by changing the name of the variable shown in Figure 1, reporting their full name in English.

Reviewer#1, Concern # 4:

There is not enough explanation about the obtain parameters which have been used in Figures (11-14). I would had expected a more thorough discussion here.

Author response: Figures 11-14 refer to the results obtained by the competitor systems (namely, Isolation Forest, Local Outlier Factor and One Class SVM). In the previous version of the paper, we reported that we adopted the default values suggested in their respective papers, but we agree with the reviewer about the need to explicitly report the parameter values used in our experiments to guarantee the self-consistency of the experimental evaluation.

Author action: We updated the manuscript by explicitly reporting the parameters used for competitor systems (see Section IV.B).

Reviewer#1, Concern # 5:

However, it is not clear how the ECHAD was trained to perform detection multivariate time series in smart grids.

Author response: The method learned by ECHAD is actually represented by $\overline{D_i}$ and σD_i that are the mean and the standard deviation, respectively, of the Euclidean distance between an instance and its p nearest neighbors at time *i*, in the reduced K-dimensional feature space obtained through an embedding method (see Figure 2). Indeed, σD_i is used to compute the decision threshold T_i (see Equation 6) that, together with the value of $\overline{D_i}$, is used to decide whether a new observation can be classified as a change or not, according to Equation 7. Note that the initial feature space is M-dimensional, where each feature represents a time series. Therefore, an input training matrix $W_i \in R^{N_i \times M}$ represents M different time series, each consisting of N_i time points (see the beginning of Section III.A). The detection of changes is performed using the whole set of features altogether (i.e., on multivariate time series), using Equations 7 (for a single instance) and 8 (for a window).

We agree with the reviewer on the fact that some aspects needed to be clarified on the paper.

Author action: We updated the manuscript by clarifying how $\overline{D_i}$ and σD_i are computed. In particular, in Section III.B of the revised manuscript, we added Equation 5 and some explanatory text that clarify how they are computed. Moreover, we clarified that the detection of changes exploits the whole set of features altogether and, therefore, identifies changes on multivariate time series (see the beginning of Section III.A).

Reviewer#1, Concern # 6:

A comparison with other techniques in terms of accuracy and sensitivity is not presented.

Author response: We apologize for the possible confusion. As we mentioned in the response to Concern #1, a quantitative comparison with existing state-of-the-art techniques (namely, Isolation Forest, Local Outlier Factor and One Class SVM) is already reported on the paper, in terms of Accuracy, Precision, Recall (a.k.a. sensitivity) and F1-Score in Figures 5-8 for synthetic datasets and in terms of a qualitative evaluation for real datasets.

Author action: We updated the manuscript by reporting a comparison between ECHAD and all the competitors in terms of Precision, Recall, F1 Score and Accuracy and running times on synthetic datasets, and in terms of a qualitative evaluation and running times for the real dataset (see Section IV.C).

Reviewer#1, Concern # 7:

the references need updates

Author response: We appreciate the reviewer's comment. As suggested, we performed an additional literature review and identified some additional relevant works that deserved to be mentioned.

Author action: We updated the manuscript by mentioning and referencing additional relevant articles concerning the identification of anomalies and changes in power grids in Section II. The added references do not resort to the one-class learning setting but are still relevant for the task at hand.

Reviewer#2, Concern # 1:

The experimental setup needs to be clarified in more detail.

Author response: In the previous version of the paper, we reported the parameter configuration for ECHAD and we briefly mentioned that we adopted the default values for competitor methods. We agree with the reviewer that this choice lacked clarity in the experimental setup. In the new version of the paper, we explicitly report the parameter values used in our experiments for all method considered, in order to guarantee the self-consistency of the experimental evaluation.

Author action: We updated the manuscript by explicitly reporting the experimental setup and the values of the parameters used for all methods, including competitor systems (see Section IV.B).

Reviewer#2, Concern # 2:

The reason for shortlisting the three reference method should be explained more clearly.

Author response: Thank you for the interesting comment. Our analysis of the literature reveals that one-class classification methods appear the most suitable class of approaches to address the task of interest in our study. This choice is motivated by the lack of availability of positively labeled data representing change points. In particular, one-class classification methods offer the flexibility to learn a model from an initial data distribution and are able to flag data that significantly differ from the learned distribution. In our experiments, we choose three competitor methods in this class of methods (One-class SVM, Isolation Forest, Local Outlier Factor) which are widely adopted in recent literature in a variety of domains, and are shown to provide highly accurate predictions.

Author action: We updated the manuscript by clarifying our rationale behind the choice of the competitor methods used in the experiments. In the new version of the paper, this discussion is incorporated in Section IV.B.

Reviewer#2, Concern # 3:

The manuscript needs a review of grammar. e.g.
Page 3 line 27 column 1- Only one class of instances is actually know... and column 2 line 29.

Author response: We corrected the language issues highlighted by the reviewer. Moreover, we performed a proofreading iteration taking advantage of the support of a native English speaker. The new version of the manuscript was modified by the authors to incorporate his remarks.

Author action: We updated the manuscript by correcting the language issues highlighted by the reviewer, in addition to other minor issues. These parts are highlighted in the new version of the manuscript.

Reviewer#3, Concern # 1:

In experimental analysis the authors stated that ECHAD is actually more accurate, since its robustness to false detection is not due to a generally higher conservatives is not sufficient to observe the behavior of our system in real scenarios. Need more justification for this statement. Valid justifications of experimental results are required.

Author response: Thank you for the interesting comment. In the paper, a quantitative comparison of ECHAD with respect to existing state-of-the-art techniques (namely, Isolation Forest, Local Outlier Factor and One Class SVM) is reported in terms of Accuracy, Precision, Recall (also known as sensitivity) and F1-Score in Figures 5-8 for synthetic datasets. These results show that the competitors that achieve high precision are strongly conservative, with the result of losing some relevant changes. On the contrary, such a phenomenon is not observed on the results returned by ECHAD, which leads to strong results in terms of both Precision and Recall.

As detailed on the paper, computing such measures on the real dataset is not feasible since the ground truth is not available. However, one important observation from our qualitative analysis is that, in addition to being more conservative than competitor methods (i.e., ECHAD avoids false positives), it also predicts clearly correct changes, that are not detected by the other methods. In the new version of the paper, we better present our qualitative results and emphasize this capability presented by ECHAD.

Author action: We updated the manuscript by improving the discussion in Section IV.C, related to Figures 9-14. Specifically, we emphasized the capability of ECHAD to avoid false positives and to predict clearly correct changes, not detected by competitor systems. We describe in qualitative terms two scenarios identified by ECHAD. In the former, a change in the offset indicates that the power grid is not working properly. In the latter, ECHAD identifies a scenario with a possible phase difference between voltage and current. Both these relevant scenarios were missed by all the competitor systems.

Note: *References suggested by reviewers should only be added if it is relevant to the article and makes it more complete. Excessive cases of recommending non-relevant articles should be reported to ieeeaccesseic@ieee.org*

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.
Digital Object Identifier 10.1109/ACCESS.2017.DOI

ECHAD: embedding-based change detection from multivariate time series in smart grids

MICHELANGELO CECI[1,3,4], ROBERTO CORIZZO[2] (Member, IEEE), NATHALIE JAPKOWICZ[2], PAOLO MIGNONE[1,4] and GIANVITO PIO[1,4]
[1]Department of Computer Science, University of Bari Aldo Moro, Via Orabona, 4, 70125 Bari (BA), Italy (e-mail: name.surname@uniba.it)
[2]Department of Computer Science, American University, 4400 Massachusetts Ave NW, Washington, DC 20016, United States (e-mail: rcorizzo@american.edu, japkowic@american.edu)
[3]Dept. of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, 1000, Slovenia
[4]Big Data Laboratory - National Interuniversity Consortium for Informatics (CINI), Roma, Italy
Corresponding author: Michelangelo Ceci (e-mail: michelancelo.ceci@uniba.it).

ABSTRACT Smart grids are power grids where clients may actively participate in energy production, storage and distribution. Smart grid management raises several challenges, including the possible changes and evolutions in terms of energy consumption and production, that must be taken into account in order to properly regulate the energy distribution. In this context, machine learning methods can be fruitfully adopted to support the analysis and to predict the behavior of smart grids, by exploiting the large amount of streaming data generated by sensor networks. In this paper, we propose a novel change detection method, called ECHAD (Embedding-based CHAnge Detection), that leverages embedding techniques, one-class learning, and a dynamic detection approach that incrementally updates the learned model to reflect the new data distribution. Our experiments show that ECHAD achieves optimal performances on synthetic data representing challenging scenarios. Moreover, a qualitative analysis of the results obtained on real data of a real power grid reveals the quality of the change detection of ECHAD. Specifically, a comparison with state-of-the-art approaches shows the ability of ECHAD in identifying additional relevant changes, not detected by competitors, avoiding false positive detections.

INDEX TERMS Change detection algorithms, smart grids, one-class learning, neural networks, embedding

I. INTRODUCTION

POWER grids are complex systems consisting of generation, transmission, and distribution infrastructures. They represent an important evolution of power grids, where clients are not necessarily passive consumers but have the opportunity to actively participating in the grid, by producing energy from renewable sources and by storing energy through batteries or alternative systems.

One of the most relevant challenges in the context of smart grids is represented by possible changes and evolutions in terms of consumption and production, also due to the influence of some uncontrollable factors. In particular, the production of energy from renewable sources is inherently characterized by instability issues due, for example, to weather conditions. This uncertainty may negatively impact the performance of analytical tools used in power grids for scheduling, planning and regulation purposes.

Additional sources of changes in power grids include variations in the power load as well as the need to adequate the infrastructure to new scenarios (e.g., the installation of car charging stations), that may cause a significant increase of the concurrent consumption of energy, as well as changes in the voltage measured on specific network components.

In this context, machine learning methods can provide significant support in analyzing, optimizing and predicting the behavior of such complex systems, by exploiting the large amount of streaming data generated by sensor networks. Moreover, being able to detect changes from streaming data related to multiple variables (i.e., multivariate time series - see Figure 1) can enable the system to provide prompt alerts that can suggest maintenance activities in a timely manner.

However, the identification of such changes and evolutions in smart grids poses three main challenges:

- sparse and isolated observed peaks should not affect the detection of changes, namely, the system should be robust to possible outliers;

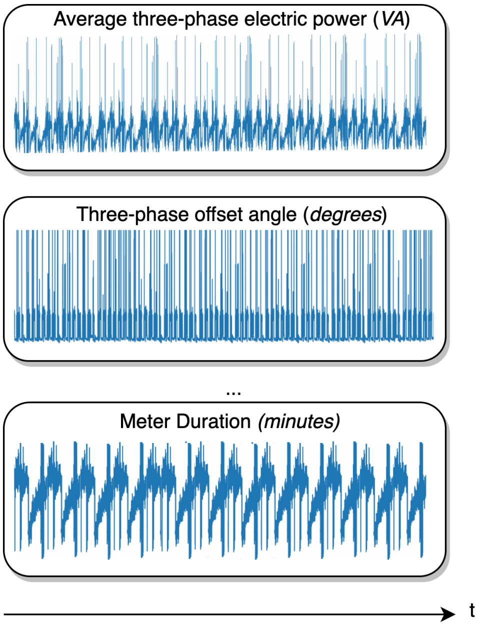

FIGURE 1. A graphical representation of multiple time series corresponding to real data possibly observed in a power grid.

- the amount of available labelled examples is very poor;
- multivariate time series, consisting of a huge number of observed variables, may introduce collinearity phenomena [1] due to possible variable correlation, that may compromise the detection accuracy.

Such challenges make the direct application of classical supervised methods unfeasible, and even semi-supervised methods may appear inadequate due to the strongly unbalancing between the amount of labelled and unlabelled examples. Moreover, in some novel real-world scenarios like that of smart grids, critical conditions or changes have rarely (sometimes never) been observed. Such a situation suggests that the most proper way to tackle this problem is the adoption of approaches able to model the standard/regular scenario and to evaluate the presence of changes according to the coherence with such a model.

In this context, approaches based on one-class learning [2]–[6] find their natural application, since the built model is fitted on one scenario (the regular one) and can subsequently be exploited to detect changes.

Following this line of research, in this paper we propose ECHAD (Embedding-based CHAnge Detection), a novel unsupervised change detection method able to analyze streaming data generated by sensors located in smart grids. ECHAD leverages embedding techniques and a one-class learning approach. The former allow us to extract a new feature space that better represents the inherently complex content of multivariate time series data for the subsequent learning task, also mitigating the collinearity phenomena by incorporating latent interactions among features. The latter (i.e., the proposed one-class learning approach) allows us to analyze data in an unsupervised manner, using only explicit knowledge of the standard/regular behavior of power grids. Finally, ECHAD adopts a novel change detection approach which identifies changes and updates the model accordingly, in order to reflect the new data distribution.

The major contribution of the work can be summarized as follows:

- An investigation of the possible benefits provided by an innovative method that synergically combines embedding techniques and a novel one-class learning approach for tackling the change detection task in multivariate time series data;
- A novel strategy to dynamically adapt the model when changes are detected, in the presence of a concept drift;
- A comprehensive experimental evaluation of the proposed ECHAD, including its parametrization;
- Empirical comparison with state-of-the-art methods on both synthetic and real-world datasets related to power grids.

The rest of the paper is organized as follows. In Section II we discuss the work related to this paper, from both application and methodological point of views; in Section III we describe in detail our proposed method ECHAD; in Section IV, we describe the results obtained on both synthetic and real-world datasets, showing the competitiveness of ECHAD with respect to state-of-the-art methods; finally, in Section V, we draw some conclusions regarding the applicability of ECHAD as a powerful tool in analytical tasks for smart grids, and outline possible future works.

II. BACKGROUND

Several machine learning approaches have been proposed in the literature to support analytical tasks in the energy field. Among them, significant efforts have been devoted to the forecasting of the energy produced by plants in smart grids [7]–[11]. Solving this task is particularly important to support grid power balancing, especially when the energy is produced by renewable sources. At the same time, accurate predictions of the energy produced at a specific time horizon may be useful for other scenarios, such as the optimization of energy trading operations [12], [13].

Recently, research activities have been directed towards approaches for the simultaneous forecasting of the energy produced in multiple plants, mainly exploiting time series analysis [7], autoregressive (AR) models [8], predictive clustering models [9], artificial neural networks (ANNs) [10], or SVM classifiers [11]. Recent studies [10], [14]–[18] have also investigated the possible exploitation of spatial and temporal autocorrelation phenomena to improve forecasting accuracy. For example, in [17], the authors exploit geo-distributed weather observations in the neighborhood of wind plants, while in [14], the authors extract statistical indicators that model the spatio-temporal autocorrelation between plants for each descriptive feature.

The common aspect among these solutions consists of the possible exploitation of additional factors, including temporal and spatial closeness among multiple plants, as well as external uncontrollable factors (e.g., measured or predicted weather conditions). The main motivations for taking into account these additional factors come from the possible simultaneous changes and evolutions of the behavior that can be observed in plants working in similar conditions (e.g., spatially closed and subject to similar weather conditions). These motivations also justify the need to detect and model changes in the distribution of some variables (also known as *concept drift* [19] [20] in the literature), that may be fruitfully exploited to timely predict changes in similar/related plants.

The focus of the present paper is specifically in this area of research. In particular, we propose a method to detect changes in time series, possibly coming from sensors. As introduced in Section I, our approach works in an unsupervised setting and models the standard/regular scenario to properly detect changes in the data distribution.

In this context, existing methods mostly rely on the one-class learning setting. Alternative methods are based on Long-Short Term Memory neural networks [21], Empirical Mode Decomposition [22], Symbolic Dynamic Filtering [23] and the Margin Setting Algorithm [24], although they focus primarily on the detection of anomalies and attacks in the smart grid, rather than generic changes.

One-class learning was first proposed in [25] and subsequently studied in [26] and [27]. Differently than binary (or multi-class) classification approaches, that learn to discriminate between positive and negative examples (or among multiple classes), one-class learning methods focus on modeling one single class of examples and identify whether unseen examples belong to the learned class or not. A similar rationale has found application also in outlier detection [28], novelty detection [29] and positive-unlabelled learning [30] approaches.

It is important to mention that, in classical supervised settings, standard (binary or multi-class) classification methods easily outperform one-class learning approaches in discriminating among the possible classes [31]. However, there are specific scenarios in which one-class learning approaches are the most appropriate, or even the only applicable solutions. Such scenarios include situations in which:

- Only one class of instances is actually known, while other possible classes are not known a-priori.
- The dataset at hand is strongly unbalanced. In this case, most standard approaches may be biased towards the majority class.
- The goal is explicitly to detect rare, particular situations.

When the task under consideration is the detection of changes, as in the specific application domain considered in this paper, the stable/regular situation is strongly over-represented in the available dataset, and only a small fraction of instances representing the changes is actually available. This aspect confirms that the adoption of a one-class learning approach is the most suitable solution.

In the literature, several one-class classification methods have been proposed. Among them, it is worth mentioning One-Class SVM [2], [32]–[35], Isolation Forest [3], [36], [37], One-Class Local Outlier Factor (LOF) [4], [38], [39] and approaches based on autoencoders [5], [6].

One-Class SVM [2], [32] is an unsupervised learning algorithm that learns a decision function for the detection of changes. One-Class SVM learns such a function exclusively from instances of a single class and classifies new instances as similar or different to the training set. One recognized limitation of this approach is the possibility to deal with high-dimensional data, and its sensitivity to the presence of outliers in training data [40].

LOF [4] measures the local density deviation of a given instance concerning its neighbors. In particular, the LOF score of an instance is computed as the ratio of the average local density of its k-nearest neighbors, and its own local density. Instances appearing similar to the training data distribution are expected to exhibit a local density similar to that of its neighbors. On the contrary, instances representing a change in the distribution are expected to show a much smaller local density. However, when this locality property is not present/satisfied in the application domain at hand, the performance of these methods may be compromised.

Isolation Forest [3] is a tree-based method that isolates instances that appear different than the training data distribution. The algorithm recursively partitions the sample of instances by randomly selecting a feature and a split value. Instances that require a small number of splits to be isolated in leaf nodes are more likely to represent changes or outliers with respect to the training data distribution. Isolation Forest shows a low time complexity and low memory requirements. However, high-dimensional data may affect their detection performance. Moreover, its decision boundaries are limited to vertical and horizontal shapes.

Approaches based on autoencoders learn the data distribution of one-class data through special kinds of neural networks. Subsequently, the learned distribution is exploited to determine whether new instances belong to the same known distribution or differ from it significantly. They offer the opportunity to learn non-linear relationships in the data, by exploiting non-linear activation functions in the hidden layers. However, existing approaches [5], [6], [16] are mostly focused on identifying point anomalies, and do not address the problem of identifying changes in the data distribution, which is the main focus of this study.

Relevant surveys presenting one-class classification methods include [41], [42], and [43], whereas surveys discussing anomaly or outlier detection methods that include one-class classification can be found in [44] and [45].

Compared with such existing approaches, ECHAD has the advantage to properly deal with possible collinearity issues of multi-variate time series, thanks to the embedding approach adopted, and to the capability of dynamically adapting the model when changes are detected, in the presence of a concept drift.

