

Received August 7, 2020, accepted August 20, 2020, date of publication August 24, 2020, date of current version September 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019095

ECHAD: Embedding-Based Change Detection From Multivariate Time Series in Smart Grids

MICHELANGELO CECI^{1,2,3}, ROBERTO CORIZZO⁴, (Member, IEEE), NATHALIE JAPKOWICZ⁴,
PAOLO MIGNONE^{1,3}, AND GIANVITO PIO^{1,3}

¹Department of Computer Science, University of Bari Aldo Moro, 70125 Bari, Italy

²Department of Knowledge Technologies, Jožef Stefan Institute, 1000 Ljubljana, Slovenia

³Big Data Laboratory, National Interuniversity Consortium for Informatics (CINI), 00185 Rome, Italy

⁴Department of Computer Science, American University, Washington, DC 20016, USA

Corresponding author: Michelangelo Ceci (michelangelo.ceci@uniba.it)

This work was supported by the Ministry of University and Research (MUR) through the Project Community Energy Storage (ComESto): Gestione Aggregata di Sistemi d'Accumulo dell'Energia in Power Cloud (Community Energy Storage: Aggregate Management of Energy Storage Systems in Power Cloud) under Grant ARS01_01259. The work of Roberto Corizzo was supported by the U.S. Defense Advanced Research Projects Agency (DARPA) through the Project Lifelong Streaming Anomaly Detection under Grant A19-0131-003. The work of Gianvito Pio was supported by the Ministry of University and Research (MUR) through the Project Big Data Analytics under Grant AIM 1852414, activity 1, line 1.

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 article, 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

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

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.

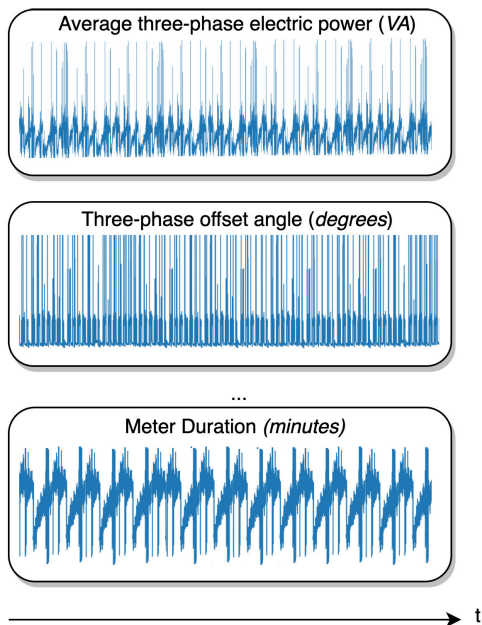


FIGURE 1. A graphical representation of multiple time series corresponding to real data possibly observed in a power grid.

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;
- 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 article 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 article, 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 geodistributed 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 article, 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.

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.

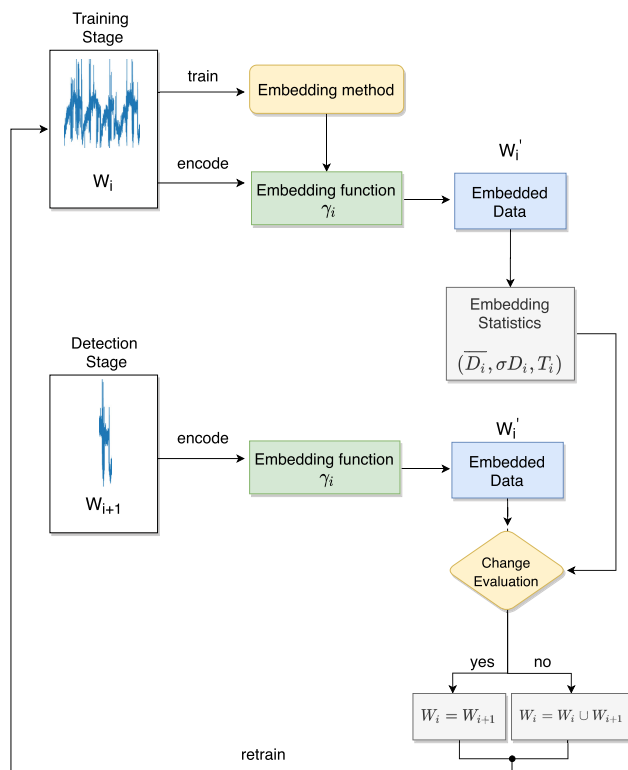


FIGURE 2. A graphical representation of the proposed method implemented in ECHAD.

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 $\gamma_i: \mathbb{R}^M \rightarrow \mathbb{R}^K$, 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 article, 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:

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

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:

$$\gamma_i(w) = e_i(w) \quad (3)$$

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: $x \in \mathbb{R}^K$ be a training instance belonging to N_i and $Neigh(x)$ be the set of the p nearest neighbors of x . Then:

$$\begin{aligned} \overline{D}_i &= \frac{1}{p} \sum_{q \in Neigh(x)} eucl_dist(x, q) \\ \sigma D_i &= \sqrt{\frac{1}{p} \sum_{q \in Neigh(x)} (eucl_dist(x, q) - \overline{D}_i)^2} \end{aligned} \quad (5)$$

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):

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

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 ($O(N_i \cdot \log N_i)$, assuming to use a tree-based structure), and the complexity of computing \overline{D}_i , σD_i and T_i ($O(p \cdot K \cdot N_i)$, according to Equation 5). Therefore, the overall time complexity of ECHAD is $O(N_i \cdot M^2 + M^3 + N_i \cdot \log N_i + p \cdot K \cdot N_i)$ for the variant based on PCA and $O(N_i \cdot M^2 + N_i \cdot \log N_i + p \cdot K \cdot N_i)$ for the variant based on autoencoders. If we assume that $M^2 > \log N_i$ 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