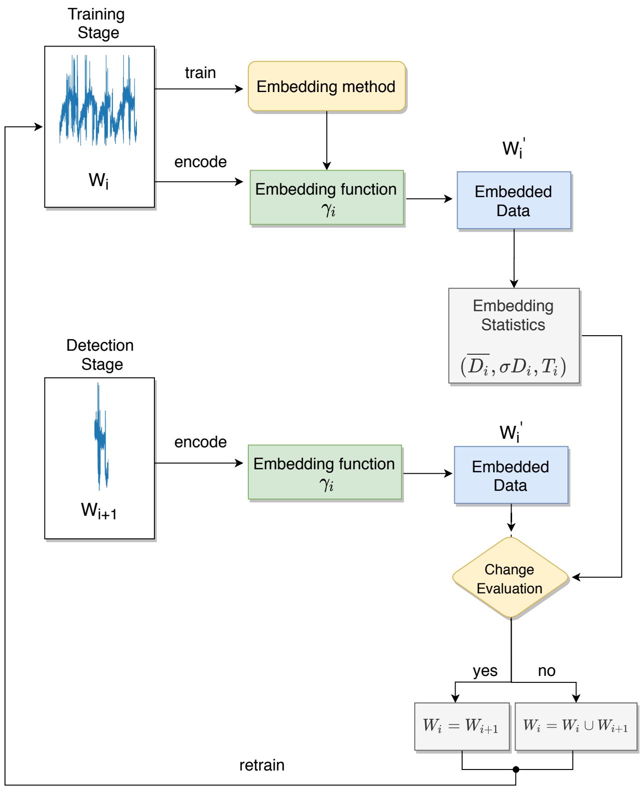

FIGURE 2. A graphical representation of the proposed method implemented in ECHAD

III. THE METHOD ECHAD

In this section, we describe our novel approach ECHAD, an embedding-based change detection algorithm that is able to detect changes in time series data generated by smart grids. We stress that the peculiarity of our approach is the combination of an embedding solution with a one-class learning method that is able to dynamically update the learned model. These aspects allow ECHAD to analyze complex and dynamic multivariate time series and to identify changes in the data distribution, leveraging exclusively the knowledge of the standard/regular behavior of the smart grid.

ECHAD consists of two main phases, namely *i)* learning an embedding model from historical time series data falling into a specific interval (time window); *ii)* detecting changes on newly observed data, using a streaming test-and-retrain workflow. A graphical overview of the general workflow followed by ECHAD is depicted in Figure 2, while in the following subsections, we explain its main phases in detail.

A. LEARNING EMBEDDING MODELS

Let W_i be a time window consisting of M time series, corresponding to M features measured over N_i time points. In this phase, we learn a reduced, latent, K-dimensional feature space, with $K \ll M$. More formally, the time series data of the time window can be represented as a matrix $W_i \in \mathbb{R}^{N_i \times M}$, and the goal is to learn a function γ_i: $\mathbb{R}^M \to \mathbb{R}^K$, that maps each M-dimensional time point of a time series to the reduced, K-dimensional feature space. The function, although learned from the time window W_i, can naturally be applied to other, also unseen, time points in order to project their features into the reduced feature space.

To perform this step, any approach to identify a reduced feature space can be plugged into our system. In this paper, we consider the classical Principal Component Analysis (PCA) [46] and the more recent Stacked Auto-encoders [47], [48], for which we provide some details in the following.

PCA is one of the most popularly known dimensionality reduction technique, which effectiveness has been shown in several scientific fields, ranging from chemistry to geology [49]–[51]. Specifically, PCA estimates the correlation among the variables and extracts a reduced set of features that are as much as (linearly) uncorrelated as possible. This transformation is performed such that each extracted feature, called principal component, explains the largest possible amount of data variance, with the constraint of being orthogonal to all the previously extracted features. In this way, PCA extracts a reduced representation of the data, that explains a given overall percentage of data variance, possibly discarding the noise. Formally, given the input matrix $W_i \in \mathbb{R}^{N \times M}$, PCA computes the covariance matrix $C \in \mathbb{R}^{M \times M}$, from which it extracts the first K eigenvectors, associated to the largest eigenvalues, obtaining the matrix $Z_i \in \mathbb{R}^{M \times K}$. The matrix Z_i can finally be used to compute the embedding of a new time point $w \in \mathbb{R}^M$ as follows:

$$\gamma_i(w) = Z_i^T \cdot w \tag{1}$$

While PCA properly deals with collinearity problems thanks to the orthogonality of the extracted features, its main limitation is in its ability to catch only linear dependencies among variables. Such a limitation also defines one of the strong points of an alternative approach that has recently been proposed in the literature, namely stacked auto-encoders. They are special kinds of neural networks, whose main purpose is to reconstruct a given data distribution with the lowest possible reconstruction error. The dimensionality reduction is achieved by exploiting bottleneck features extracted at their hidden layers [52].

Thanks to the stacked structure, each layer represents data at a different abstraction level. For example, in the domain of images, the first layer may represent edges, while deeper levels may represent contours or corners of objects.

More formally, an auto-encoder aims at learning two functions, namely the encoding function $e : \mathbb{R}^M \to \mathbb{R}^K$ and the decoding function $d : \mathbb{R}^K \to \mathbb{R}^M$, such that:

$$\langle e_i(\cdot), d_i(\cdot) \rangle = \arg \min_{\langle e_i(\cdot), d_i(\cdot) \rangle} \| W_i - d_i(e_i(W_i)) \|^2 \tag{2}$$

It is noteworthy that the encoding function e can be directly used as an embedding function for new time points $w \in \mathbb{R}^M$, namely:

$$\gamma_i(w) = e_i(w) \tag{3}$$

As previously mentioned, auto-encoders are potentially able to catch non-linear relationships among features. This is achievable by adopting non-linear functions as activation functions in their hidden layer. On the other hand, there is no guarantee on the orthogonality of the extracted features, since they are identified on the basis of the reconstruction error. Therefore, collinearity issues may still be present in the reduced space.

B. CHANGE DETECTION

Let $W_i \in \mathbb{R}^{N_i \times M}$ be the time window currently designated to train the model. After exploiting it to learn the embedding function γ_i, we compute $W_i' \in \mathbb{R}^{N_i \times K}$ by applying γ_i to each time series in W_i. More formally:

$$W_i' = \Gamma_i(W_i) = \begin{bmatrix} \gamma_i(W_i[1,*]) \\ \gamma_i(W_i[2,*]) \\ \dots \\ \gamma_i(W_i[N_i,*]) \end{bmatrix} \tag{4}$$

where $W_i[j,*]$ represents the whole j-th row (i.e., the j-th time series) of the matrix W_i. Intuitively, $\Gamma_i(\cdot)$ represents the learned model valid at time i.

Then, we use W_i' to compute $\overline{D_i}$ and σD_i, which are the mean and the standard deviation, respectively, of the Euclidean distance between an instance and its p nearest neighbors, in the reduced K-dimensional feature space. Formally, let: $x \in R^K$ be a training instance belonging to N_i and $Neigh(x)$ be the set of the p nearest neighbors of x. Then:

$$\overline{D_i} = \frac{1}{p} \sum_{q \in Neigh(x)} eucl\_dist(x, q)$$

$$\sigma D_i = \sqrt{\frac{1}{p} \sum_{q \in Neigh(x)} (eucl\_dist(x, q) - \overline{D_i})^2} \tag{5}$$

where $eucl\_dist(a, b)$ is the Euclidean distance between a and b. $\overline{D_i}$ and σD_i allow us to estimate the data distribution, and to define a threshold T_i that is exploited to detect if future observations deviate significantly from the current data distribution. The threshold T_i is calculated using a τ-sigma rule as follows:

$$T_i = \tau \cdot \sigma D_i \tag{6}$$

where τ is a user-defined parameter.

When new data arrive, belonging to a new time window $W_{i+1} \in \mathbb{R}^{N_{i+1} \times M}$, we compute W_{i+1}' by exploiting the previously learned embedding function. Formally, following Equation (4), we compute W_{i+1}' as $W_{i+1}' = \Gamma_i(W_{i+1})$.

Using W_{i+1}', for each time series (i.e., row of the matrix) $w \in W_{i+1}'$[1], we compute D_{i+1}^w that is the mean of the Euclidean distance between w and its p-nearest neighbors.

[1]Here w is a time series of the window, namely a row of the matrix.

Using such a measure, we consider an instance w as a change (or not) when the following function is 1 (or 0):

$$c(w) = \begin{cases} 1 & \text{if } (D_{i+1}^w > \overline{D_i} + T_i) \lor (D_{i+1}^w < \overline{D_i} - T_i) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

It is noteworthy that such a change is defined at an instance level. Considering it as the final output of our detection approach would lead to being highly sensitive to outliers and spurious peaks. To overcome this issue, we work at the level of time window, and consider it, i.e. the window, as a change if more than a given ratio c_r of instances are detected as a change. Formally:

$$C(W_{i+1}) = \begin{cases} 1 & \text{if } \left(\sum_{w \in W_{i+1}} c(w) \right) > (N_{i+1} \cdot c_r) \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Independently of the output of $C(W_{i+1})$, ECHAD adapts the model representing the data distribution of regular scenarios, leading to a new threshold T_{i+1}. In particular, if a change is not detected, the embedding function is updated considering a merged time window $W_i \cup W_{i+1}$. Note that, following the mixed windows model [53], W_i can be either the single window preceding W_{i+1}, or a wider window obtained by merging multiple previous windows, when no change was detected (see Figure 3). On the contrary, if a change is detected, previous windows are discarded and the embedding function is re-learned from scratch only from W_{i+1}.

This strategy allows ECHAD to simultaneously be robust to the presence of outliers and to properly adapt to new data distributions for proper detection of subsequent changes.

A final remark regards the computational complexity of ECHAD. This can be easily computed by summing up the complexity of the embedding phase ($O(N_i \cdot M^2 + M^3)$ for PCA and $O(N_i \cdot M^2)$ for autoencoders), the complexity of identifying the p neighbors for each instance ($O(N_i \cdot logN_i)$, assuming to use a tree-based structure), and the complexity of computing $\overline{D_i}, \sigma D_i$ and T_i ($O(p \cdot K \cdot N_i)$, according to Equation 5). Therefore, the overall time complexity of ECHAD is $O(N_i \cdot M^2 + M^3 + N_i \cdot logN_i + p \cdot K \cdot N_i)$ for the variant based on PCA and $O(N_i \cdot M^2 + N_i \cdot logN_i + p \cdot K \cdot N_i)$ for the variant based on autoencoders. If we assume that $M^2 > logN_i$ and $M^2 > p \cdot K$, we have that the total time complexity is as follows: $O(N_i \cdot M^2 + M^3)$ for the variant based on PCA and $O(N_i \cdot M^2)$ for the variant based on autoencoders.

IV. EXPERIMENTS

In this section, we present the experiment for the evaluation of ECHAD. First, we introduce the adopted datasets and the experimental setting, together with the considered state-of-the-art competitor systems. Finally, we show and discuss the obtained results.

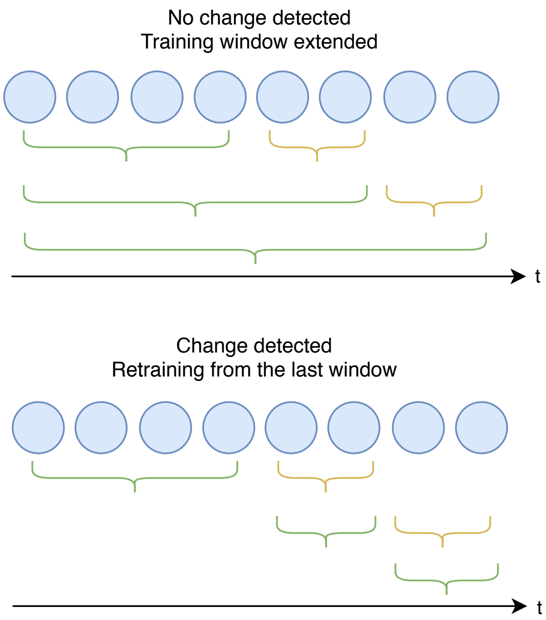

FIGURE 3. A graphical representation of the adopted mixed windows model. The schema shows the evolution of the training (green) and testing (yellow) windows over time. When no changes are detected, the training window is extended over time (see the top part of the figure). When a change is detected, the last testing window becomes the new training window (see the bottom part of the figure).

A. DATASETS

We performed experiments with five different datasets. The first four datasets are synthetically generated and represent different change detection scenarios that are relevant to power grids. The fifth dataset consists of real-time series observed in a real power grid and allows us to observe the behavior of our system in real scenarios.

The synthetic datasets have been generated by considering 5 multivariate (20 variables) Gaussian distribution of $2,500$ time points, with $\mu \in \{10, 20, 35, 80, 110\}$ and a varying standard deviation $\sigma \in \{5, 8, 10, 12\}$. The four resulting datasets represent an increasing level of complexity. Indeed, datasets with a low standard deviation (i.e., $\sigma \in \{5, 8\}$) are visibly characterized by narrow Gaussian curves (see Figure 4 (a) and (b)), which potentially facilitate the change detection task due to the weak overlap among them. On the contrary, datasets with a larger standard deviation (i.e., $\sigma \in \{10, 12\}$) are characterized by wider Gaussian curves (see Figure 4 (c) and (d)), which reasonably lead to a higher difficulty in the change detection task due to the significant overlap among the Gaussian curves.

The real-world dataset has been provided in the context of the project "ComESto - Community Energy Storage" (http://www.comesto.eu/) for the Italian energy distribution network, that is managed by *e-distribuzione S.p.A*. The data consist of 200 variables measured by sensors located into medium voltage/low voltage (MV/LV) transformer rooms, related to 131,374 time points falling in the period from November 1, 2019 to December 19, 2019.

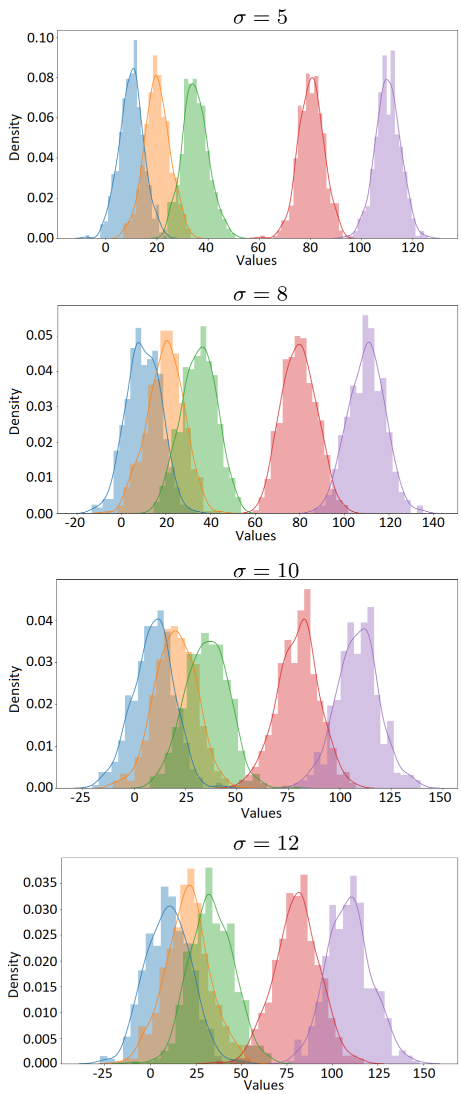

FIGURE 4. A graphical representation of the synthetic datasets, with different values of the standard deviation σ.

B. EXPERIMENTAL SETUP

As discussed in Section II, the most suitable class of approaches to address the task of interest in our study is that of one-class classification methods. Indeed, they offer the flexibility to learn a model from an initial (regular) data distribution and are able to flag data that significantly differ from the learned distribution. For this reason, in order to evaluate the performance obtained by ECHAD, in our experiments we considered three state-of-the-art competitor methods falling in this class, namely One-Class SVM [2], [33]–[35], Isolation Forest [3], [36], [37], and LOF [4], [38], [39], which are widely adopted in the recent literature, and are shown to provide highly accurate predictions.

After a preliminary evaluation, their parameters were set to the values suggested in their respective papers. In particular, for Isolation Forest, we set: the number of base estimators in the ensemble $n\_estimators = 100$; the contamination of the dataset, i.e., the proportion of expected outliers,

$contamination = 10\%$; the number of features to draw at random for each base estimator $max\_features = M$, i.e., the whole set of features. For LOF, we set: the number of neighbors to use for k-neighbors queries $n\_neighbors = 2$; $Minkowski$ measure as distance measure. For One-Class SVM, we set: the coefficient of the Radial Basis Function (RBF) kernel $gamma = 0.1$.

As regards ECHAD, for the autoencoder we used the LBFGS optimizer, with the objective of minimizing the reconstruction error on training data. We set the maximum number of training epochs to 500 and the minimum reduction of the training error between two subsequent epochs, used as early stopping criterion for the learning phase, equal to $10e^{-5}$. The number of hidden layers of the auto-encoder architecture is set to 3, which leads to an architecture of 5 total layers that takes input data and performs two stages of encoding and decoding. After preliminary experiments, we set other parameter values as follows:

- The number of neighbors considered for the identification of the k-nearest neighbors has been set to $p = 100$;
- The dimensionality of the reduced embedding space has been set to $K = 5$;
- A window is considered to represent a change if it contains at least 70% of data instances identified as a change, namely ($c_r = 0.7$);
- $\tau = 1$, which means that an instance is considered as a change if it differs from the mean observed in the training window of at least σD_i (see Equation (6));

For the synthetic datasets, the size of the testing window has been set to $N_{i+1} \in \{25, 50, 75\}$, while the size of the first training window has been set to the double of the size of the testing window, namely $N_1 = 50$, $N_1 = 100$ and $N_1 = 150$, respectively. For the real dataset, considering the larger amount of data points, we considered $N_{i+1} = 190$ (approximately 1.5 hours) and $N_1 = 750$ (approximately 6 hours). These ranges of values have been suggested by the domain experts, participating in the project.

For the synthetic datasets, where the ground truth is known, the performances of the considered systems have been evaluated in terms of Precision, Recall, F-Score and Accuracy, while for the real dataset, the performances have been evaluated from a qualitative point of view, involving an expert in the evaluation.

C. RESULTS AND DISCUSSION

The results on the synthetic datasets show that ECHAD combined with PCA (denoted as ECHAD-PCA) achieved the best results by catching correctly all the windows representing changes (see Figures 5 - 8). ECHAD combined with autoencoders (denoted as ECHAD-AUTOENC) returned some false detections only in the configuration presenting the largest window size (i.e., $N_1 = 150$ and $N_{i+1} = 75$). This result depends on the challenging scenario of catching a change in data with a large training window, that presents multiple heterogeneous data distributions (see Figure 4). However, in a real setting, training the model on a single data distribution is a reasonable assumption that limits the occurrence of false positives. Under this condition, the method correctly identifies changes and is re-trained using the new data distribution.

Comparing the results with those obtained by competitors, it is clear that in most cases, ECHAD outperforms them using both embedding-based models (PCA and autoencoders). Moreover, the initial intuition about the low difficulty in detecting changes on the first dataset ($\sigma = 5$) is confirmed, since it consists of four non-overlapped and narrow Gaussian curves. On the other hand, when the dataset presents overlapping and large Gaussian curves (i.e., $\sigma \in \{8, 10, 12\}$), the task became harder and induced the competitors to worse results. Going into detail, we can observe that ECHAD-PCA achieved an F1-score equal to 1.0 in all the situations, while different competitors acted differently in terms Precision and Recall. In particular, we can observe that all the methods achieved a precision of 1.0, meaning that they did not produce false positive detections, but the measured Recall is very low in some cases (see OneClass SVM and LOF).