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

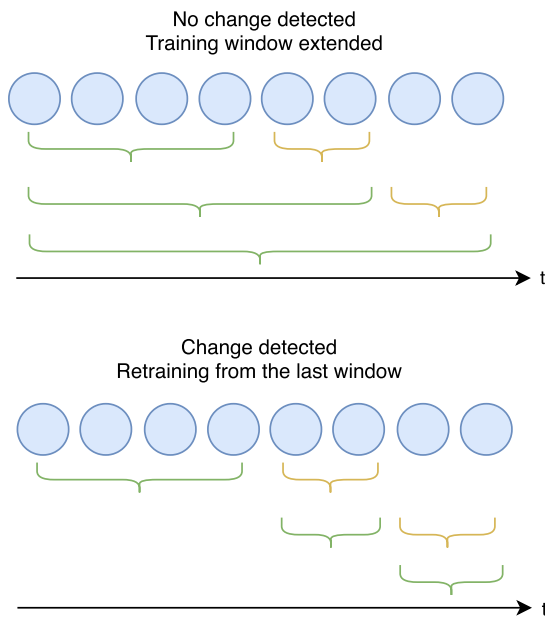


FIGURE 3. A graphical representation of the adopted mixed windows model. The schema shows the evolution of the training (green) and testing (yellow) windows over time. When no changes are detected, the training window is extended over time (see the top part of the figure). When a change is detected,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

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.

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.*

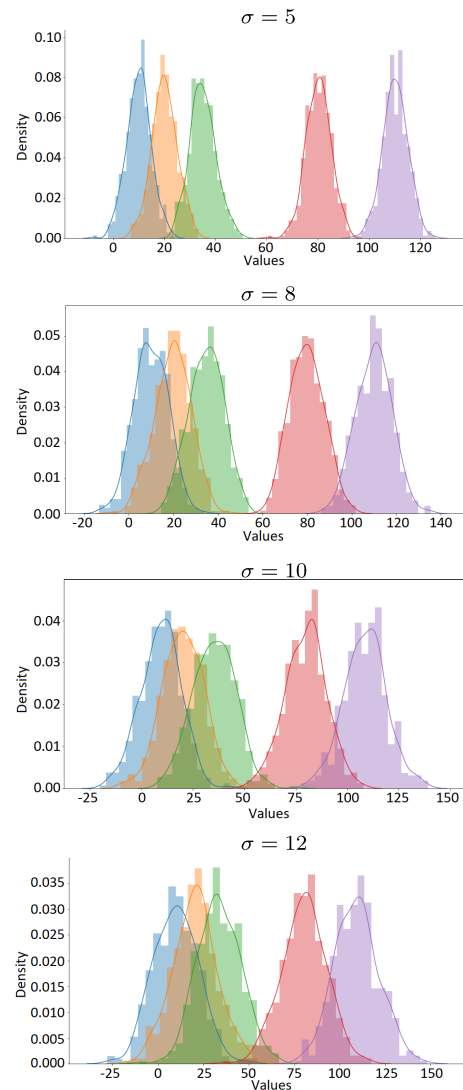


FIGURE 4. A graphical representation of the synthetic datasets, with different values of the standard deviation σ .

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

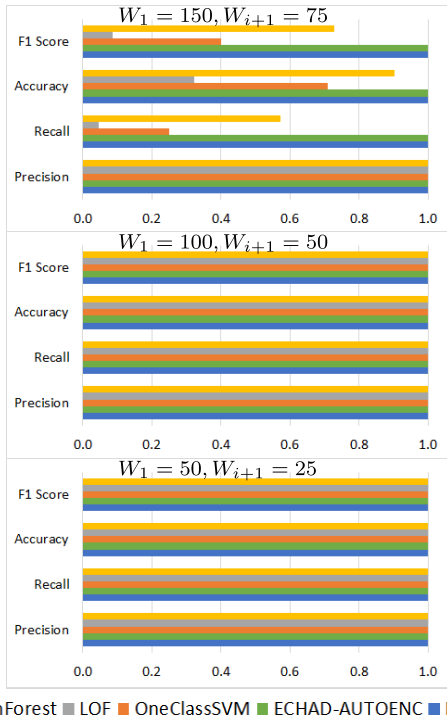


FIGURE 5. F1 Score, Accuracy, Precision, and Recall measured on the first synthetic datasets ($\sigma = 5$).

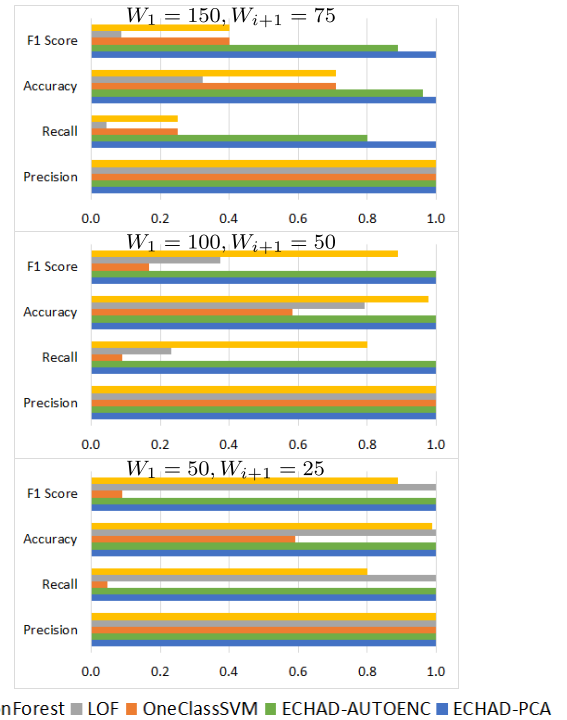


FIGURE 7. F1 Score, Accuracy, Precision, and Recall measured on the third synthetic datasets ($\sigma = 10$).

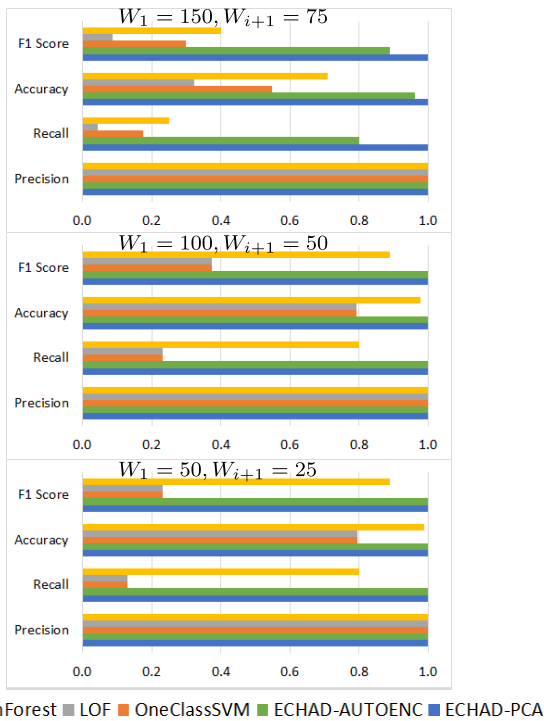


FIGURE 6. F1 Score, Accuracy, Precision, and Recall measured on the second synthetic datasets ($\sigma = 8$).

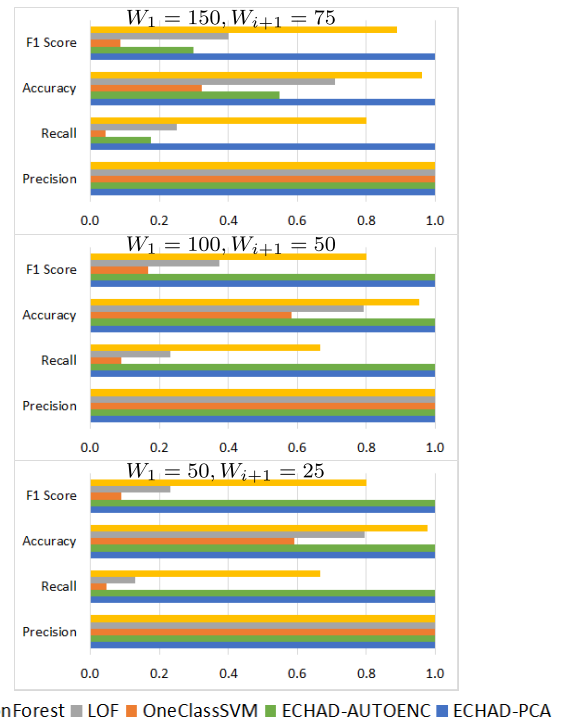


FIGURE 8. F1 Score, Accuracy, Precision, and Recall measured on the fourth synthetic datasets ($\sigma = 12$).