This means that such a high precision was achieved through a strongly conservative strategy, that led to losing some relevant changes. On the contrary, such a phenomenon is not observed on the results returned by ECHAD, which led to strong results in terms of both Precision and Recall.

A comparison in terms of running times revealed that ECHAD was able to complete every single run on average in 6 minutes; OneClass SVM required on average 2.5 seconds; LOF required on average 5 seconds; Isolation Forest required on average 6.5 minutes. Although OneClass SVM and LOF show significantly lower running times than ECHAD and Isolation Forest, their results, as already shown in Figures 5-8, are much worse. Note that ECHAD simultaneously shows higher Precision, Recall, F1 Score and Accuracy, and lower running times with respect to Isolation Forest that, in these experiments, appears to be the strongest competitor.

As regards the analysis of the real dataset, a quantitative evaluation was actually not feasible, due to the lack of the ground truth. In this case, we show a graphic representation of the time windows that ECHAD correctly identified as changes, that were missed by all competitor methods. We focus on two time series that are well known to be characterized by changes in power grids, namely, *three-phase offset angle* and *three-phase total reactive power*. The first time series is considered important since a change in the offset (ideally, the offset should be close to 120 degrees) indicates that the network is not working properly. Whereas, the second time series indicates a possible phase difference between voltage and current. These results correspond to real scenarios of changes in a power grid and, therefore, are highly important to detect, in order to provide alerts and trigger timely maintenance activities.

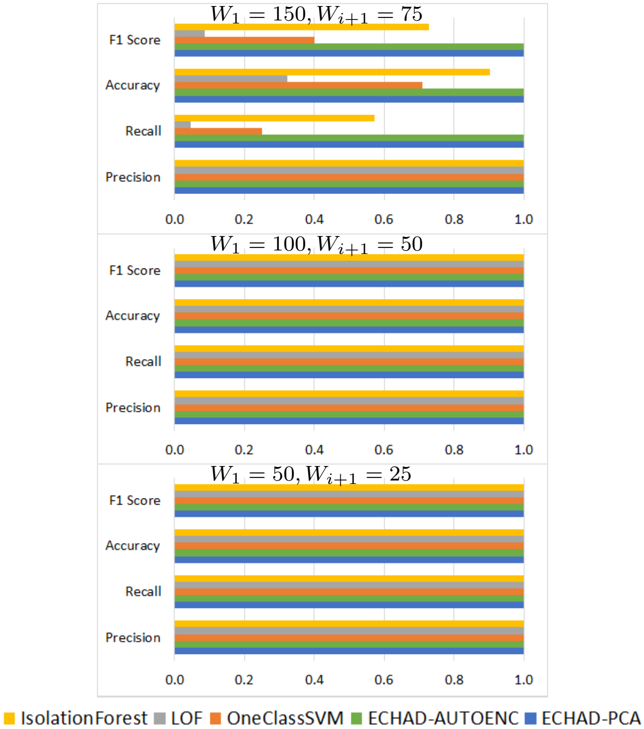

FIGURE 5. F1 Score, Accuracy, Precision, and Recall measured on the first synthetic datasets ($\sigma = 5$).

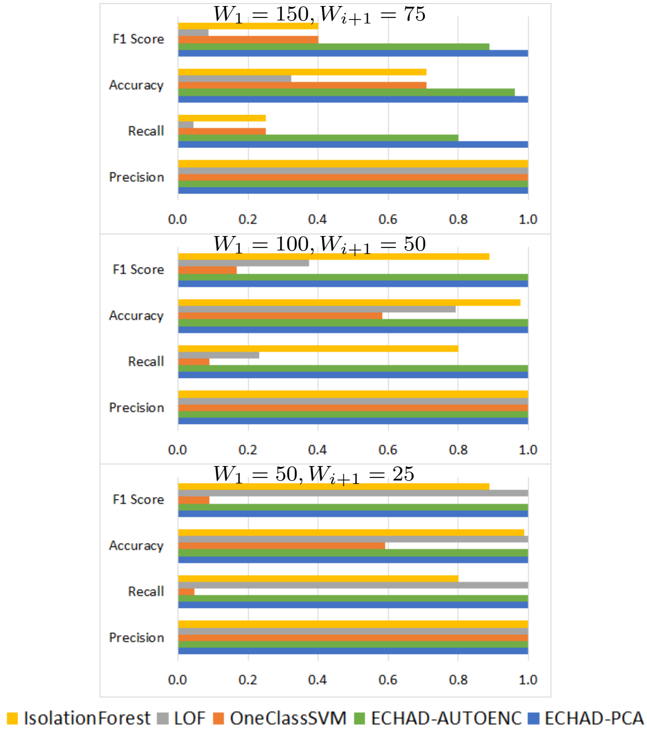

FIGURE 7. F1 Score, Accuracy, Precision, and Recall measured on the third synthetic datasets ($\sigma = 10$).

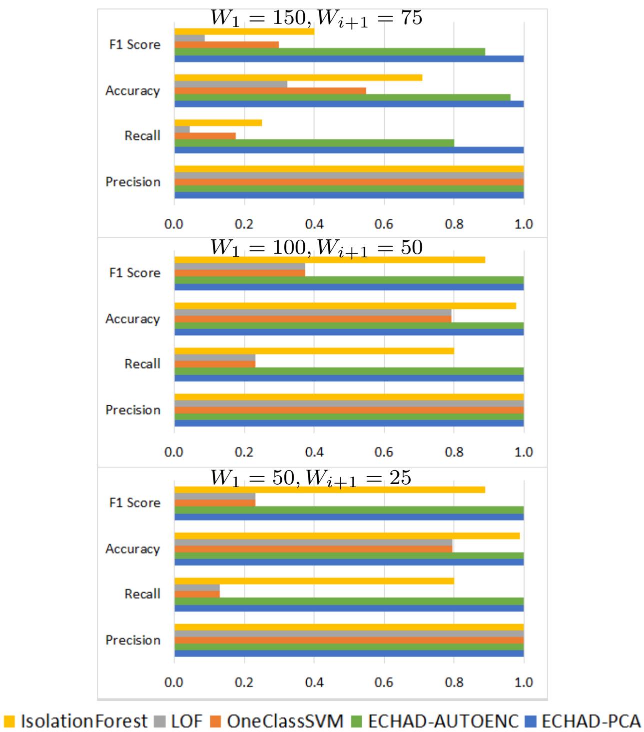

FIGURE 6. F1 Score, Accuracy, Precision, and Recall measured on the second synthetic datasets ($\sigma = 8$).

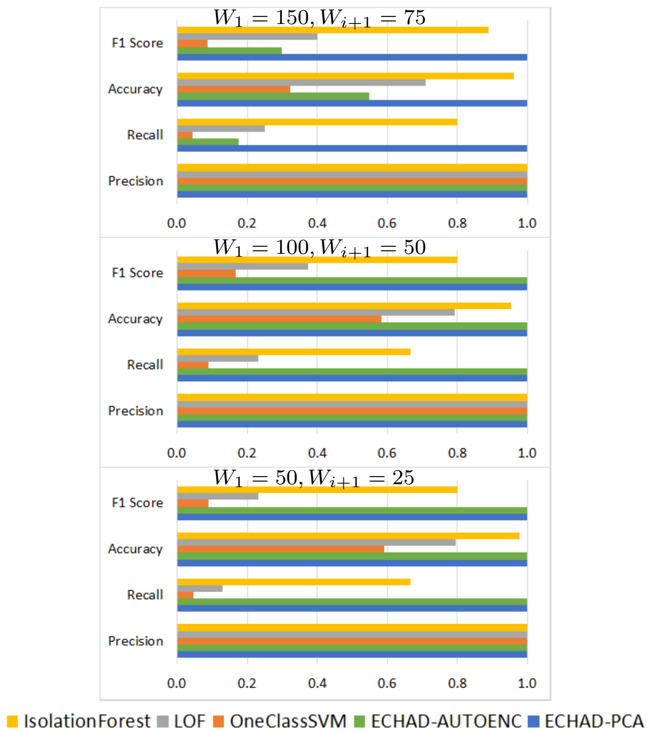

FIGURE 8. F1 Score, Accuracy, Precision, and Recall measured on the fourth synthetic datasets ($\sigma = 12$).

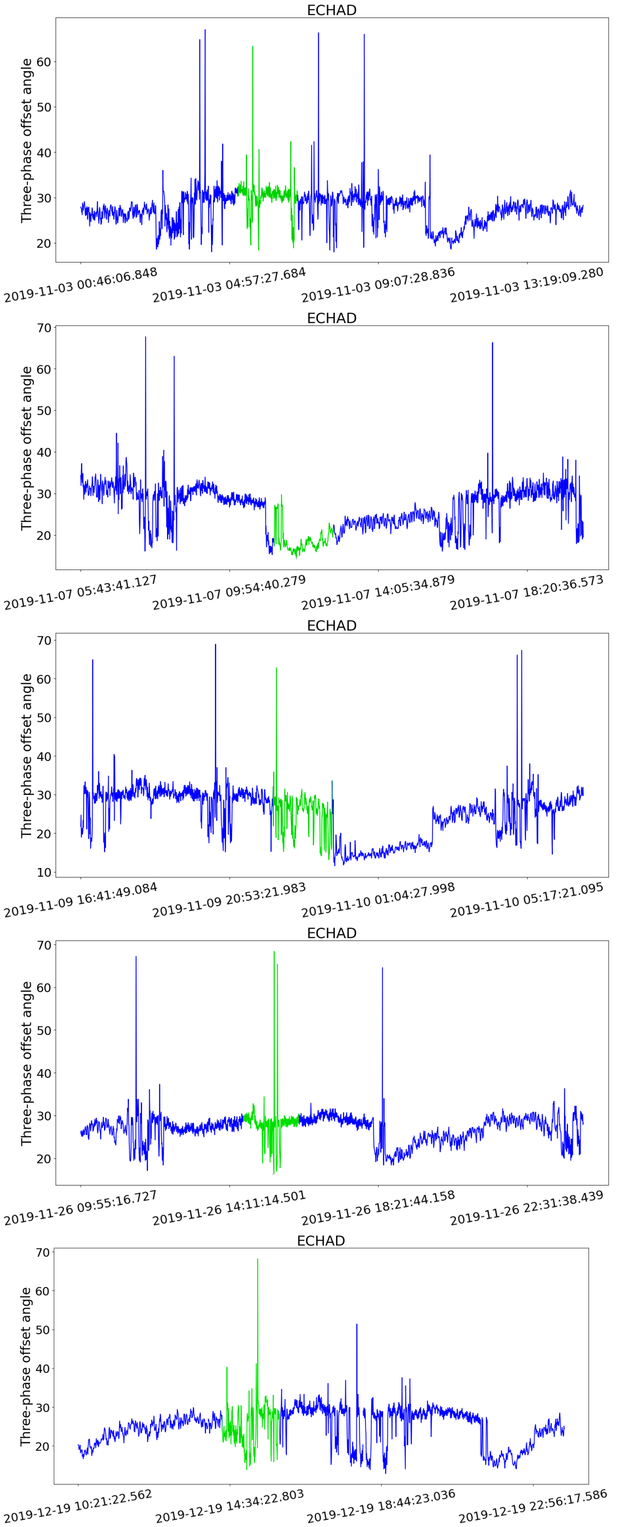

FIGURE 9. Examples of time windows representing a change (in green), identified by ECHAD but not by Isolation Forest and LOF for *three-phase offset angle*.

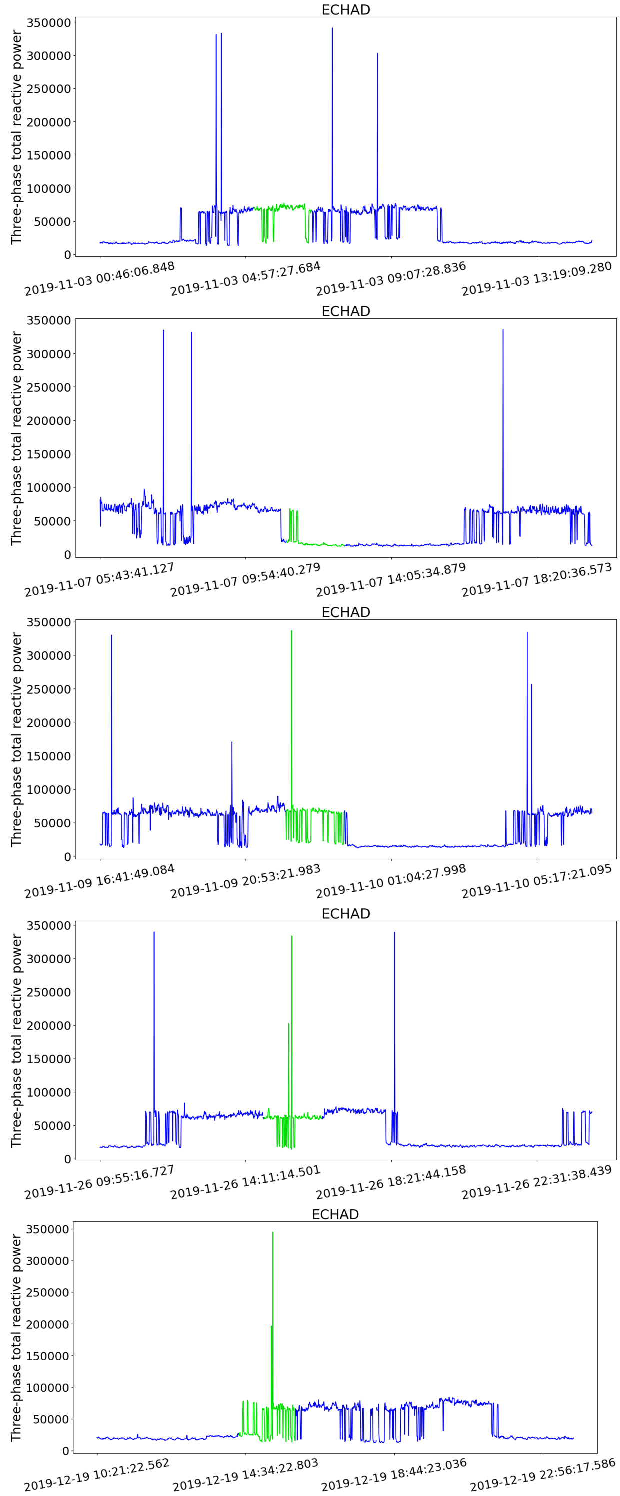

FIGURE 10. Examples of time windows representing a change (in green), identified by ECHAD but not by Isolation Forest and LOF for *three-phase total reactive power*.

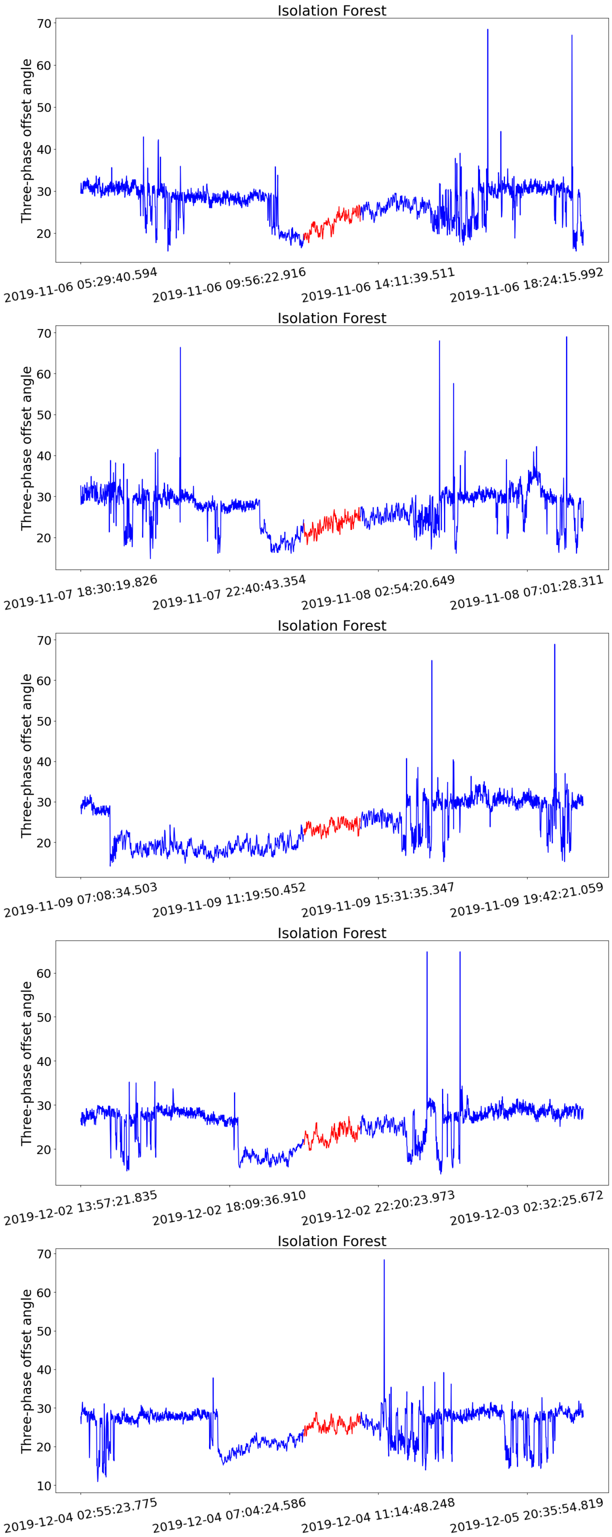

FIGURE 11. Examples of false positive detections (in red) returned by Isolation Forest, that were not returned by ECHAD, for *three-phase offset angle.*

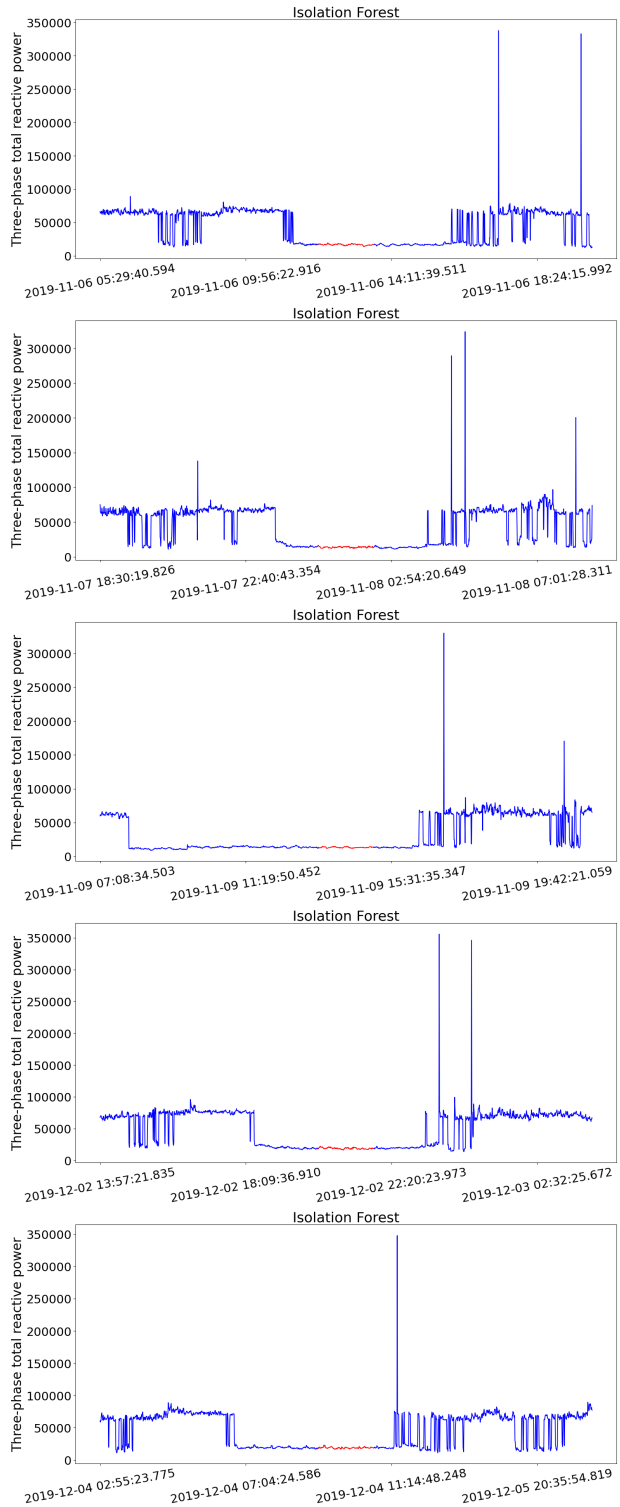

FIGURE 12. Examples of false positive detections (in red) returned by Isolation Forest, that were not returned by ECHAD, for *three-phase total reactive power*.