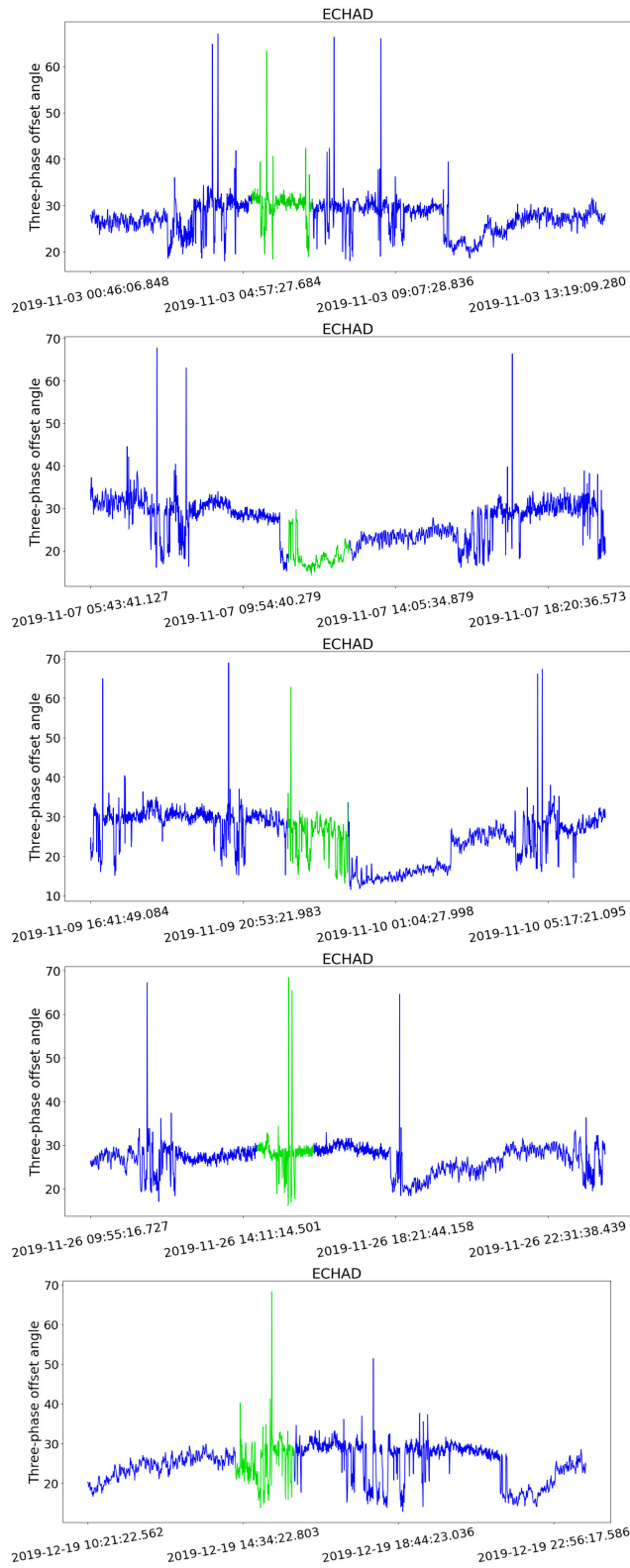


FIGURE 9. Examples of time windows representing a change (in green), identified by ECHAD but not by Isolation Forest and LOF for *three-phase offset angle*.

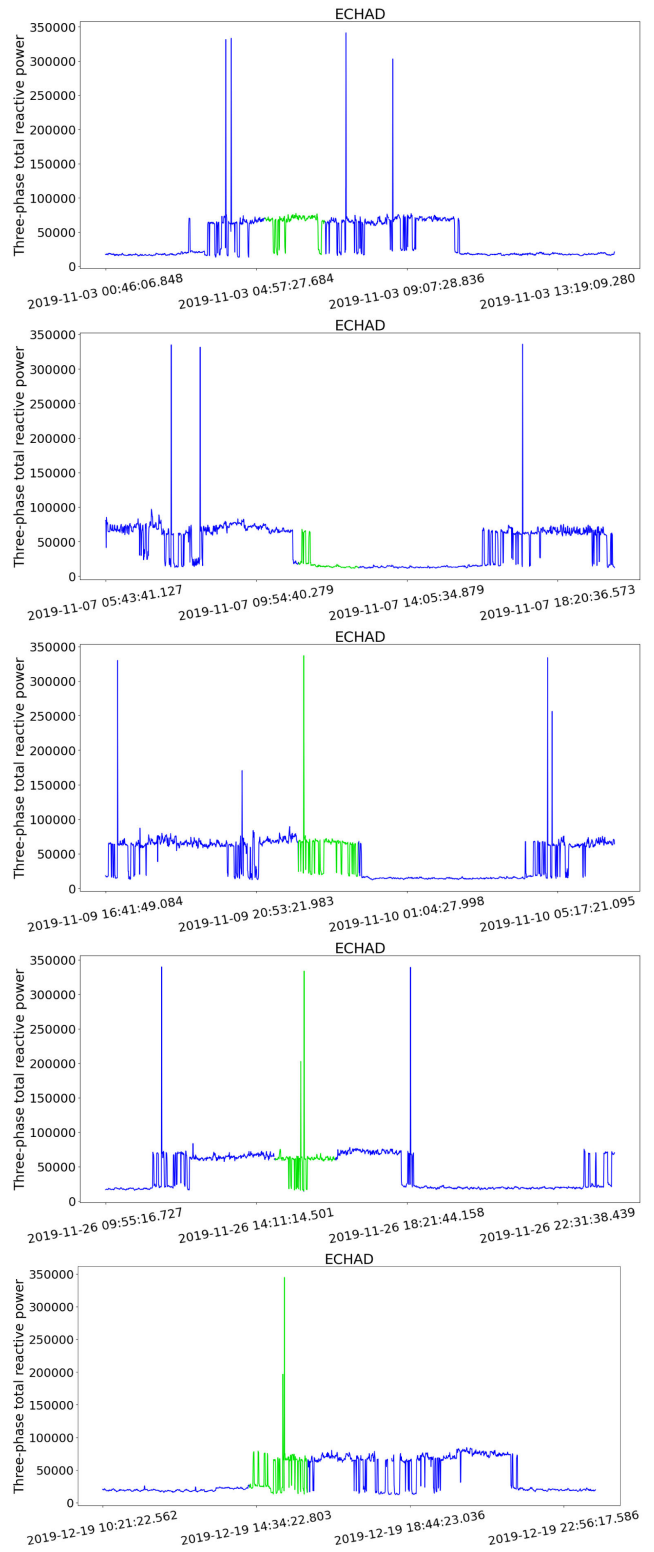


FIGURE 10. Examples of time windows representing a change (in green), identified by ECHAD but not by Isolation Forest and LOF for *three-phase total reactive power*.

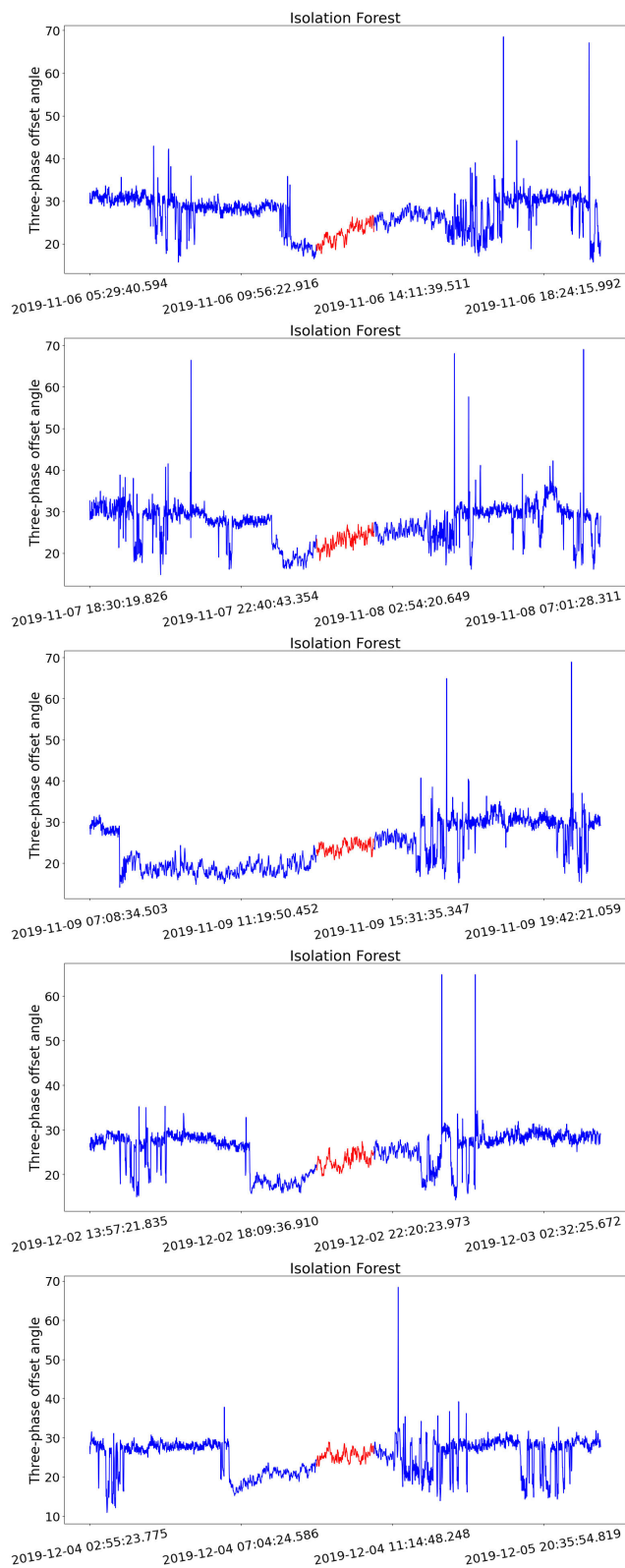


FIGURE 11. Examples of false positive detections (in red) returned by Isolation Forest, that were not returned by ECHAD, for *three-phase offset angle*.

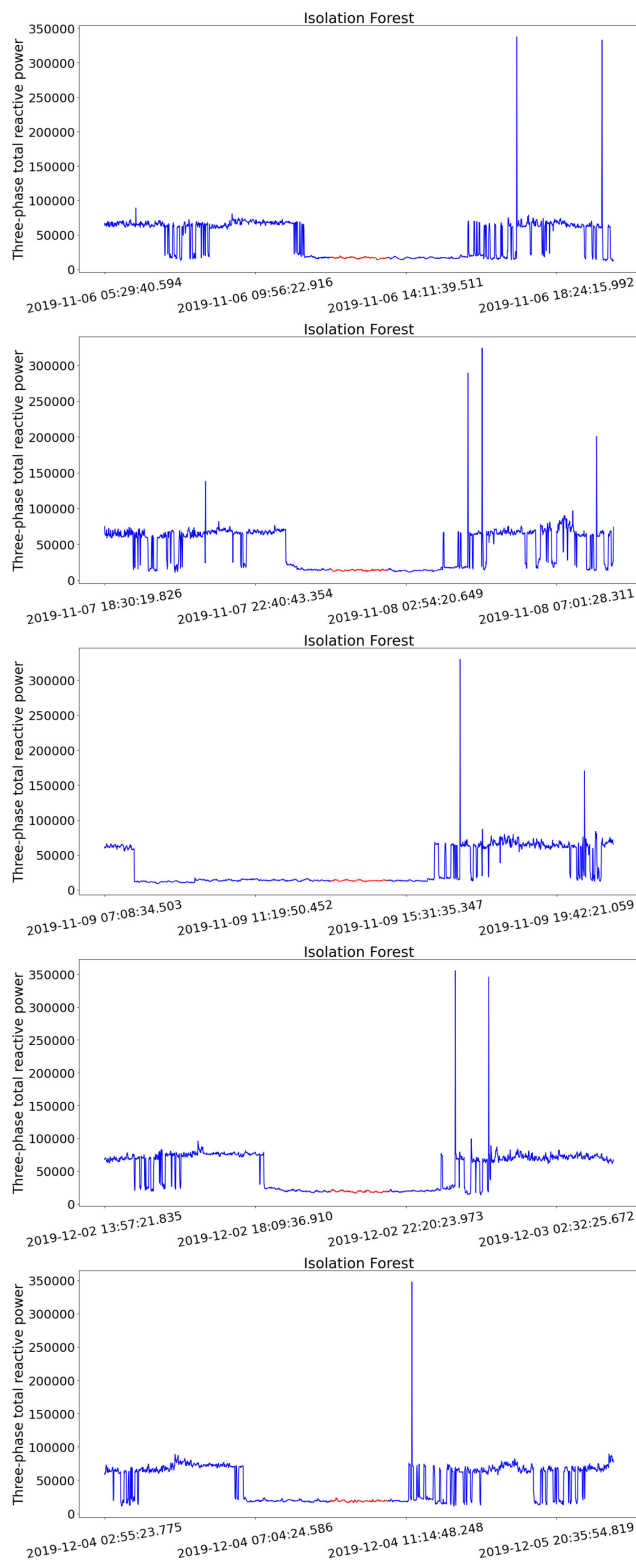


FIGURE 12. Examples of false positive detections (in red) returned by Isolation Forest, that were not returned by ECHAD, for *three-phase total reactive power*.