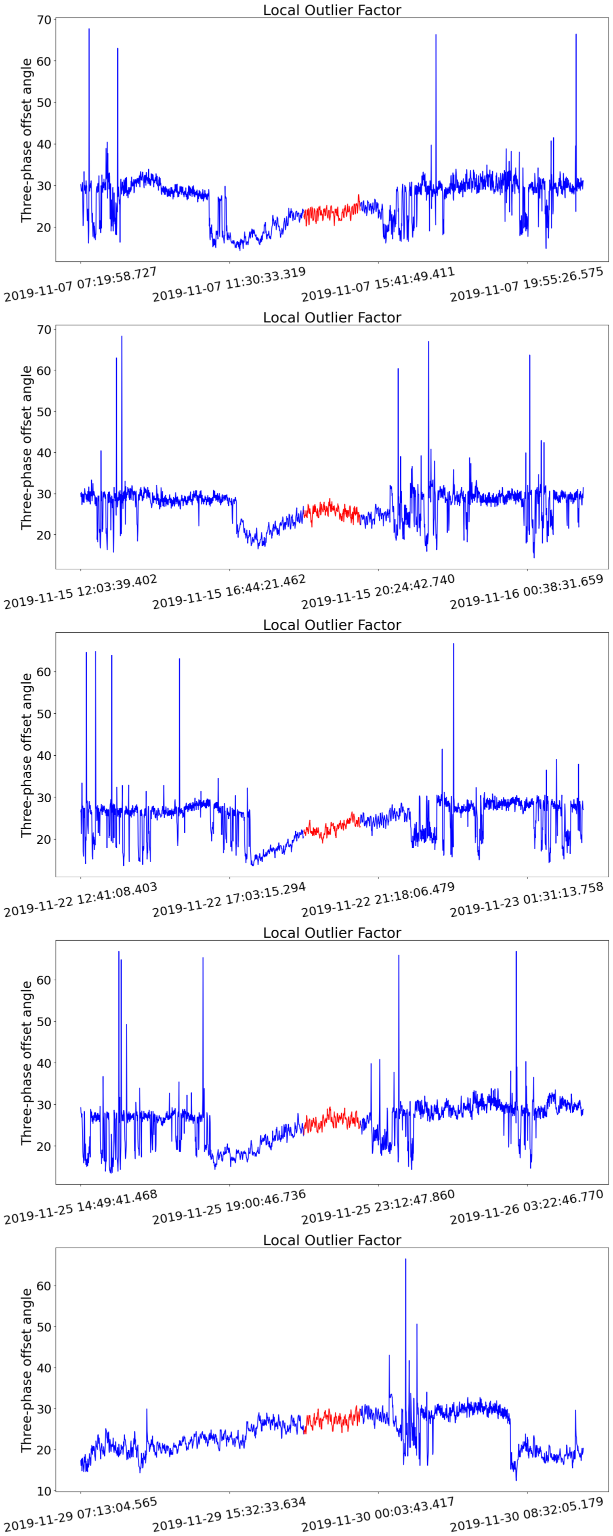

FIGURE 13. Examples of false positive detections (in red) returned by LOF, that were not returned by ECHAD, for *three-phase offset angle*.

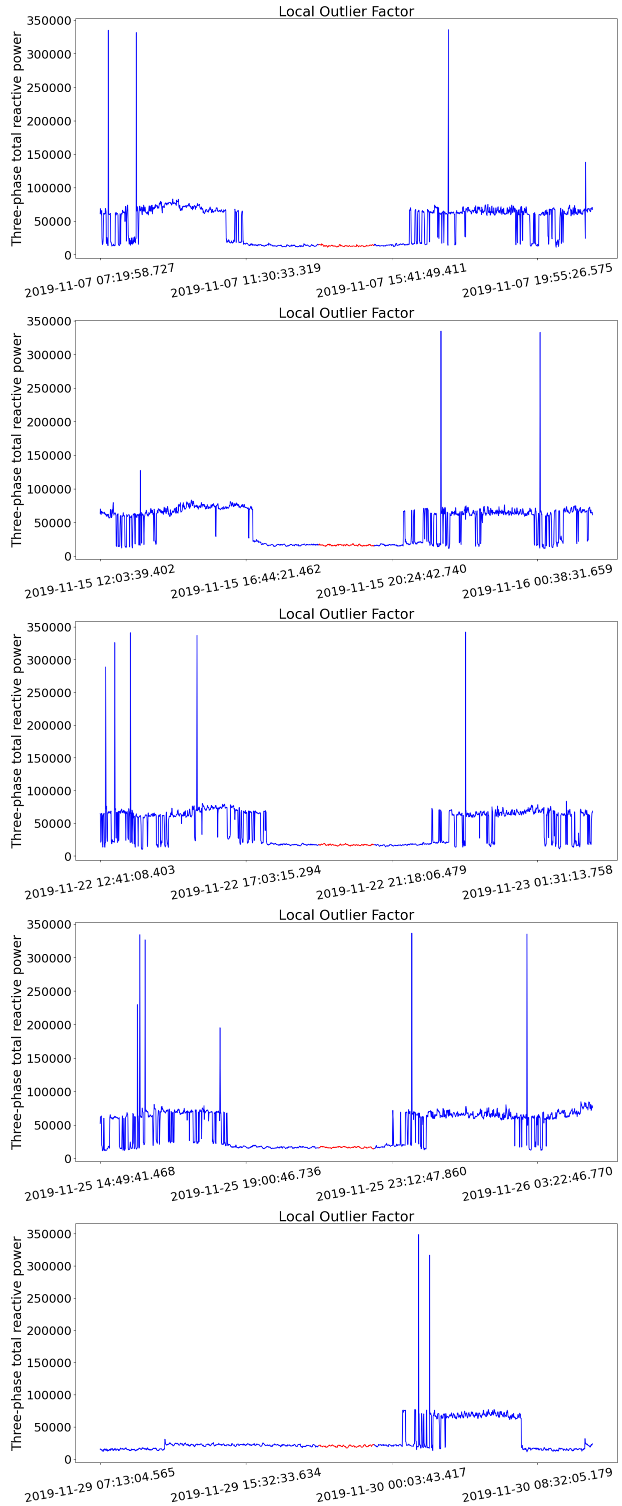

FIGURE 14. Examples of false positive detections (in red) returned by LOF, that were not returned by ECHAD, for *three-phase total reactive power*.

Since ECHAD-PCA achieved the best results on synthetic datasets, we considered this variant for the analysis of real data. In Figures 9 and 10, we graphically emphasize in green the windows that appear to be correctly flagged as changes by ECHAD, that were missed by the competitors. Such situations clearly represent false negatives for the competitors.

On the other hand, in Figure 11-12 and 13-14, we emphasize in red the time windows that appear to be incorrectly flagged as changes by Isolation Forest and LOF, respectively, that were correctly considered by ECHAD as stationary. These cases clearly represent false positives for competitor approaches. As concerns One-Class SVM, we do not report the results since it appeared to be too sensitive, with the effect of identifying many changes (and high false positive rate).

Although we cannot compute specific performance metrics on the real dataset due to the lack of ground truth, we can note that, out of a total of 687 windows, ECHAD detected 77 windows as a change, Isolation Forest detected 87 windows, LOF detected 77 windows and One-Class SVM detected all the 687 windows. Such results may lead to observe that ECHAD is more conservative than competitors. However, it is noteworthy that it also correctly identified several changes that were ignored by competitors (see Figures 9 and 10). This means that ECHAD is actually more accurate, since its robustness to false detections is not due to a generally higher conservativeness. This is also confirmed by the previous analysis of synthetic data (see Figures 5-8), where we observed that being conservative is not a sufficient condition that systematically leads to high-quality change detection.

The obtained results confirm our initial intuitions: The proposed method ECHAD, thanks to the adopted embedding approach and to the dynamic update of the model, is able to correctly detect changes in time series generated by sensors located in smart grids and to properly adapt the model according to such changes, limiting the number of false positives and providing high-quality detections that were not identified by competitors and that, according to a manual visual inspection by the experts, appear to be realistic.

A comparative analysis in terms of running times shows analogous phenomena to those observed on synthetic datasets. In particular, ECHAD required an average of 9 minutes; One-Class SVM required an average of 6 seconds; LOF required an average of 3 minutes; Isolation Forest required an average of 8.5 minutes. In conclusion, ECHAD outperformed Isolation Forest in terms of accuracy with similar running times, while the other competitors, even if showed lower running times, obtained significantly worse performances in terms of accuracy.

V. CONCLUSION

In this paper, we proposed ECHAD, a novel unsupervised change detection method able to analyze streaming data generated by sensors located in smart grids. The embedding techniques we implemented in ECHAD allow us to extract and exploit a new feature space that better represents the inherent complexity of multivariate time series, also mitigating collinearity phenomena and catching latent interactions among features. On the other hand, the proposed one-class learning approach, supported by a novel change evaluation method and a dynamic strategy to update the model, allow ECHAD to identify changes accurately.

Our experimental evaluation showed that, compared to three state-of-the-art methods, ECHAD achieves optimal change detection performance on synthetic data, also in challenging scenarios that present a high degree of overlap between evolving data distributions. Moreover, ECHAD showed high-quality results on real data observed in a real power grid. In particular, it detected several changes in the data, that were qualitatively confirmed and that were not detected by competitors. On the other hand, contrary to the competitors, ECHAD was robust to false positive detections.

As future work, we plan to integrate some existing techniques tailored for modeling time series, such as Long Short Term Memory (LSTM) neural networks, due to their capability of being well-suited for sequential data. In addition, we aim to deeply assess the influence of the parameters on the results, and to generalize our method for solving change detection tasks with time series data in other application domains.

ACKNOWLEDGMENT

We acknowledge the support of the Ministry of University and Research (MUR) through the project “ComESto - Community Energy Storage: Gestione Aggregata di Sistemi d’Accumulo dell’Energia in Power Cloud”, Grant No. ARS01_01259 (Community Energy Storage: Aggregate Management of Energy Storage Systems in Power Cloud) for the activities of WP6 on the application of machine learning and data mining techniques for supporting power grid planning. Dr. Gianvito Pio acknowledges the support of Ministry of University and Research (MUR) through the project “Big Data Analytics”, AIM 1852414, activity 1, line 1. Dr Roberto Corizzo acknowledges the support of the U.S. Defense Advanced Research Projects Agency (DARPA) through the project “Lifelong Streaming Anomaly Detection” (Grant No. A19-0131-003). The authors thank Claudio Alberti and Francesco Dura from e-distribuzione S.p.A. for fruitful discussion and for providing real-world datasets.

REFERENCES

[1] N. Draper and H. Smith, *Applied regression analysis*, ser. Wiley series in probability and statistics: Texts and references section. Wiley, 1998, no. v. 1.

[2] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, “Support vector method for novelty detection,” in *Advances in neural information processing systems*, 2000, pp. 582–588.

[3] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[5] N. Japkowicz, C. Myers, and M. Gluck, “A novelty detection approach to classification,” in *IJCAI*, 1995, vol. 1, pp. 518–523.

[6] R. Corizzo, M. Ceci, E. Zdravevski, and N. Japkowicz, "Scalable autoencoders for gravitational waves detection from time series data," *Expert Systems with Applications*, vol. 151, p. 113378, 2020.
[7] W. Xu, H. Hu, and W. Yang, "Energy time series forecasting based on empirical mode decomposition and frbf-ar model," *IEEE Access*, vol. 7, pp. 36 540–36 548, 2019.
[8] L. Cavalcante, R. J. Bessa, M. Reis, and J. Browell, "Lasso vector autoregression structures for very short-term wind power forecasting," *Wind Energy*, vol. 20, no. 4, pp. 657–675, 2017.
[9] R. Corizzo, G. Pio, M. Ceci, and D. Malerba, "Dencast: Distributed density-based clustering for multi-target regression." *Springer Journal of Big Data*, 2019.
[10] M. Ceci, R. Corizzo, D. Malerba, and A. Rashkovska, "Spatial autocorrelation and entropy for renewable energy forecasting," *Data Mining and Knowledge Discovery*, 2019.
[11] N. Sharma, P. Sharma, D. E. Irwin, and P. J. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *SmartGridComm*. IEEE, 2011, pp. 528–533.
[12] G. N. Bathurst, J. Weatherill, and G. Strbac, "Trading wind generation in short term energy markets," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 782–789, 2002.
[13] Chen Changsong, Duan Shanxu, Cai Tao, Liu Bangyin, and Yin Jinjun, "Energy trading model for optimal microgrid scheduling based on genetic algorithm," in *2009 IEEE 6th International Power Electronics and Motion Control Conference*, 2009, pp. 2136–2139.
[14] M. Ceci, R. Corizzo, F. Fumarola, D. Malerba, and A. Rashkovska, "Predictive modeling of pv energy production: How to set up the learning task for a better prediction?" *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2016.
[15] R. J. Bessa, A. Trindade, and V. Miranda, "Spatial-temporal solar power forecasting for smart grids," *IEEE Trans. Industrial Informatics*, vol. 11, no. 1, pp. 232–241, 2015.
[16] R. Corizzo, M. Ceci, and N. Japkowicz, "Anomaly detection and repair for accurate predictions in geo-distributed big data," *Big Data Research*, vol. 16, pp. 18–35, 2019.
[17] T. Gneiting, K. Larson, K. Westrick, M. G. Genton, and E. Aldrich, "Calibrated probabilistic forecasting at the stateline wind energy center: The regime-switching space–time method," *Journal of the American Statistical Association*, vol. 101, no. 475, pp. 968–979, 2006.
[18] R. Corizzo, M. Ceci, H. Fanaee-T, and J. Gama, "Multi-aspect renewable energy forecasting," *Information Sciences*, 2020.
[19] R. Bessa, V. Miranda, and J. Gama, "Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting," *Power Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 1657–1666, 2009.
[20] J. Gama and M. M. Gaber, Eds., *Learning from Data Streams*. Springer, 2007.
[21] G. Fenza, M. Gallo, and V. Loia, "Drift-aware methodology for anomaly detection in smart grid," *IEEE Access*, vol. 7, pp. 9645–9657, 2019.
[22] Z. Li, Q. Li, Z. Wu, J. Yu, and R. Zheng, "A fault diagnosis method for on load tap changer of aerospace power grid based on the current detection," *IEEE Access*, vol. 6, pp. 24 148–24 156, 2018.
[23] H. Karimipour, A. Dehghantanha, R. M. Parizi, K.-K. R. Choo, and H. Leung, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," *IEEE Access*, vol. 7, pp. 80 778–80 788, 2019.
[24] Y. Wang, M. M. Amin, J. Fu, and H. B. Moussa, "A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids," *IEEE Access*, vol. 5, pp. 26 022–26 033, 2017.
[25] M. M. Moya, M. W. Koch, and L. D. Hostetler, "One-class classifier networks for target recognition applications," *NASA STI/Recon Technical Report N*, vol. 93, 1993.
[26] N. Japkowicz, *Concept-learning in the absence of counter-examples: an autoassociation-based approach to classification*, 1999.
[27] D. Tax, "One-class classification; Concept-learning in the absence of counterexamples," *Ph. D thesis. Delft University of Technology, ASCI Dissertation Series, 2001*, vol. 146, 2001.
[28] G. Ritter and M. T. Gallegos, "Outliers in statistical pattern recognition and an application to automatic chromosome classification," *Pattern Recognition Letters*, vol. 18, no. 6, pp. 525–539, 1997.
[29] C. M. Bishop, "Novelty detection and neural network validation," *IEE Proceedings-Vision, Image and Signal processing*, vol. 141, no. 4, pp. 217–222, 1994.
[30] P. Mignone, G. Pio, D. D'Elia, and M. Ceci, "Exploiting transfer learning for the reconstruction of the human gene regulatory network," *Bioinformatics*, vol. 36, no. 5, pp. 1553–1561, 10 2019.
[31] C. Bellinger, S. Sharma, and N. Japkowicz, "One-class versus binary classification: Which and when?" in *2012 11th International Conference on Machine Learning and Applications*, 2012, vol. 2, pp. 102–106:.
[32] T. Kefi-Fatteh, R. Ksantini, M.-B. Kaâniche, and A. Bouhoula, "A novel incremental one-class support vector machine based on low variance direction," *Pattern Recognition*, vol. 91, pp. 308 – 321, 2019.
[33] Z. Tong, Y. Zhou, and J. Wang, "Identifying potential drug targets in hepatocellular carcinoma based on network analysis and one-class support vector machine," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.
[34] T. M. Hoang, N. M. Nguyen, and T. Q. Duong, "Detection of eavesdropping attack in uav-aided wireless systems: Unsupervised learning with one-class svm and k-means clustering," *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 139–142, 2019.
[35] F. Harrou, A. Dairi, B. Taghezouit, and Y. Sun, "An unsupervised monitoring procedure for detecting anomalies in photovoltaic systems using a one-class support vector machine," *Solar Energy*, vol. 179, pp. 48–58, 2019.
[36] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.
[37] R. Wang, F. Nie, Z. Wang, F. He, and X. Li, "Multiple features and isolation forest-based fast anomaly detector for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.
[38] S. Yu, X. Li, L. Zhao, and J. Wang, "Hyperspectral anomaly detection based on low-rank representation using local outlier factor," *IEEE Geoscience and Remote Sensing Letters*, 2020.
[39] C. Lee, S. Jo, D. Kwon, and M. Pecht, "Capacity-fading behavior analysis for early detection of unhealthy li-ion batteries," *IEEE Transactions on Industrial Electronics*, 2020.
[40] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, Oct. 2016.
[41] S. S. Khan and M. G. Madden, "A survey of recent trends in one class classification," in *Irish conference on artificial intelligence and cognitive science*, 2009, pp. 188–197:.
[42] S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," *Knowledge Eng. Review*, vol. 29, no. 3, pp. 345–374, 2014.
[43] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.
[44] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
[45] C. C. Aggarwal, "Outlier analysis," in *Data mining*, 2015, pp. 237–263:.
[46] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.
[47] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.
[48] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc, 2009.
[49] P. Sikdar and S. Chakraborty, "Genesis of arsenic in groundwater of north bengal plain using pca: a case study of english bazar block, malda district, west bengal, india," *Hydrological Processes: An International Journal*, vol. 22, no. 12, pp. 1796–1809, 2008.
[50] M. Misra, H. H. Yue, S. J. Qin, and C. Ling, "Multivariate process monitoring and fault diagnosis by multi-scale pca," *Computers & Chemical Engineering*, vol. 26, no. 9, pp. 1281–1293, 2002.
[51] R. Noori, A. Karbassi, A. Moghaddamnia, D. Han, M. Zokaei-Ashtiani, A. Farokhnia, and M. G. Gousheh, "Assessment of input variables determination on the svm model performance using pca, gamma test, and forward selection techniques for monthly stream flow prediction," *Journal of Hydrology*, vol. 401, no. 3-4, pp. 177–189, 2011.
[52] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.
[53] A. Impedovo, M. Ceci, and T. Calders, "Efficient and accurate non-exhaustive pattern-based change detection in dynamic networks," in *Discovery Science*, P. Kralj Novak, T. Šmuc, and S. Džeroski, Eds. Cham: Springer International Publishing, 2019, pp. 396–411.

MICHELANGELO CECI, Ph.D., is an Associate Professor of Computer Science, University of Bari, Italy. He published more than 150 papers in journals and conferences on Machine Learning and Data Mining. He is the unit coordinator of EU and national projects. He has been in the PC of many conferences (e.g., IEEE ICDM, SDM, IJCAI and AAAI) and in the editorial board of DMKD, MLJ, JIIS. He was PC co-chair of SEBD2007, Discovery Science 2016, ISMIS 2018 and General Chair of ECML-PKDD 2017.

ROBERTO CORIZZO, Ph.D, is an Assistant Professor of Research, Dept. of Computer Science, American University. Prior to that, he was a Research Fellow in the Department of Computer Science at University of Bari, Italy. He co-authored 16 articles, including 7 publications in journals such as IEEE Transactions on Industrial Informatics, Data Mining and Knowledge Discovery, and Information Sciences. He participated in the scientific committee of international conferences and served as a reviewer for several international journals.

NATHALIE JAPKOWICZ, Ph.D., is a Professor of Computer Science at American University. Prior to that she directed the Laboratory for Research on Machine Learning applied for Defense and Security at the University of Ottawa. She trained over 30 graduate students and collaborated with Canadian and US governmental agencies, as well as private industry. She published a co-authored book, one edited book and over 100 book chapters, journal and conference papers. She received five best paper awards, including the prestigious ECML2014 Test of Time award.

PAOLO MIGNONE , Ph.D, is a Research Fellow at the Dept. of Computer Science, University of Bari (Italy). He has published papers in top-journals including papers in Bioinformatics, IEEE Transactions on Human-Machine Systems, and conference proceedings. He participated in the scientific committee of international conferences and served as a reviewer in a wide range of international conferences and journals. His research interests are mainly focused on Data Mining and Knowledge Discovery, Bioinformatics, Transfer Learning, Link Prediction, Big Data Analytics, Cybersecurity, and Social Network Analysis.

GIANVITO PIO, Ph.D, is an Assistant Professor at the Dept. of Computer Science, University of Bari, Italy. He published 30 papers, including 13 papers in journals, such as Machine Learning, Bioinformatics, BMC Bioinformatics, PloS ONE, Information Sciences and IEEE Transactions on Knowledge and Data Engineering. He participated in the scientific committee of international conferences and served as a reviewer for several international journals. His interests include big data analytics, bioinformatics and methods to analyze heterogeneous networks.

• • •

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

ECHAD: embedding-based change detection from multivariate time series in smart grids

MICHELANGELO CECI[1,3,4], ROBERTO CORIZZO[2] (Member, IEEE), NATHALIE JAPKOWICZ[2], PAOLO MIGNONE[1,4] and GIANVITO PIO[1,4]

[1]Department of Computer Science, University of Bari Aldo Moro, Via Orabona, 4, 70125 Bari (BA), Italy (e-mail: name.surname@uniba.it)

[2]Department of Computer Science, American University, 4400 Massachusetts Ave NW, Washington, DC 20016, United States (e-mail: rcorizzo@american.edu, japkowic@american.edu)

[3]Dept. of Knowledge Technologies, Jožef Stefan Institute, Ljubljana, 1000, Slovenia

[4]Big Data Laboratory - National Interuniversity Consortium for Informatics (CINI), Roma, Italy

Corresponding author: Michelangelo Ceci (e-mail: michelancelo.ceci@uniba.it).

ABSTRACT Smart grids are power grids where clients may actively participate in energy production, storage and distribution. Smart grid management raises several challenges, including the possible changes and evolutions in terms of energy consumption and production, that must be taken into account in order to properly regulate the energy distribution. In this context, machine learning methods can be fruitfully adopted to support the analysis and to predict the behavior of smart grids, by exploiting the large amount of streaming data generated by sensor networks. In this paper, we propose a novel change detection method, called ECHAD (Embedding-based CHAnge Detection), that leverages embedding techniques, one-class learning, and a dynamic detection approach that incrementally updates the learned model to reflect the new data distribution. Our experiments show that ECHAD achieves optimal performances on synthetic data representing challenging scenarios. Moreover, a qualitative analysis of the results obtained on real data of a real power grid reveals the quality of the change detection of ECHAD. Specifically, a comparison with state-of-the-art approaches shows the ability of ECHAD in identifying additional relevant changes, not detected by competitors, avoiding false positive detections.

INDEX TERMS Change detection algorithms, smart grids, one-class learning, neural networks, embedding

I. INTRODUCTION

POWER grids are complex systems consisting of generation, transmission, and distribution infrastructures. They represent an important evolution of power grids, where clients are not necessarily passive consumers but have the opportunity to actively participating in the grid, by producing energy from renewable sources and by storing energy through batteries or alternative systems.

One of the most relevant challenges in the context of smart grids is represented by possible changes and evolutions in terms of consumption and production, also due to the influence of some uncontrollable factors. In particular, the production of energy from renewable sources is inherently characterized by instability issues due, for example, to weather conditions. This uncertainty may negatively impact the performance of analytical tools used in power grids for scheduling, planning and regulation purposes.

Additional sources of changes in power grids include variations in the power load as well as the need to adequate the infrastructure to new scenarios (e.g., the installation of car charging stations), that may cause a significant increase of the concurrent consumption of energy, as well as changes in the voltage measured on specific network components.

In this context, machine learning methods can provide significant support in analyzing, optimizing and predicting the behavior of such complex systems, by exploiting the large amount of streaming data generated by sensor networks. Moreover, being able to detect changes from streaming data related to multiple variables (i.e., multivariate time series - see Figure 1) can enable the system to provide prompt alerts that can suggest maintenance activities in a timely manner.

However, the identification of such changes and evolutions in smart grids poses three main challenges:

- sparse and isolated observed peaks should not affect the detection of changes, namely, the system should be robust to possible outliers;

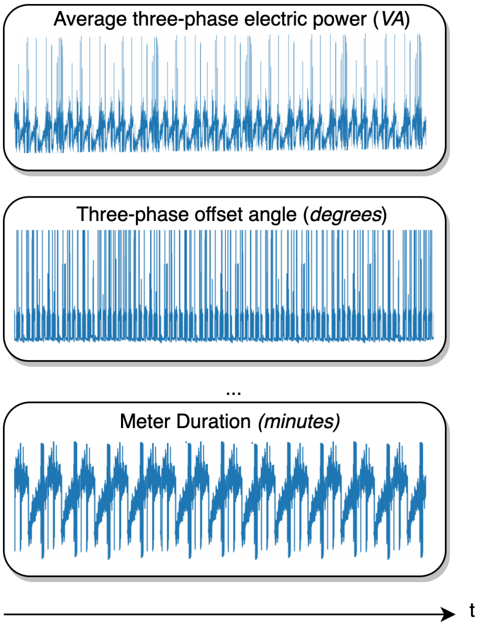

FIGURE 1. A graphical representation of multiple time series corresponding to real data possibly observed in a power grid.

- the amount of available labelled examples is very poor;
- multivariate time series, consisting of a huge number of observed variables, may introduce collinearity phenomena [1] due to possible variable correlation, that may compromise the detection accuracy.

Such challenges make the direct application of classical supervised methods unfeasible, and even semi-supervised methods may appear inadequate due to the strongly unbalancing between the amount of labelled and unlabelled examples. Moreover, in some novel real-world scenarios like that of smart grids, critical conditions or changes have rarely (sometimes never) been observed. Such a situation suggests that the most proper way to tackle this problem is the adoption of approaches able to model the standard/regular scenario and to evaluate the presence of changes according to the coherence with such a model.

In this context, approaches based on one-class learning [2]–[6] find their natural application, since the built model is fitted on one scenario (the regular one) and can subsequently be exploited to detect changes.

Following this line of research, in this paper we propose ECHAD (Embedding-based CHAnge Detection), a novel unsupervised change detection method able to analyze streaming data generated by sensors located in smart grids. ECHAD leverages embedding techniques and a one-class learning approach. The former allow us to extract a new feature space that better represents the inherently complex content of multivariate time series data for the subsequent learning task, also mitigating the collinearity phenomena by incorporating latent interactions among features. The latter (i.e., the proposed one-class learning approach) allows us to analyze data in an unsupervised manner, using only explicit knowledge of the standard/regular behavior of power grids. Finally, ECHAD adopts a novel change detection approach which identifies changes and updates the model accordingly, in order to reflect the new data distribution.

The major contribution of the work can be summarized as follows:

- An investigation of the possible benefits provided by an innovative method that synergically combines embedding techniques and a novel one-class learning approach for tackling the change detection task in multivariate time series data;
- A novel strategy to dynamically adapt the model when changes are detected, in the presence of a concept drift;
- A comprehensive experimental evaluation of the proposed ECHAD, including its parametrization;
- Empirical comparison with state-of-the-art methods on both synthetic and real-world datasets related to power grids.

The rest of the paper is organized as follows. In Section II we discuss the work related to this paper, from both application and methodological point of views; in Section III we describe in detail our proposed method ECHAD; in Section IV, we describe the results obtained on both synthetic and real-world datasets, showing the competitiveness of ECHAD with respect to state-of-the-art methods; finally, in Section V, we draw some conclusions regarding the applicability of ECHAD as a powerful tool in analytical tasks for smart grids, and outline possible future works.

II. BACKGROUND

Several machine learning approaches have been proposed in the literature to support analytical tasks in the energy field. Among them, significant efforts have been devoted to the forecasting of the energy produced by plants in smart grids [7]–[11]. Solving this task is particularly important to support grid power balancing, especially when the energy is produced by renewable sources. At the same time, accurate predictions of the energy produced at a specific time horizon may be useful for other scenarios, such as the optimization of energy trading operations [12], [13].

Recently, research activities have been directed towards approaches for the simultaneous forecasting of the energy produced in multiple plants, mainly exploiting time series analysis [7], autoregressive (AR) models [8], predictive clustering models [9], artificial neural networks (ANNs) [10], or SVM classifiers [11]. Recent studies [10], [14]–[18] have also investigated the possible exploitation of spatial and temporal autocorrelation phenomena to improve forecasting accuracy. For example, in [17], the authors exploit geo-distributed weather observations in the neighborhood of wind plants, while in [14], the authors extract statistical indicators that model the spatio-temporal autocorrelation between plants for each descriptive feature.

The common aspect among these solutions consists of the possible exploitation of additional factors, including temporal and spatial closeness among multiple plants, as well as external uncontrollable factors (e.g., measured or predicted weather conditions). The main motivations for taking into account these additional factors come from the possible simultaneous changes and evolutions of the behavior that can be observed in plants working in similar conditions (e.g., spatially closed and subject to similar weather conditions). These motivations also justify the need to detect and model changes in the distribution of some variables (also known as *concept drift* [19] [20] in the literature), that may be fruitfully exploited to timely predict changes in similar/related plants.

The focus of the present paper is specifically in this area of research. In particular, we propose a method to detect changes in time series, possibly coming from sensors. As introduced in Section I, our approach works in an unsupervised setting and models the standard/regular scenario to properly detect changes in the data distribution.

In this context, existing methods mostly rely on the one-class learning setting. Alternative methods are based on Long-Short Term Memory neural networks [21], Empirical Mode Decomposition [22], Symbolic Dynamic Filtering [23] and the Margin Setting Algorithm [24], although they focus primarily on the detection of anomalies and attacks in the smart grid, rather than generic changes.

One-class learning was first proposed in [25] and subsequently studied in [26] and [27]. Differently than binary (or multi-class) classification approaches, that learn to discriminate between positive and negative examples (or among multiple classes), one-class learning methods focus on modeling one single class of examples and identify whether unseen examples belong to the learned class or not. A similar rationale has found application also in outlier detection [28], novelty detection [29] and positive-unlabelled learning [30] approaches.

It is important to mention that, in classical supervised settings, standard (binary or multi-class) classification methods easily outperform one-class learning approaches in discriminating among the possible classes [31]. However, there are specific scenarios in which one-class learning approaches are the most appropriate, or even the only applicable solutions. Such scenarios include situations in which:

- Only one class of instances is actually known, while other possible classes are not known a-priori.
- The dataset at hand is strongly unbalanced. In this case, most standard approaches may be biased towards the majority class.
- The goal is explicitly to detect rare, particular situations.

When the task under consideration is the detection of changes, as in the specific application domain considered in this paper, the stable/regular situation is strongly over-represented in the available dataset, and only a small fraction of instances representing the changes is actually available. This aspect confirms that the adoption of a one-class learning approach is the most suitable solution.

In the literature, several one-class classification methods have been proposed. Among them, it is worth mentioning One-Class SVM [2], [32]–[35], Isolation Forest [3], [36], [37], One-Class Local Outlier Factor (LOF) [4], [38], [39] and approaches based on autoencoders [5], [6].

One-Class SVM [2], [32] is an unsupervised learning algorithm that learns a decision function for the detection of changes. One-Class SVM learns such a function exclusively from instances of a single class and classifies new instances as similar or different to the training set. One recognized limitation of this approach is the possibility to deal with high-dimensional data, and its sensitivity to the presence of outliers in training data [40].

LOF [4] measures the local density deviation of a given instance concerning its neighbors. In particular, the LOF score of an instance is computed as the ratio of the average local density of its k-nearest neighbors, and its own local density. Instances appearing similar to the training data distribution are expected to exhibit a local density similar to that of its neighbors. On the contrary, instances representing a change in the distribution are expected to show a much smaller local density. However, when this locality property is not present/satisfied in the application domain at hand, the performance of these methods may be compromised.

Isolation Forest [3] is a tree-based method that isolates instances that appear different than the training data distribution. The algorithm recursively partitions the sample of instances by randomly selecting a feature and a split value. Instances that require a small number of splits to be isolated in leaf nodes are more likely to represent changes or outliers with respect to the training data distribution. Isolation Forest shows a low time complexity and low memory requirements. However, high-dimensional data may affect their detection performance. Moreover, its decision boundaries are limited to vertical and horizontal shapes.

Approaches based on autoencoders learn the data distribution of one-class data through special kinds of neural networks. Subsequently, the learned distribution is exploited to determine whether new instances belong to the same known distribution or differ from it significantly. They offer the opportunity to learn non-linear relationships in the data, by exploiting non-linear activation functions in the hidden layers. However, existing approaches [5], [6], [16] are mostly focused on identifying point anomalies, and do not address the problem of identifying changes in the data distribution, which is the main focus of this study.

Relevant surveys presenting one-class classification methods include [41], [42], and [43], whereas surveys discussing anomaly or outlier detection methods that include one-class classification can be found in [44] and [45].

Compared with such existing approaches, ECHAD has the advantage to properly deal with possible collinearity issues of multi-variate time series, thanks to the embedding approach adopted, and to the capability of dynamically adapting the model when changes are detected, in the presence of a concept drift.

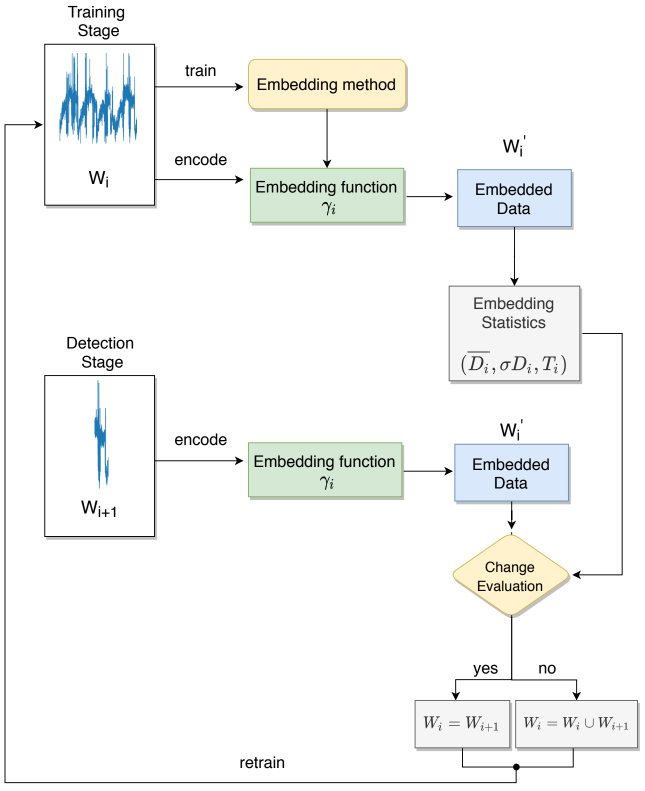

FIGURE 2. A graphical representation of the proposed method implemented in ECHAD

III. THE METHOD ECHAD

In this section, we describe our novel approach ECHAD, an embedding-based change detection algorithm that is able to detect changes in time series data generated by smart grids. We stress that the peculiarity of our approach is the combination of an embedding solution with a one-class learning method that is able to dynamically update the learned model. These aspects allow ECHAD to analyze complex and dynamic multivariate time series and to identify changes in the data distribution, leveraging exclusively the knowledge of the standard/regular behavior of the smart grid.

ECHAD consists of two main phases, namely *i)* learning an embedding model from historical time series data falling into a specific interval (time window); *ii)* detecting changes on newly observed data, using a streaming test-and-retrain workflow. A graphical overview of the general workflow followed by ECHAD is depicted in Figure 2, while in the following subsections, we explain its main phases in detail.

A. LEARNING EMBEDDING MODELS

Let W_i be a time window consisting of M time series, corresponding to M features measured over N_i time points. In this phase, we learn a reduced, latent, K-dimensional feature space, with $K \ll M$. More formally, the time series data of the time window can be represented as a matrix $W_i \in \mathbb{R}^{N_i \times M}$, and the goal is to learn a function γ_i: $\mathbb{R}^M \rightarrow \mathbb{R}^K$, that maps each M-dimensional time point of a time series to the reduced, K-dimensional feature space. The function, although learned from the time window W_i, can naturally be applied to other, also unseen, time points in order to project their features into the reduced feature space.

To perform this step, any approach to identify a reduced feature space can be plugged into our system. In this paper, we consider the classical Principal Component Analysis (PCA) [46] and the more recent Stacked Auto-encoders [47], [48], for which we provide some details in the following.