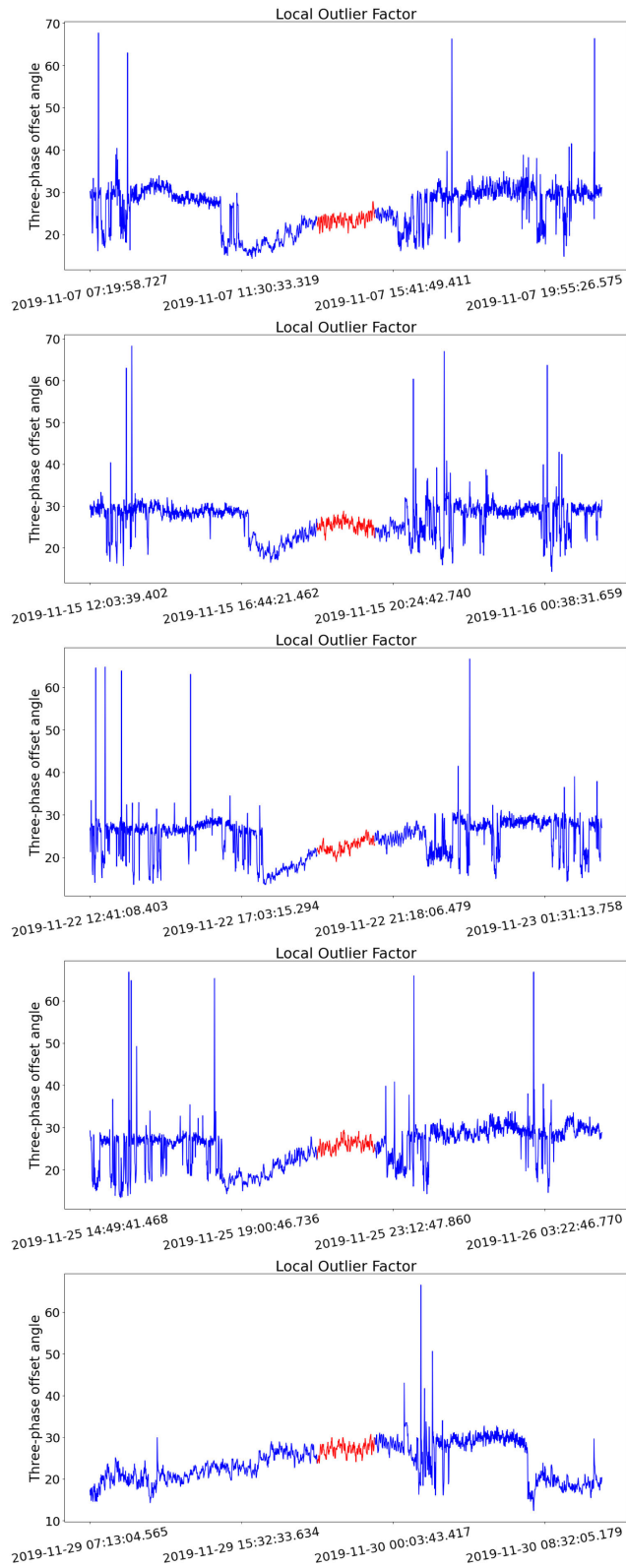


FIGURE 13. Examples of false positive detections (in red) returned by LOF, that were not returned by ECHAD, for *three-phase offset angle*.