PCA is one of the most popularly known dimensionality reduction technique, which effectiveness has been shown in several scientific fields, ranging from chemistry to geology [49]–[51]. Specifically, PCA estimates the correlation among the variables and extracts a reduced set of features that are as much as (linearly) uncorrelated as possible. This transformation is performed such that each extracted feature, called principal component, explains the largest possible amount of data variance, with the constraint of being orthogonal to all the previously extracted features. In this way, PCA extracts a reduced representation of the data, that explains a given overall percentage of data variance, possibly discarding the noise. Formally, given the input matrix $W_i \in \mathbb{R}^{N \times M}$, PCA computes the covariance matrix $C \in \mathbb{R}^{M \times M}$, from which it extracts the first K eigenvectors, associated to the largest eigenvalues, obtaining the matrix $Z_i \in \mathbb{R}^{M \times K}$. The matrix Z_i can finally be used to compute the embedding of a new time point $w \in \mathbb{R}^M$ as follows:

$$\gamma_i(w) = Z_i^T \cdot w \tag{1}$$

While PCA properly deals with collinearity problems thanks to the orthogonality of the extracted features, its main limitation is in its ability to catch only linear dependencies among variables. Such a limitation also defines one of the strong points of an alternative approach that has recently been proposed in the literature, namely stacked auto-encoders. They are special kinds of neural networks, whose main purpose is to reconstruct a given data distribution with the lowest possible reconstruction error. The dimensionality reduction is achieved by exploiting bottleneck features extracted at their hidden layers [52].

Thanks to the stacked structure, each layer represents data at a different abstraction level. For example, in the domain of images, the first layer may represent edges, while deeper levels may represent contours or corners of objects.

More formally, an auto-encoder aims at learning two functions, namely the encoding function $e : \mathbb{R}^M \rightarrow \mathbb{R}^K$ and the decoding function $d : \mathbb{R}^K \rightarrow \mathbb{R}^M$, such that:

$$\langle e_i(\cdot), d_i(\cdot) \rangle = \arg \min_{\langle e_i(\cdot), d_i(\cdot) \rangle} \|W_i - d_i(e_i(W_i))\|^2 \tag{2}$$

It is noteworthy that the encoding function e can be directly used as an embedding function for new time points $w \in \mathbb{R}^M$, namely:

$$\gamma_i(w) = e_i(w) \tag{3}$$

As previously mentioned, auto-encoders are potentially able to catch non-linear relationships among features. This is achievable by adopting non-linear functions as activation functions in their hidden layer. On the other hand, there is no guarantee on the orthogonality of the extracted features, since they are identified on the basis of the reconstruction error. Therefore, collinearity issues may still be present in the reduced space.

B. CHANGE DETECTION

Let $W_i \in \mathbb{R}^{N_i \times M}$ be the time window currently designated to train the model. After exploiting it to learn the embedding function γ_i, we compute $W_i' \in \mathbb{R}^{N_i \times K}$ by applying γ_i to each time series in W_i. More formally:

$$W_i' = \Gamma_i(W_i) = \begin{bmatrix} \gamma_i(W_i[1,*]) \\ \gamma_i(W_i[2,*]) \\ \dots \\ \gamma_i(W_i[N_i,*]) \end{bmatrix} \quad (4)$$

where $W_i[j,*]$ represents the whole j-th row (i.e., the j-th time series) of the matrix W_i. Intuitively, $\Gamma_i(\cdot)$ represents the learned model valid at time i.

Then, we use W_i' to compute $\overline{D_i}$ and σD_i, which are the mean and the standard deviation, respectively, of the Euclidean distance between an instance and its p nearest neighbors, in the reduced K-dimensional feature space. Formally, let: $x \in R^K$ be a training instance belonging to N_i and $Neigh(x)$ be the set of the p nearest neighbors of x. Then:

$$\overline{D_i} = \frac{1}{p} \sum_{q \in Neigh(x)} eucl\_dist(x,q)$$

$$\sigma D_i = \sqrt{\frac{1}{p} \sum_{q \in Neigh(x)} \left(eucl\_dist(x,q) - \overline{D_i}\right)^2} \quad (5)$$

where $eucl\_dist(a,b)$ is the Euclidean distance between a and b. $\overline{D_i}$ and σD_i allow us to estimate the data distribution, and to define a threshold T_i that is exploited to detect if future observations deviate significantly from the current data distribution. The threshold T_i is calculated using a τ-sigma rule as follows:

$$T_i = \tau \cdot \sigma D_i \quad (6)$$

where τ is a user-defined parameter.

When new data arrive, belonging to a new time window $W_{i+1} \in \mathbb{R}^{N_{i+1} \times M}$, we compute W_{i+1}' by exploiting the previously learned embedding function. Formally, following Equation (4), we compute W_{i+1}' as $W_{i+1}' = \Gamma_i(W_{i+1})$.

Using W_{i+1}', for each time series (i.e., row of the matrix) $w \in W_{i+1}'$ [1], we compute D_{i+1}^w that is the mean of the Euclidean distance between w and its p-nearest neighbors. Using such a measure, we consider an instance w as a change (or not) when the following function is 1 (or 0):

$$c(w) = \begin{cases} 1 & \text{if } (D_{i+1}^w > \overline{D_i} + T_i) \vee (D_{i+1}^w < \overline{D_i} - T_i) \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

[1] Here w is a time series of the window, namely a row of the matrix.

It is noteworthy that such a change is defined at an instance level. Considering it as the final output of our detection approach would lead to being highly sensitive to outliers and spurious peaks. To overcome this issue, we work at the level of time window, and consider it, i.e. the window, as a change if more than a given ratio c_r of instances are detected as a change. Formally:

$$C(W_{i+1}) = \begin{cases} 1 & \text{if } \left(\sum_{w \in W_{i+1}} c(w)\right) > (N_{i+1} \cdot c_r) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

Independently of the output of $C(W_{i+1})$, ECHAD adapts the model representing the data distribution of regular scenarios, leading to a new threshold T_{i+1}. In particular, if a change is not detected, the embedding function is updated considering a merged time window $W_i \cup W_{i+1}$. Note that, following the mixed windows model [53], W_i can be either the single window preceding W_{i+1}, or a wider window obtained by merging multiple previous windows, when no change was detected (see Figure 3). On the contrary, if a change is detected, previous windows are discarded and the embedding function is re-learned from scratch only from W_{i+1}.

This strategy allows ECHAD to simultaneously be robust to the presence of outliers and to properly adapt to new data distributions for proper detection of subsequent changes.

A final remark regards the computational complexity of ECHAD. This can be easily computed by summing up the complexity of the embedding phase ($O(N_i \cdot M^2 + M^3)$ for PCA and $O(N_i \cdot M^2)$ for autoencoders), the complexity of identifying the p neighbors for each instance ($O(N_i \cdot logN_i)$, assuming to use a tree-based structure), and the complexity of computing $\overline{D_i}, \sigma D_i$ and T_i ($O(p \cdot K \cdot N_i)$, according to Equation 5). Therefore, the overall time complexity of ECHAD is $O(N_i \cdot M^2 + M^3 + N_i \cdot logN_i + p \cdot K \cdot N_i)$ for the variant based on PCA and $O(N_i \cdot M^2 + N_i \cdot logN_i + p \cdot K \cdot N_i)$ for the variant based on autoencoders. If we assume that $M^2 > logN_i$ and $M^2 > p \cdot K$, we have that the total time complexity is as follows: $O(N_i \cdot M^2 + M^3)$ for the variant based on PCA and $O(N_i \cdot M^2)$ for the variant based on autoencoders.

IV. EXPERIMENTS

In this section, we present the experiment for the evaluation of ECHAD. First, we introduce the adopted datasets and the experimental setting, together with the considered state-of-the-art competitor systems. Finally, we show and discuss the obtained results.

A. DATASETS

We performed experiments with five different datasets. The first four datasets are synthetically generated and represent different change detection scenarios that are relevant to power grids. The fifth dataset consists of real-time series observed in a real power grid and allows us to observe the behavior of our system in real scenarios.

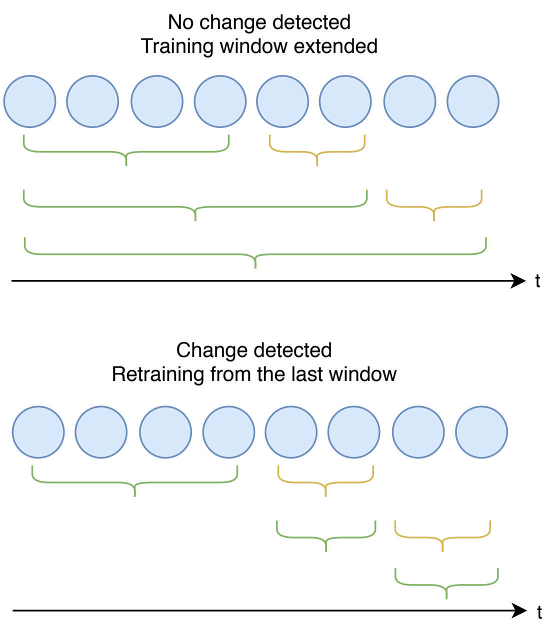

FIGURE 3. A graphical representation of the adopted mixed windows model. The schema shows the evolution of the training (green) and testing (yellow) windows over time. When no changes are detected, the training window is extended over time (see the top part of the figure). When a change is detected, the last testing window becomes the new training window (see the bottom part of the figure).

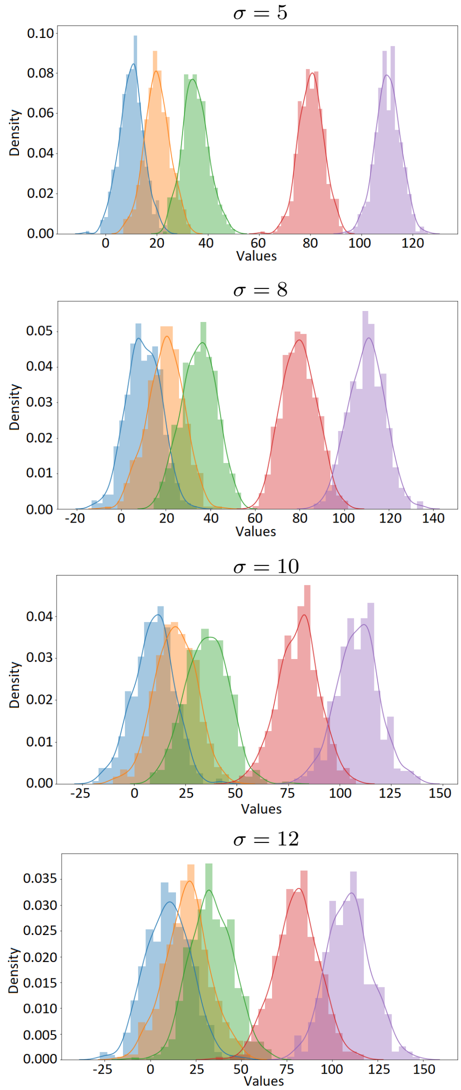

FIGURE 4. A graphical representation of the synthetic datasets, with different values of the standard deviation σ.

The synthetic datasets have been generated by considering 5 multivariate (20 variables) Gaussian distribution of $2,500$ time points, with $\mu \in \{10, 20, 35, 80, 110\}$ and a varying standard deviation $\sigma \in \{5, 8, 10, 12\}$. The four resulting datasets represent an increasing level of complexity. Indeed, datasets with a low standard deviation (i.e., $\sigma \in \{5, 8\}$) are visibly characterized by narrow Gaussian curves (see Figure 4 (a) and (b)), which potentially facilitate the change detection task due to the weak overlap among them. On the contrary, datasets with a larger standard deviation (i.e., $\sigma \in \{10, 12\}$) are characterized by wider Gaussian curves (see Figure 4 (c) and (d)), which reasonably lead to a higher difficulty in the change detection task due to the significant overlap among the Gaussian curves.

The real-world dataset has been provided in the context of the project "ComESto - Community Energy Storage" (http://www.comesto.eu/) for the Italian energy distribution network, that is managed by *e-distribuzione S.p.A*. The data consist of 200 variables measured by sensors located into medium voltage/low voltage (MV/LV) transformer rooms, related to 131,374 time points falling in the period from November 1, 2019 to December 19, 2019.

B. EXPERIMENTAL SETUP

As discussed in Section II, the most suitable class of approaches to address the task of interest in our study is that of one-class classification methods. Indeed, they offer the flexibility to learn a model from an initial (regular) data distribution and are able to flag data that significantly differ from the learned distribution. For this reason, in order to evaluate the performance obtained by ECHAD, in our experiments we considered three state-of-the-art competitor methods falling in this class, namely One-Class SVM [2], [33]–[35], Isolation Forest [3], [36], [37], and LOF [4], [38], [39], which are widely adopted in the recent literature, and are shown to provide highly accurate predictions.

After a preliminary evaluation, their parameters were set to the values suggested in their respective papers. In particular, for Isolation Forest, we set: the number of base estimators in the ensemble $n\_estimators = 100$; the contamination of the dataset, i.e., the proportion of expected outliers, $contamination = 10\%$; the number of features to draw at random for each base estimator $max\_features = M$, i.e., the whole set of features. For LOF, we set: the number of neighbors to use for k-neighbors queries $n\_neighbors = 2$; $Minkowski$ measure as distance measure. For One-Class SVM, we set: the coefficient of the Radial Basis Function

(RBF) kernel $gamma = 0.1$.

As regards ECHAD, for the autoencoder we used the LBFGS optimizer, with the objective of minimizing the reconstruction error on training data. We set the maximum number of training epochs to 500 and the minimum reduction of the training error between two subsequent epochs, used as early stopping criterion for the learning phase, equal to $10e^{-5}$. The number of hidden layers of the auto-encoder architecture is set to 3, which leads to an architecture of 5 total layers that takes input data and performs two stages of encoding and decoding. After preliminary experiments, we set other parameter values as follows:

- The number of neighbors considered for the identification of the k-nearest neighbors has been set to $p = 100$;
- The dimensionality of the reduced embedding space has been set to $K = 5$;
- A window is considered to represent a change if it contains at least 70% of data instances identified as a change, namely ($c_r = 0.7$);
- $\tau = 1$, which means that an instance is considered as a change if it differs from the mean observed in the training window of at least σD_i (see Equation (6));

For the synthetic datasets, the size of the testing window has been set to $N_{i+1} \in \{25, 50, 75\}$, while the size of the first training window has been set to the double of the size of the testing window, namely $N_1 = 50$, $N_1 = 100$ and $N_1 = 150$, respectively. For the real dataset, considering the larger amount of data points, we considered $N_{i+1} = 190$ (approximately 1.5 hours) and $N_1 = 750$ (approximately 6 hours). These ranges of values have been suggested by the domain experts, participating in the project.

For the synthetic datasets, where the ground truth is known, the performances of the considered systems have been evaluated in terms of Precision, Recall, F-Score and Accuracy, while for the real dataset, the performances have been evaluated from a qualitative point of view, involving an expert in the evaluation.

C. RESULTS AND DISCUSSION

The results on the synthetic datasets show that ECHAD combined with PCA (denoted as ECHAD-PCA) achieved the best results by catching correctly all the windows representing changes (see Figures 5 - 8). ECHAD combined with autoencoders (denoted as ECHAD-AUTOENC) returned some false detections only in the configuration presenting the largest window size (i.e., $N_1 = 150$ and $N_{i+1} = 75$). This result depends on the challenging scenario of catching a change in data with a large training window, that presents multiple heterogeneous data distributions (see Figure 4). However, in a real setting, training the model on a single data distribution is a reasonable assumption that limits the occurrence of false positives. Under this condition, the method correctly identifies changes and is re-trained using the new data distribution.

Comparing the results with those obtained by competitors, it is clear that in most cases, ECHAD outperforms them using both embedding-based models (PCA and autoencoders). Moreover, the initial intuition about the low difficulty in detecting changes on the first dataset ($\sigma = 5$) is confirmed, since it consists of four non-overlapped and narrow Gaussian curves. On the other hand, when the dataset presents overlapping and large Gaussian curves (i.e., $\sigma \in \{8, 10, 12\}$), the task became harder and induced the competitors to worse results. Going into detail, we can observe that ECHAD-PCA achieved an F1-score equal to 1.0 in all the situations, while different competitors acted differently in terms Precision and Recall. In particular, we can observe that all the methods achieved a precision of 1.0, meaning that they did not produce false positive detections, but the measured Recall is very low in some cases (see OneClass SVM and LOF).

This means that such a high precision was achieved through a strongly conservative strategy, that led to losing some relevant changes. On the contrary, such a phenomenon is not observed on the results returned by ECHAD, which led to strong results in terms of both Precision and Recall.

A comparison in terms of running times revealed that ECHAD was able to complete every single run on average in 6 minutes; OneClass SVM required on average 2.5 seconds; LOF required on average 5 seconds; Isolation Forest required on average 6.5 minutes. Although OneClass SVM and LOF show significantly lower running times than ECHAD and Isolation Forest, their results, as already shown in Figures 5-8, are much worse. Note that ECHAD simultaneously shows higher Precision, Recall, F1 Score and Accuracy, and lower running times with respect to Isolation Forest that, in these experiments, appears to be the strongest competitor.

As regards the analysis of the real dataset, a quantitative evaluation was actually not feasible, due to the lack of the ground truth. In this case, we show a graphic representation of the time windows that ECHAD correctly identified as changes, that were missed by all competitor methods. We focus on two time series that are well known to be characterized by changes in power grids, namely, *three-phase offset angle* and *three-phase total reactive power*. The first time series is considered important since a change in the offset (ideally, the offset should be close to 120 degrees) indicates that the network is not working properly. Whereas, the second time series indicates a possible phase difference between voltage and current. These results correspond to real scenarios of changes in a power grid and, therefore, are highly important to detect, in order to provide alerts and trigger timely maintenance activities.

Since ECHAD-PCA achieved the best results on synthetic datasets, we considered this variant for the analysis of real data. In Figures 9 and 10, we graphically emphasize in green the windows that appear to be correctly flagged as changes by ECHAD, that were missed by the competitors. Such situations clearly represent false negatives for the competitors.

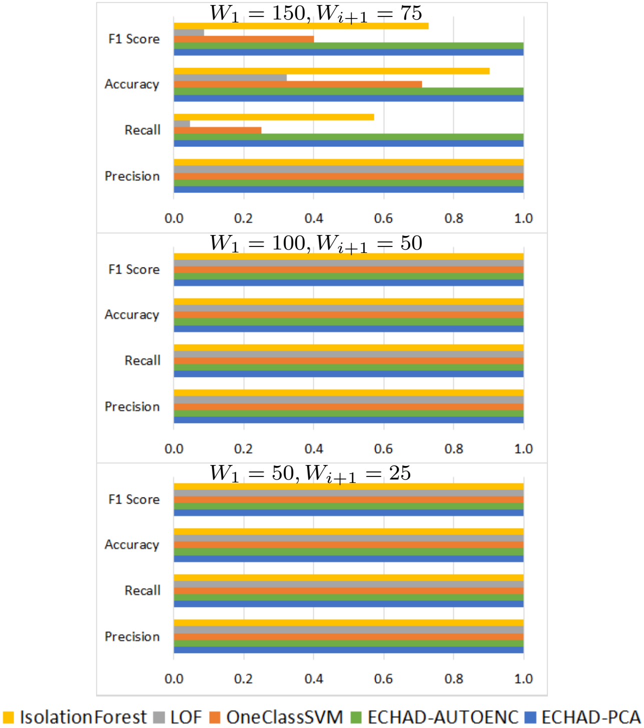

FIGURE 5. F1 Score, Accuracy, Precision, and Recall measured on the first synthetic datasets ($\sigma = 5$).

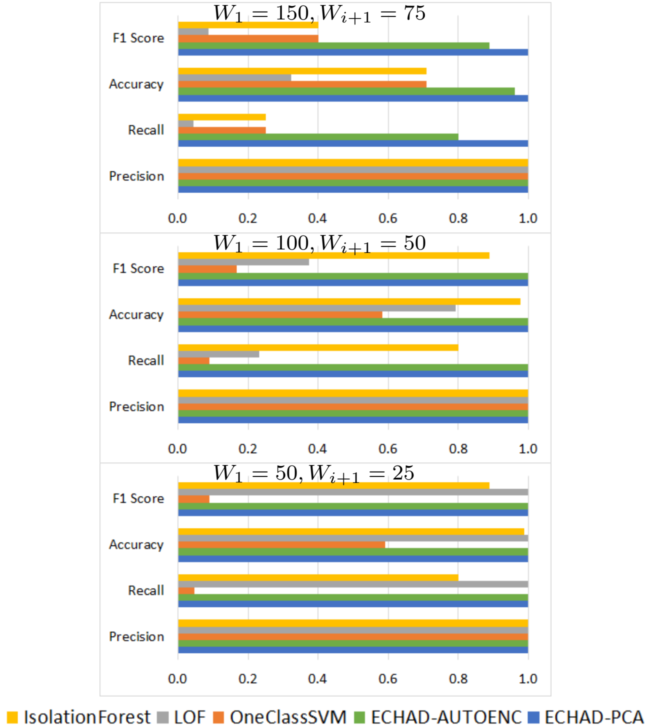

FIGURE 7. F1 Score, Accuracy, Precision, and Recall measured on the third synthetic datasets ($\sigma = 10$).

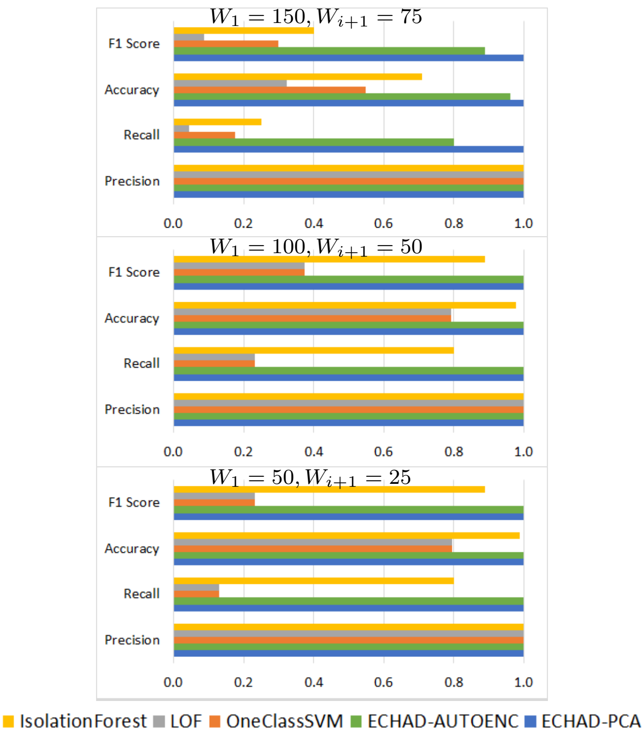

FIGURE 6. F1 Score, Accuracy, Precision, and Recall measured on the second synthetic datasets ($\sigma = 8$).

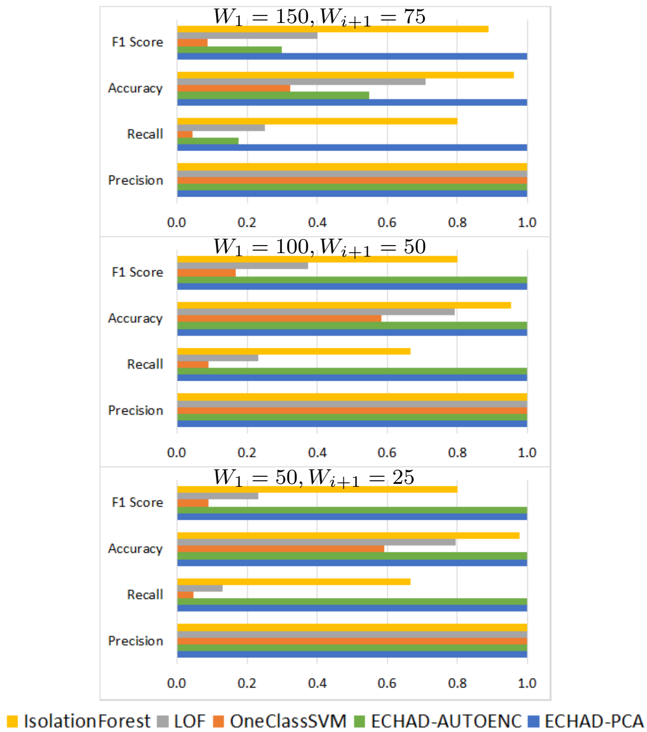

FIGURE 8. F1 Score, Accuracy, Precision, and Recall measured on the fourth synthetic datasets ($\sigma = 12$).

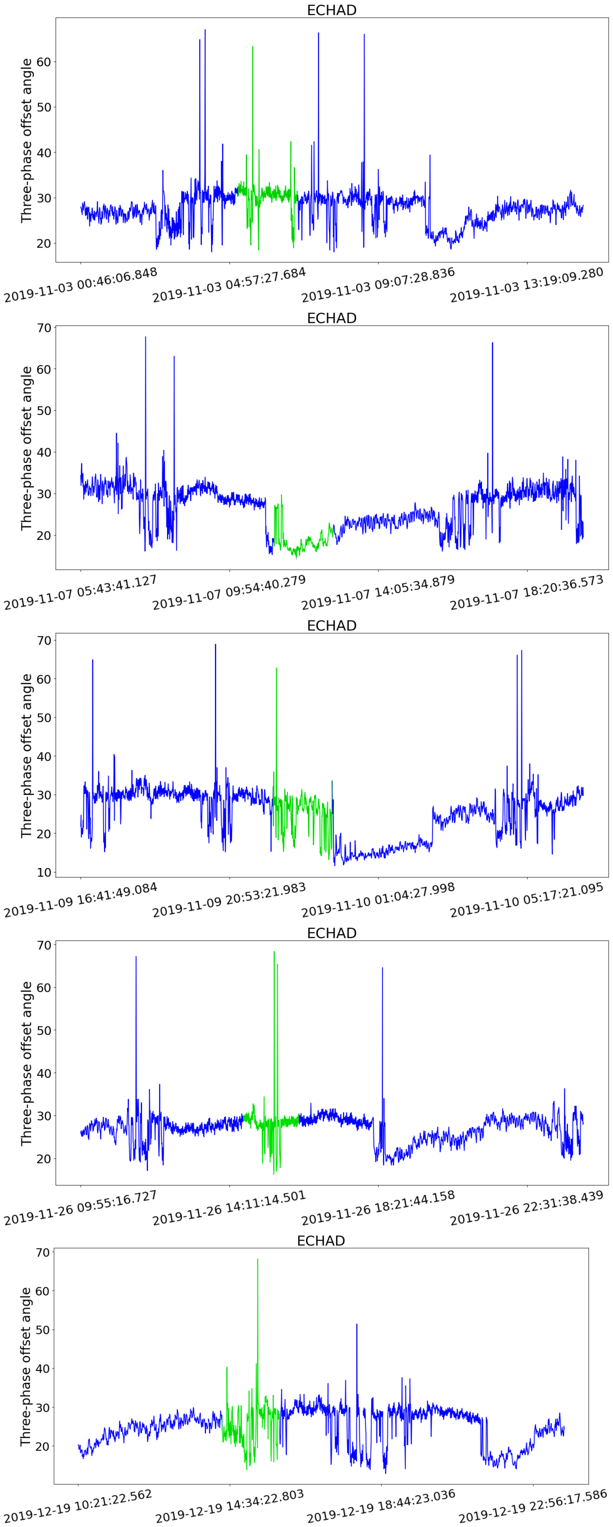

FIGURE 9. Examples of time windows representing a change (in green), identified by ECHAD but not by Isolation Forest and LOF for *three-phase offset angle*.

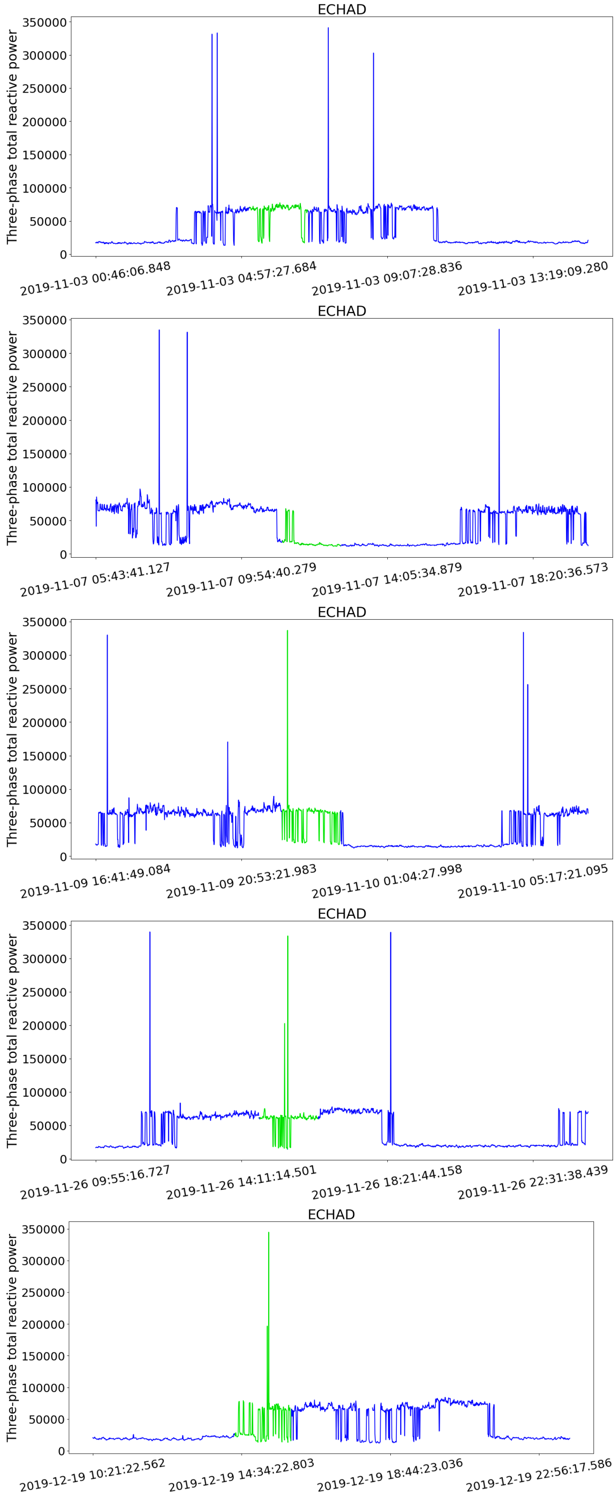

FIGURE 10. Examples of time windows representing a change (in green), identified by ECHAD but not by Isolation Forest and LOF for *three-phase total reactive power*.

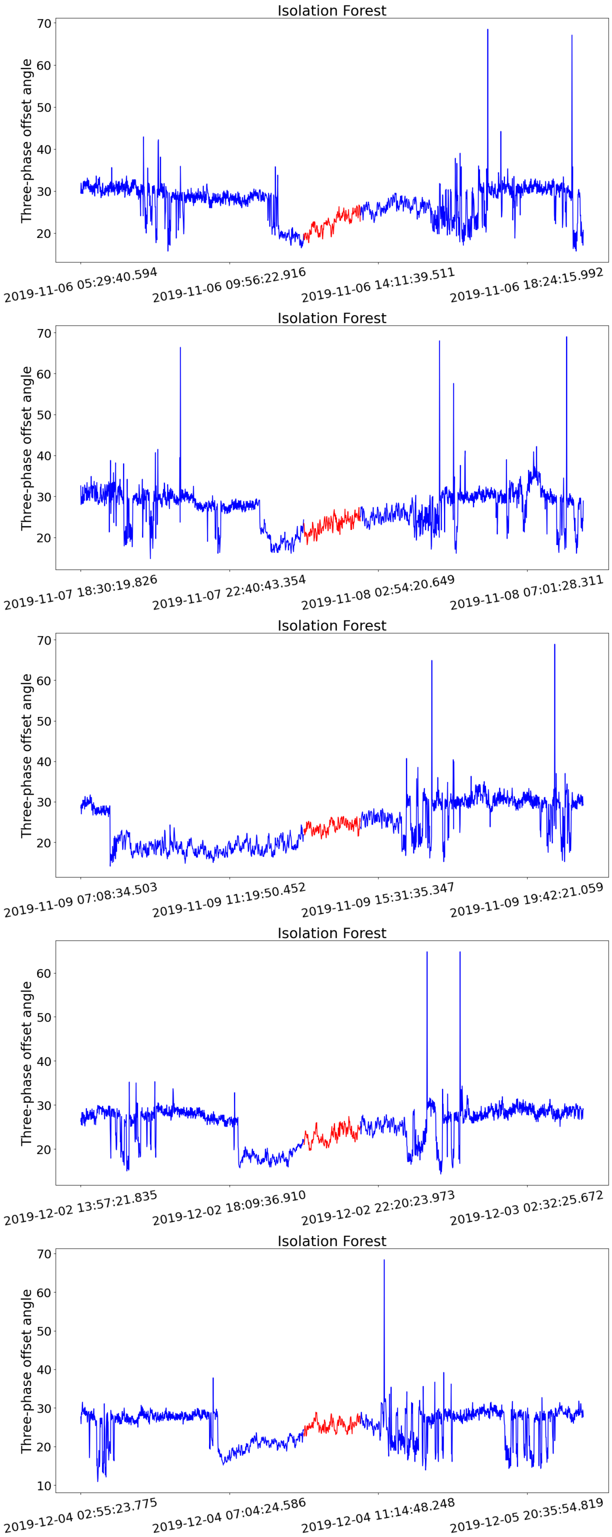

FIGURE 11. Examples of false positive detections (in red) returned by Isolation Forest, that were not returned by ECHAD, for *three-phase offset angle.*

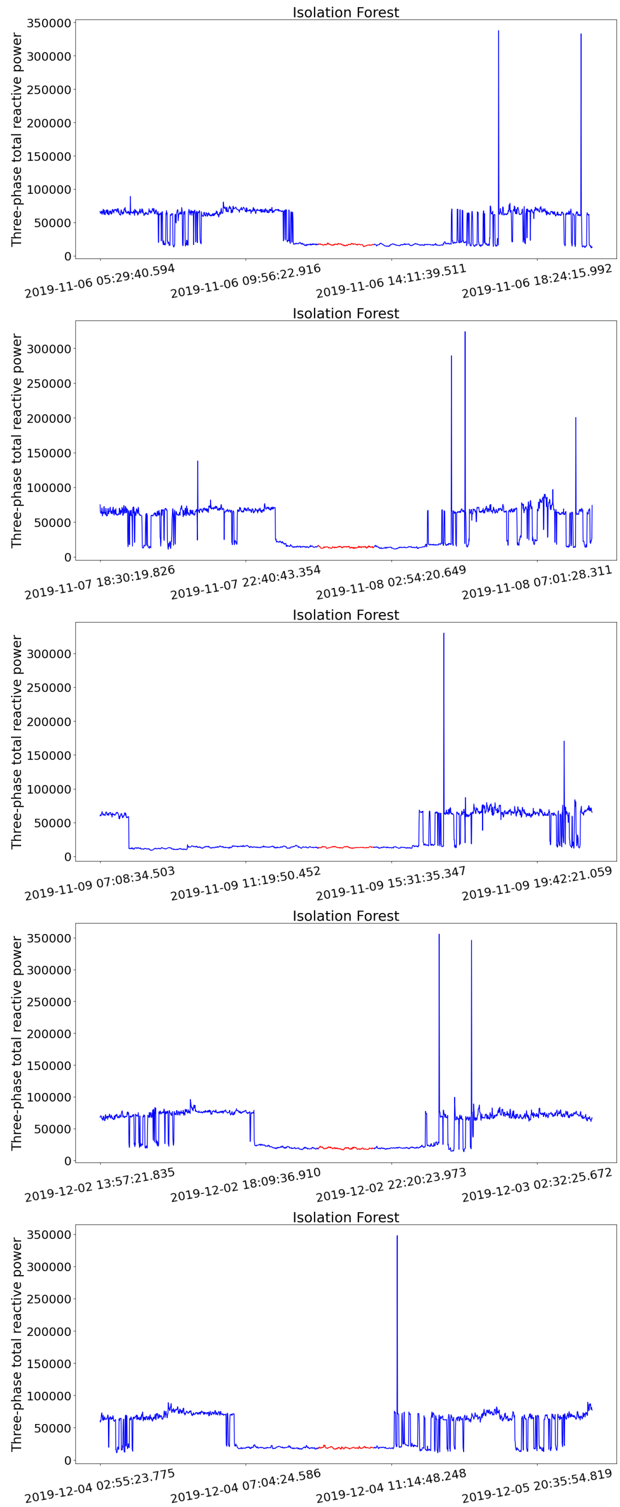

FIGURE 12. Examples of false positive detections (in red) returned by Isolation Forest, that were not returned by ECHAD, for *three-phase total reactive power*.

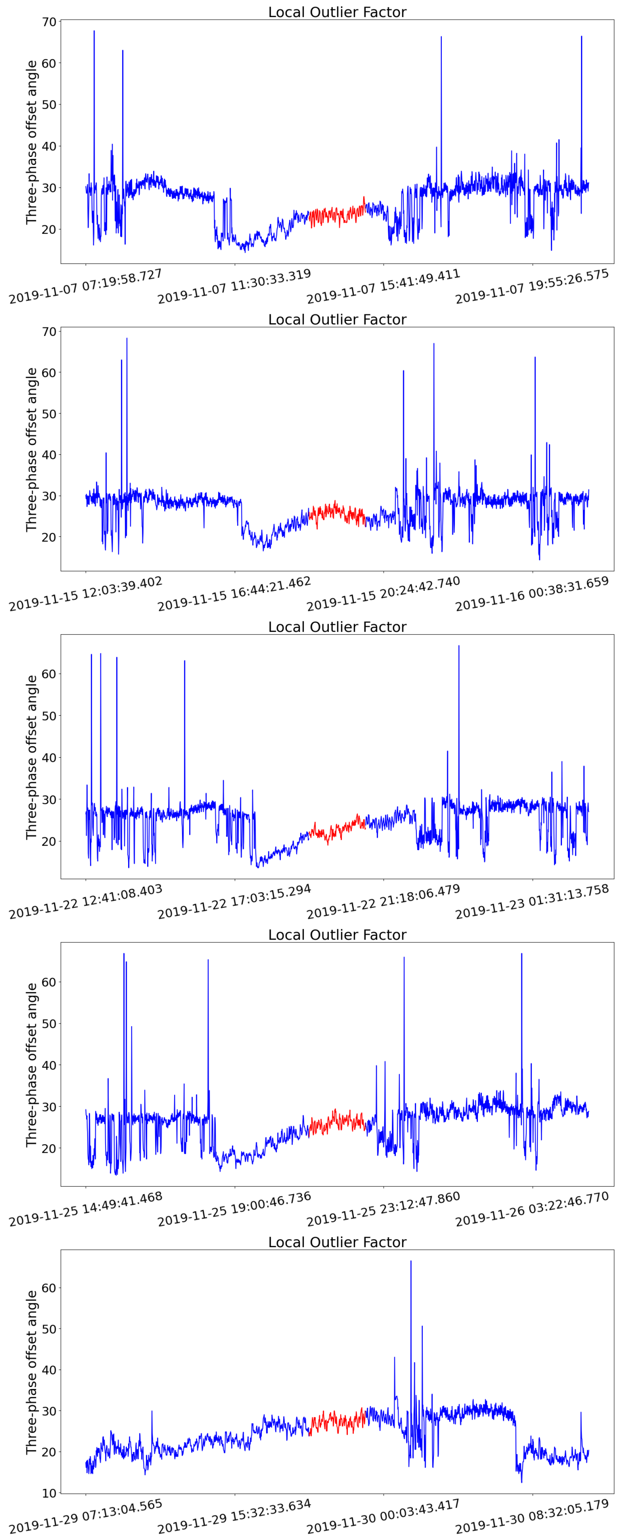

FIGURE 13. Examples of false positive detections (in red) returned by LOF, that were not returned by ECHAD, for *three-phase offset angle*.