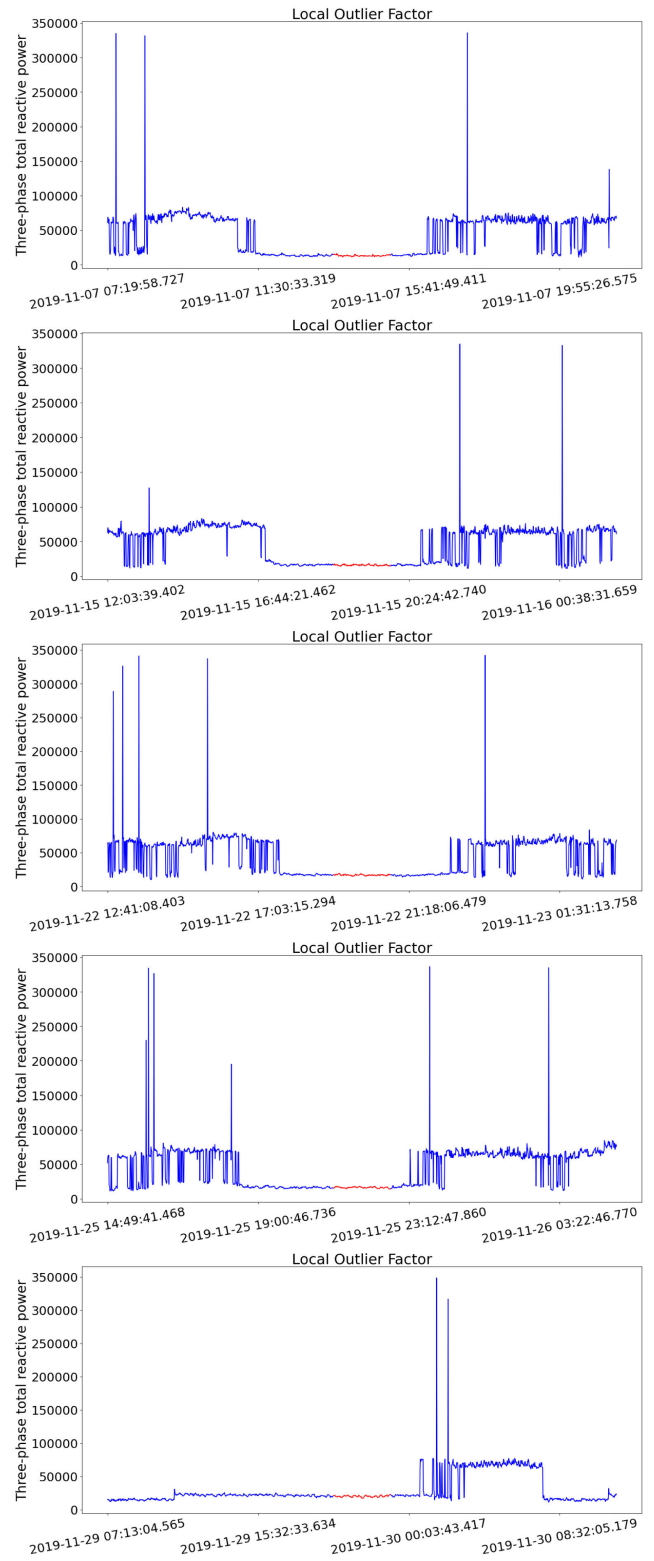


FIGURE 14. Examples of false positive detections (in red) returned by LOF, that were not returned by ECHAD, for *three-phase total reactive power*.

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.

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 article, 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

The authors acknowledge the support of the Ministry of University and Research (MUR) through the project Community Energy Storage (ComESto): Gestione Aggregata di Sistemi d'Accumulo dell'Energia in Power Cloud (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. The authors thank Claudio Alberti and Francesco Dura from e-distribuzione S.p.A. for fruitful discussion and for providing real-world datasets. 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).

REFERENCES

- [1] N. Draper and H. Smith, *Applied regression analysis* (Wiley series in probability and statistics: Texts and references section), vol. 1. Hoboken, NJ, USA: Wiley, 1998.
- [2] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 582–588.
- [3] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.
- [4] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.

- [5] N. Japkowicz, C. Myers, and M. A. Gluck, "A novelty detection approach to classification," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, vol. 1, Aug. 1995, pp. 518–523.
- [6] R. Corizzo, M. Ceci, E. Zdravetski, and N. Japkowicz, "Scalable autoencoders for gravitational waves detection from time series data," *Expert Syst. Appl.*, vol. 151, Aug. 2020, Art. no. 113378.
- [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. 36540–36548, 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, Apr. 2017.
- [9] R. Corizzo, G. Pio, M. Ceci, and D. Malerba, "DENCAST: Distributed density-based clustering for multi-target regression," *J. Big Data*, vol. 6, no. 1, p. 43, Dec. 2019.
- [10] M. Ceci, R. Corizzo, D. Malerba, and A. Rashkovska, "Spatial autocorrelation and entropy for renewable energy forecasting," *Data Mining Knowl. Discovery*, vol. 33, pp. 698–729, Jan. 2019.
- [11] N. Sharma, P. Sharma, D. Irwin, and P. Shenoy, "Predicting solar generation from weather forecasts using machine learning," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Oct. 2011, pp. 528–533.
- [12] G. N. Bathurst, J. Weatherill, and G. Strbac, "Trading wind generation in short term energy markets," *IEEE Trans. Power Syst.*, vol. 17, no. 3, pp. 782–789, Aug. 2002.
- [13] C. Changsong, D. Shanxu, C. Tao, L. Bangyin, and Y. Jinjun, "Energy trading model for optimal microgrid scheduling based on genetic algorithm," in *Proc. IEEE 6th Int. Power Electron. Motion Control Conf.*, May 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 Trans. Ind. Informat.*, vol. 13, no. 3, pp. 956–966, Jun. 2017.
- [15] R. J. Bessa, A. Trindade, and V. Miranda, "Spatial-temporal solar power forecasting for smart grids," *IEEE Trans. Ind. Informat.*, vol. 11, no. 1, pp. 232–241, Feb. 2015.
- [16] R. Corizzo, M. Ceci, and N. Japkowicz, "Anomaly detection and repair for accurate predictions in geo-distributed big data," *Big Data Res.*, vol. 16, pp. 18–35, Jul. 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," *J. Amer. Stat. Assoc.*, vol. 101, no. 475, pp. 968–979, 2006.
- [18] R. Corizzo, M. Ceci, H. Fanaee-T, and J. Gama, "Multi-aspect renewable energy forecasting," *Inf. Sci.*, p. 15718, 2020.
- [19] R. J. Bessa, V. Miranda, and J. Gama, "Entropy and correntropy against minimum square error in offline and online three-day ahead wind power forecasting," *IEEE Trans. Power Syst.*, vol. 24, no. 4, pp. 1657–1666, Nov. 2009.
- [20] J. Gama and M. M. Gaber, Eds., *Learning From Data Streams*. Berlin, Germany: Springer-Verlag, 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. 24148–24156, 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. 80778–80788, 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. 26022–26033, 2017.
- [25] M. M. Moya, M. W. Koch, and L. D. Hostetler, "One-class classifier networks for target recognition applications," NASA, Washington, DC, USA, STI/Recon Tech. Rep. SAND-93-0084C ON: DE93006288, 1993, vol. 93.
- [26] N. Japkowicz, "Concept-learning absence counter-examples: Autoassociation-based approach to classification," Rutgers Univ. Libraries, New Brunswick, NJ, USA, Tech. Rep. DCS-TR-390, 1999. [Online]. Available: <https://rucore.libraries.rutgers.edu/rutgers-lib/59019/>
- [27] D. Tax, "One-class classification; Concept-learning in the absence of counterexamples," Ph.D. dissertation, Delft Univ. Technol., Delft, The Netherlands, 2001, vol. 146.
- [28] G. Ritter and M. T. Gallegos, "Outliers in statistical pattern recognition and an application to automatic chromosome classification," *Pattern Recognit. Lett.*, vol. 18, no. 6, pp. 525–539, Jun. 1997.
- [29] C. M. Bishop, "Novelty detection and neural network validation," *IEEE Proc.-Vis., Image Signal Process.*, vol. 141, no. 4, pp. 217–222, Aug. 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*, pp. 1553–1561, Oct. 2019.
- [31] C. Bellinger, S. Sharma, and N. Japkowicz, "One-class versus binary classification: Which and when?" in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, Dec. 2012, pp. 102–106.
- [32] T. Kefi-Fatfeh, R. Ksantini, M.-B. Kaàniche, and A. Bouhoula, "A novel incremental one-class support vector machine based on low variance direction," *Pattern Recognit.*, vol. 91, pp. 308–321, Jul. 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," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, Dec. 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 Commun. Lett.*, vol. 9, no. 2, pp. 139–142, Feb. 2020.
- [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," *Sol. Energy*, vol. 179, pp. 48–58, Feb. 2019.
- [36] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation-based anomaly detection," *ACM Trans. Knowl. Discovery from Data*, vol. 6, no. 1, pp. 1–39, Mar. 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 Trans. Geosci. Remote Sens.*, early access, Mar. 18, 2020, doi: [10.1109/TGRS.2020.2978491](https://doi.org/10.1109/TGRS.2020.2978491).
- [38] S. Yu, X. Li, L. Zhao, and J. Wang, "Hyperspectral anomaly detection based on low-rank representation using local outlier factor," *IEEE Geosci. Remote Sens. Lett.*, early access, Jun. 1, 2020, doi: [10.1109/LGRS.2020.2994745](https://doi.org/10.1109/LGRS.2020.2994745).
- [39] C. Lee, S. Jo, D. Kwon, and M. Pecht, "Capacity-fading behavior analysis for early detection of unhealthy li-ion batteries," *IEEE Trans. Ind. Electron.*, early access, Feb. 13, 2020, doi: [10.1109/TIE.2020.2972468](https://doi.org/10.1109/TIE.2020.2972468).
- [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 Recognit.*, 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 *Proc. Irish Conf. Artif. Intell. Cogn. Sci.*, 2009, pp. 188–197.
- [42] S. S. Khan and M. G. Madden, "One-class classification: Taxonomy of study and review of techniques," *Knowl. Eng. Rev.*, vol. 29, no. 3, pp. 345–374, Jun. 2014.
- [43] M. A. F. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Process.*, vol. 99, pp. 215–249, Jun. 2014.
- [44] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, p. 15, 2009.
- [45] C. C. Aggarwal, "Outlier analysis," in *Data Mining*. Springer, 2015, pp. 237–263.
- [46] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Phil. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, Nov. 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," *J. Mach. Learn. Res.*, vol. 11, no. 12, 2010.
- [48] Y. Bengio, *Learning Deep Architectures for AI*. New York, NY, USA: Now Publishers Inc, 2009.
- [49] P. K. 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*, 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," *Comput. Chem. Eng.*, vol. 26, no. 9, pp. 1281–1293, Sep. 2002.
- [51] R. Noori, A. R. Karbassi, A. Moghaddammia, D. Han, M. H. Zokaie-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," *J. Hydrol.*, vol. 401, nos. 3–4, pp. 177–189, May 2011.
- [52] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 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. K. Novak, T. Šmuc, and S. Džeroski, Eds. Cham, Switzerland: Springer, 2019, pp. 396–411.