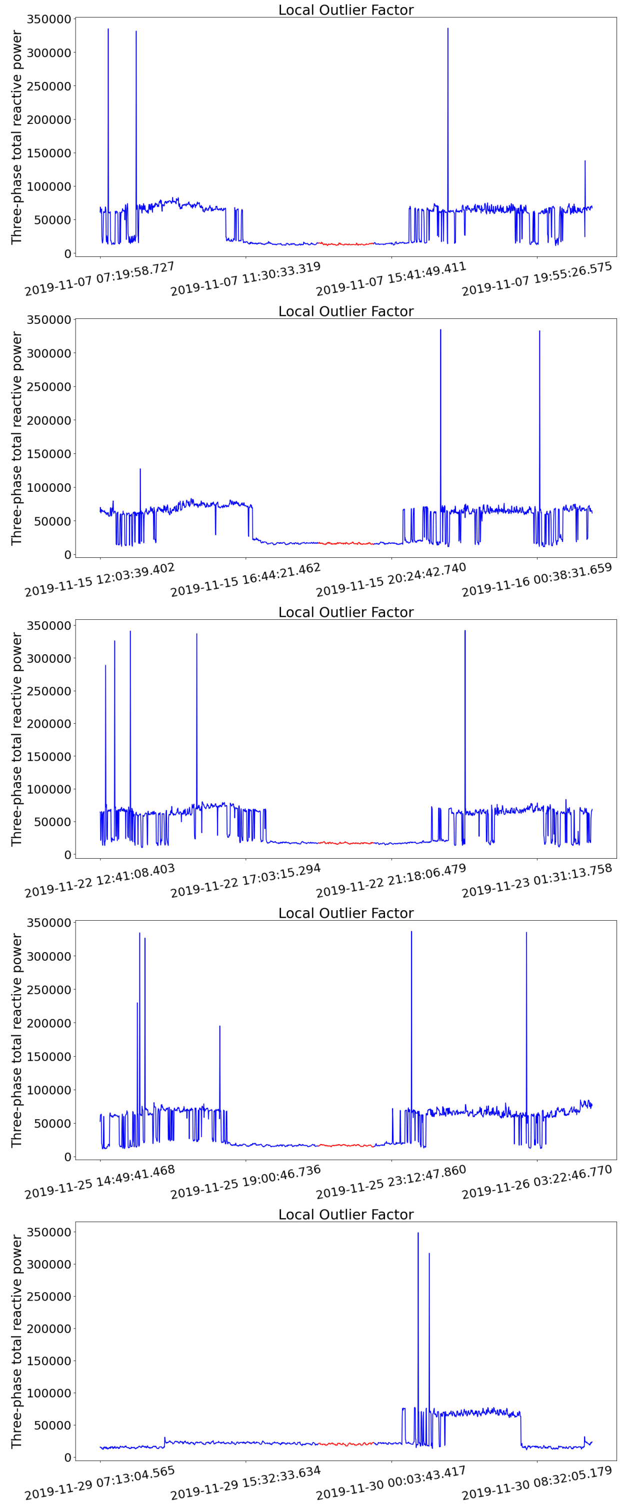

FIGURE 14. Examples of false positive detections (in red) returned by LOF, that were not returned by ECHAD, for *three-phase total reactive power*.

On the other hand, in Figure 11-12 and 13-14, we emphasize in red the time windows that appear to be incorrectly flagged as changes by Isolation Forest and LOF, respectively, that were correctly considered by ECHAD as stationary. These cases clearly represent false positives for competitor approaches. As concerns One-Class SVM, we do not report the results since it appeared to be too sensitive, with the effect of identifying many changes (and high false positive rate).

Although we cannot compute specific performance metrics on the real dataset due to the lack of ground truth, we can note that, out of a total of 687 windows, ECHAD detected 77 windows as a change, Isolation Forest detected 87 windows, LOF detected 77 windows and One-Class SVM detected all the 687 windows. Such results may lead to observe that ECHAD is more conservative than competitors. However, it is noteworthy that it also correctly identified several changes that were ignored by competitors (see Figures 9 and 10). This means that ECHAD is actually more accurate, since its robustness to false detections is not due to a generally higher conservativeness. This is also confirmed by the previous analysis of synthetic data (see Figures 5-8), where we observed that being conservative is not a sufficient condition that systematically leads to high-quality change detection.

The obtained results confirm our initial intuitions: The proposed method ECHAD, thanks to the adopted embedding approach and to the dynamic update of the model, is able to correctly detect changes in time series generated by sensors located in smart grids and to properly adapt the model according to such changes, limiting the number of false positives and providing high-quality detections that were not identified by competitors and that, according to a manual visual inspection by the experts, appear to be realistic.

A comparative analysis in terms of running times shows analogous phenomena to those observed on synthetic datasets. In particular, ECHAD required an average of 9 minutes; One-Class SVM required an average of 6 seconds; LOF required an average of 3 minutes; Isolation Forest required an average of 8.5 minutes. In conclusion, ECHAD outperformed Isolation Forest in terms of accuracy with similar running times, while the other competitors, even if showed lower running times, obtained significantly worse performances in terms of accuracy.

V. CONCLUSION

In this paper, we proposed ECHAD, a novel unsupervised change detection method able to analyze streaming data generated by sensors located in smart grids. The embedding techniques we implemented in ECHAD allow us to extract and exploit a new feature space that better represents the inherent complexity of multivariate time series, also mitigating collinearity phenomena and catching latent interactions among features. On the other hand, the proposed one-class learning approach, supported by a novel change evaluation method and a dynamic strategy to update the model, allow ECHAD to identify changes accurately.

Our experimental evaluation showed that, compared to three state-of-the-art methods, ECHAD achieves optimal change detection performance on synthetic data, also in challenging scenarios that present a high degree of overlap between evolving data distributions. Moreover, ECHAD showed high-quality results on real data observed in a real power grid. In particular, it detected several changes in the data, that were qualitatively confirmed and that were not detected by competitors. On the other hand, contrary to the competitors, ECHAD was robust to false positive detections.

As future work, we plan to integrate some existing techniques tailored for modeling time series, such as Long Short Term Memory (LSTM) neural networks, due to their capability of being well-suited for sequential data. In addition, we aim to deeply assess the influence of the parameters on the results, and to generalize our method for solving change detection tasks with time series data in other application domains.

ACKNOWLEDGMENT

We acknowledge the support of the Ministry of University and Research (MUR) through the project “ComESto - Community Energy Storage: Gestione Aggregata di Sistemi d’Accumulo dell’Energia in Power Cloud”, Grant No. ARS01_01259 (Community Energy Storage: Aggregate Management of Energy Storage Systems in Power Cloud) for the activities of WP6 on the application of machine learning and data mining techniques for supporting power grid planning. Dr. Gianvito Pio acknowledges the support of Ministry of University and Research (MUR) through the project “Big Data Analytics”, AIM 1852414, activity 1, line 1. Dr Roberto Corizzo acknowledges the support of the U.S. Defense Advanced Research Projects Agency (DARPA) through the project “Lifelong Streaming Anomaly Detection” (Grant No. A19-0131-003). The authors thank Claudio Alberti and Francesco Dura from e-distribuzione S.p.A. for fruitful discussion and for providing real-world datasets.

REFERENCES

[1] N. Draper and H. Smith, *Applied regression analysis*, ser. Wiley series in probability and statistics: Texts and references section. Wiley, 1998, no. v. 1.

[2] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, “Support vector method for novelty detection,” in *Advances in neural information processing systems*, 2000, pp. 582–588.

[3] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 413–422.

[4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “Lof: identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[5] N. Japkowicz, C. Myers, and M. Gluck, “A novelty detection approach to classification,” in *IJCAI*, 1995, vol. 1, pp. 518–523.

[6] R. Corizzo, M. Ceci, E. Zdravevski, and N. Japkowicz, “Scalable auto-encoders for gravitational waves detection from time series data,” *Expert Systems with Applications*, vol. 151, p. 113378, 2020.

[7] W. Xu, H. Hu, and W. Yang, “Energy time series forecasting based on empirical mode decomposition and frbf-ar model,” *IEEE Access*, vol. 7, pp. 36 540–36 548, 2019.

[8] L. Cavalcante, R. J. Bessa, M. Reis, and J. Browell, “Lasso vector autoregression structures for very short-term wind power forecasting,” *Wind Energy*, vol. 20, no. 4, pp. 657–675, 2017.

[9] R. Corizzo, G. Pio, M. Ceci, and D. Malerba, "Dencast: Distributed density-based clustering for multi-target regression." *Springer Journal of Big Data*, 2019.
[10] M. Ceci, R. Corizzo, D. Malerba, and A. Rashkovska, "Spatial autocorrelation and entropy for renewable energy forecasting," *Data Mining and Knowledge Discovery*, 2019.
[11] N. Sharma, P. Sharma, D. E. Irwin, and P. J. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *SmartGridComm*. IEEE, 2011, pp. 528–533.
[12] G. N. Bathurst, J. Weatherill, and G. Strbac, "Trading wind generation in short term energy markets," *IEEE Transactions on Power Systems*, vol. 17, no. 3, pp. 782–789, 2002.
[13] Chen Changsong, Duan Shanxu, Cai Tao, Liu Bangyin, and Yin Jinjun, "Energy trading model for optimal microgrid scheduling based on genetic algorithm," in *2009 IEEE 6th International Power Electronics and Motion Control Conference*, 2009, pp. 2136–2139.
[14] M. Ceci, R. Corizzo, F. Fumarola, D. Malerba, and A. Rashkovska, "Predictive modeling of pv energy production: How to set up the learning task for a better prediction?" *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2016.
[15] R. J. Bessa, A. Trindade, and V. Miranda, "Spatial-temporal solar power forecasting for smart grids," *IEEE Trans. Industrial Informatics*, vol. 11, no. 1, pp. 232–241, 2015.
[16] R. Corizzo, M. Ceci, and N. Japkowicz, "Anomaly detection and repair for accurate predictions in geo-distributed big data," *Big Data Research*, vol. 16, pp. 18–35, 2019.
[17] T. Gneiting, K. Larson, K. Westrick, M. G. Genton, and E. Aldrich, "Calibrated probabilistic forecasting at the stateline wind energy center: The regime-switching space–time method," *Journal of the American Statistical Association*, vol. 101, no. 475, pp. 968–979, 2006.
[18] R. Corizzo, M. Ceci, H. Fanaee-T, and J. Gama, "Multi-aspect renewable energy forecasting," *Information Sciences*, 2020.
[19] R. Bessa, V. Miranda, and J. Gama, "Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting," *Power Systems, IEEE Transactions on*, vol. 24, no. 4, pp. 1657–1666, 2009.
[20] J. Gama and M. M. Gaber, Eds., *Learning from Data Streams*. Springer, 2007.
[21] G. Fenza, M. Gallo, and V. Loia, "Drift-aware methodology for anomaly detection in smart grid," *IEEE Access*, vol. 7, pp. 9645–9657, 2019.
[22] Z. Li, Q. Li, Z. Wu, J. Yu, and R. Zheng, "A fault diagnosis method for on load tap changer of aerospace power grid based on the current detection," *IEEE Access*, vol. 6, pp. 24 148–24 156, 2018.
[23] H. Karimipour, A. Dehghantanha, R. M. Parizi, K.-K. R. Choo, and H. Leung, "A deep and scalable unsupervised machine learning system for cyber-attack detection in large-scale smart grids," *IEEE Access*, vol. 7, pp. 80 778–80 788, 2019.
[24] Y. Wang, M. M. Amin, J. Fu, and H. B. Moussa, "A novel data analytical approach for false data injection cyber-physical attack mitigation in smart grids," *IEEE Access*, vol. 5, pp. 26 022–26 033, 2017.
[25] M. M. Moya, M. W. Koch, and L. D. Hostetler, "One-class classifier networks for target recognition applications," *NASA STI/Recon Technical Report N*, vol. 93, 1993.
[26] N. Japkowicz, *Concept-learning in the absence of counter-examples: an autoassociation-based approach to classification*, 1999.
[27] D. Tax, "One-class classification; Concept-learning in the absence of counterexamples," *Ph. D thesis. Delft University of Technology, ASCI Dissertation Series, 2001*, vol. 146, 2001.
[28] G. Ritter and M. T. Gallegos, "Outliers in statistical pattern recognition and an application to automatic chromosome classification," *Pattern Recognition Letters*, vol. 18, no. 6, pp. 525–539, 1997.
[29] C. M. Bishop, "Novelty detection and neural network validation," *IEE Proceedings-Vision, Image and Signal processing*, vol. 141, no. 4, pp. 217–222, 1994.
[30] P. Mignone, G. Pio, D. D'Elia, and M. Ceci, "Exploiting transfer learning for the reconstruction of the human gene regulatory network," *Bioinformatics*, vol. 36, no. 5, pp. 1553–1561, 10 2019.
[31] C. Bellinger, S. Sharma, and N. Japkowicz, "One-class versus binary classification: Which and when?" in *2012 11th International Conference on Machine Learning and Applications*, 2012, vol. 2, pp. 102–106:.
[32] T. Kefi-Fatteh, R. Ksantini, M.-B. Kaâniche, and A. Bouhoula, "A novel incremental one-class support vector machine based on low variance direction," *Pattern Recognition*, vol. 91, pp. 308 – 321, 2019.
[33] Z. Tong, Y. Zhou, and J. Wang, "Identifying potential drug targets in hepatocellular carcinoma based on network analysis and one-class support vector machine," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019.
[34] T. M. Hoang, N. M. Nguyen, and T. Q. Duong, "Detection of eavesdropping attack in uav-aided wireless systems: Unsupervised learning with one-class svm and k-means clustering," *IEEE Wireless Communications Letters*, vol. 9, no. 2, pp. 139–142, 2019.
[35] F. Harrou, A. Dairi, B. Taghezouit, and Y. Sun, "An unsupervised monitoring procedure for detecting anomalies in photovoltaic systems using a one-class support vector machine," *Solar Energy*, vol. 179, pp. 48–58, 2019.
[36] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation-based anomaly detection," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 6, no. 1, pp. 1–39, 2012.
[37] R. Wang, F. Nie, Z. Wang, F. He, and X. Li, "Multiple features and isolation forest-based fast anomaly detector for hyperspectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, 2020.
[38] S. Yu, X. Li, L. Zhao, and J. Wang, "Hyperspectral anomaly detection based on low-rank representation using local outlier factor," *IEEE Geoscience and Remote Sensing Letters*, 2020.
[39] C. Lee, S. Jo, D. Kwon, and M. Pecht, "Capacity-fading behavior analysis for early detection of unhealthy li-ion batteries," *IEEE Transactions on Industrial Electronics*, 2020.
[40] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, Oct. 2016.
[41] S. S. Khan and M. G. Madden, "A survey of recent trends in one class classification," in *Irish conference on artificial intelligence and cognitive science*, 2009, pp. 188–197:.
[42] S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," *Knowledge Eng. Review*, vol. 29, no. 3, pp. 345–374, 2014.
[43] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.
[44] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
[45] C. C. Aggarwal, "Outlier analysis," in *Data mining*, 2015, pp. 237–263:.
[46] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.
[47] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research*, vol. 11, no. 12, 2010.
[48] Y. Bengio, *Learning deep architectures for AI*. Now Publishers Inc, 2009.
[49] P. Sikdar and S. Chakraborty, "Genesis of arsenic in groundwater of north bengal plain using pca: a case study of english bazar block, malda district, west bengal, india," *Hydrological Processes: An International Journal*, vol. 22, no. 12, pp. 1796–1809, 2008.
[50] M. Misra, H. H. Yue, S. J. Qin, and C. Ling, "Multivariate process monitoring and fault diagnosis by multi-scale pca," *Computers & Chemical Engineering*, vol. 26, no. 9, pp. 1281–1293, 2002.
[51] R. Noori, A. Karbassi, A. Moghaddamnia, D. Han, M. Zokaei-Ashtiani, A. Farokhnia, and M. G. Gousheh, "Assessment of input variables determination on the svm model performance using pca, gamma test, and forward selection techniques for monthly stream flow prediction," *Journal of Hydrology*, vol. 401, no. 3-4, pp. 177–189, 2011.
[52] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.
[53] A. Impedovo, M. Ceci, and T. Calders, "Efficient and accurate non-exhaustive pattern-based change detection in dynamic networks," in *Discovery Science*, P. Kralj Novak, T. Šmuc, and S. Džeroski, Eds. Cham: Springer International Publishing, 2019, pp. 396–411.

MICHELANGELO CECI, Ph.D., is an Associate Professor of Computer Science, University of Bari, Italy. He published more than 150 papers in journals and conferences on Machine Learning and Data Mining. He is the unit coordinator of EU and national projects. He has been in the PC of many conferences (e.g., IEEE ICDM, SDM, IJCAI and AAAI) and in the editorial board of DMKD, MLJ, JIIS. He was PC co-chair of SEBD2007, Discovery Science 2016, ISMIS 2018 and General Chair of ECML-PKDD 2017.

ROBERTO CORIZZO, Ph.D, is an Assistant Professor of Research, Dept. of Computer Science, American University. Prior to that, he was a Research Fellow in the Department of Computer Science at University of Bari, Italy. He co-authored 16 articles, including 7 publications in journals such as IEEE Transactions on Industrial Informatics, Data Mining and Knowledge Discovery, and Information Sciences. He participated in the scientific committee of international conferences and served as a reviewer for several international journals.

NATHALIE JAPKOWICZ, Ph.D., is a Professor of Computer Science at American University. Prior to that she directed the Laboratory for Research on Machine Learning applied for Defense and Security at the University of Ottawa. She trained over 30 graduate students and collaborated with Canadian and US governmental agencies, as well as private industry. She published a co-authored book, one edited book and over 100 book chapters, journal and conference papers. She received five best paper awards, including the prestigious ECML2014 Test of Time award.

PAOLO MIGNONE , Ph.D, is a Research Fellow at the Dept. of Computer Science, University of Bari (Italy). He has published papers in top-journals including papers in Bioinformatics, IEEE Transactions on Human-Machine Systems, and conference proceedings. He participated in the scientific committee of international conferences and served as a reviewer in a wide range of international conferences and journals. His research interests are mainly focused on Data Mining and Knowledge Discovery, Bioinformatics, Transfer Learning, Link Prediction, Big Data Analytics, Cybersecurity, and Social Network Analysis.

GIANVITO PIO, Ph.D, is an Assistant Professor at the Dept. of Computer Science, University of Bari, Italy. He published 30 papers, including 13 papers in journals, such as Machine Learning, Bioinformatics, BMC Bioinformatics, PloS ONE, Information Sciences and IEEE Transactions on Knowledge and Data Engineering. He participated in the scientific committee of international conferences and served as a reviewer for several international journals. His interests include big data analytics, bioinformatics and methods to analyze heterogeneous networks.