MICHELANGELO CECI received the Ph.D. degree. He is currently an Associate Professor of computer science with the University of Bari, Italy. He has published more than 150 papers in journals and conferences on machine learning and data mining. He is also 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). He was the PC Co-Chair of SEBD2007, Discovery Science 2016, and ISMIS 2018; and the General Chair of ECML-PKDD 2017. He has been on the Editorial Board of DMKD, MLJ, and JIIS.



ROBERTO CORIZZO (Member, IEEE) received the Ph.D. degree. He is currently an Assistant Professor of research with the Department of Computer Science, American University. Prior to that, he was a Research Fellow with the Department of Computer Science, University of Bari, Italy. He has coauthored 16 articles, including seven publications in journals, such as the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, *Data Mining and Knowledge Discovery*, and *Information Sciences*. He has participated in the Scientific Committee of international conferences and served as a Reviewer for several international journals.



NATHALIE JAPKOWICZ received the Ph.D. degree. She is currently a Professor of computer science with American University. Prior to that, she directed the Laboratory for Research on Machine Learning applied for Defense and Security, University of Ottawa. She trained over 30 graduate students and collaborated with Canadian and U.S. governmental agencies, as well as private industry. She has published a coauthored book; one edited book; and over 100 book chapters, journals, and conference papers. She received five best paper awards, including the prestigious ECML2014 Test of Time Award.



PAOLO MIGNONE received the Ph.D. degree. He is currently a Research Fellow with the Department of Computer Science, University of Bari, Italy. He has published articles in top-journals, including articles in *Bioinformatics* and the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, and has published papers in conference proceedings. He has 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 include data mining and knowledge discovery, bioinformatics, transfer learning, link prediction, big data analytics, cybersecurity, and social network analysis.



GIANVITO PIO received the Ph.D. degree. He is currently an Assistant Professor with the Department of Computer Science, University of Bari, Italy. He has published 30 articles, including 13 articles in journals, such as *Machine Learning*, *Bioinformatics*, *BMC Bioinformatics*, *PloS ONE*, *Information Sciences*, and the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING. He has participated in the Scientific Committee of international conferences and served as a Reviewer for several international journals. His research interests include big data analytics, bioinformatics, and methods to analyze heterogeneous networks.

...