



HAL
open science

Do deep neural networks contribute to multivariate time series anomaly detection?

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, Maria A. Zuluaga

► To cite this version:

Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, Maria A. Zuluaga. Do deep neural networks contribute to multivariate time series anomaly detection?. Pattern Recognition, 2022, 132, pp.108945. 10.1016/j.patcog.2022.108945 . hal-03831581

HAL Id: hal-03831581

<https://hal.science/hal-03831581>

Submitted on 27 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Do Deep Neural Networks Contribute to Multivariate Time Series Anomaly Detection?

Julien Audibert^{a,b,*}, Pietro Michiardi^b, Frédéric Guyard^c, Sébastien Marti^a,
Maria A. Zuluaga^{b,*}

^aOrange, Sophia Antipolis, France

^bEURECOM, Sophia Antipolis, France

^cOrange Labs, Sophia Antipolis, France

Abstract

Anomaly detection in time series is a complex task that has been widely studied. In recent years, the ability of unsupervised anomaly detection algorithms has received much attention. This trend has led researchers to compare only learning-based methods in their articles, abandoning some more conventional approaches. As a result, the community in this field has been encouraged to propose increasingly complex learning-based models mainly based on deep neural networks. To our knowledge, there are no comparative studies between conventional, machine learning-based and, deep neural network methods for the detection of anomalies in multivariate time series. In this work, we study the anomaly detection performance of sixteen conventional, machine learning-based and, deep neural network approaches on five real-world open datasets. By analyzing and comparing the performance of each of the sixteen methods, we show that no family of methods outperforms the others. Therefore, we encourage the community to reincorporate the three categories of methods in the anomaly detection in multivariate time series benchmarks.

Keywords: Anomaly detection, Multivariate time series, Neural Networks

1. Introduction

A multivariate time series is a set of measurements correlated with each other over time, which models and represents the behavior of a multivariate process in time. Multivariate time series are used in a large number of fields such as industrial control systems [1], finance [2], and healthcare [3]. Detecting unexpected behavior or patterns that do not conform to the expected behavior

*Corresponding author

Email addresses: julien.audibert@eurecom.fr (Julien Audibert),
pietro.michiardi@eurecom.fr (Pietro Michiardi), frederic.guyard@orange.com (Frédéric Guyard), sebastien.marti@orange.com (Sébastien Marti), zuluaga@eurecom.fr (Maria A. Zuluaga)

of previously seen data is an active research discipline called anomaly detection in multivariate time series [4, 5]. Anomalies may indicate a significant problem in several applications. For example, an anomaly in industrial control systems may indicate a malfunction, financial anomalies may be the result of a fraud, or they may indicate diseases in healthcare. Being a critical task, there is a wide range of methods that have been developed to address it [5].

Over the last decade, there has been an increased enthusiasm around deep neural networks (DNNs) [6] thanks to a reported number of successes in multiple tasks [2, 3, 7], and their demonstrated ability to infer high-order correlations in complex data with potentially large volume and dimensionality. Multivariate time series anomaly detection has been no exception to the trend, leading to an explosion of DNN-based methods suggesting methodological advances and improved performance, as presented in [8]. DNN-based methods aim to learn deep latent representations of the multivariate time series to infer a model of variability, that is then used for anomaly grading in unseen data. The rationale behind the increased use of DNN architectures lies in the need of learning potentially complex data patterns underlying the temporal evolution of multivariate data.

Under the above-mentioned argument and motivated by the good performance of DNNs in other fields, researchers have moved away from comparisons with more traditional methods, i.e. machine learning and conventional/statistical methods (e.g. [1, 9, 7]). This trend has encouraged the community to develop even more complex models to improve the performance of DNN-based methods, without any theoretical or empirical evidence that these are superior to the more established body of methods in the literature.

DNN-based models are complex to train, involving the estimation of a large amount of parameters and requiring large training sample sizes and computational resources. Moreover, their complexity continues to grow as larger models continue to be developed. Instead, conventional models are simpler, lighter, easier to interpret, and often better adapted to the constraints of real-world applications. It is therefore crucial to determine if the complexity brought in by DNN-based methods is a necessary price to pay for a gain in performance or if the progress reported in recent years is illusory [10] and the use of conventional methods should be preferred. The lack of a general comparison covering all families of methods does not allow to answer this question and hinders the translation and use of DNN-based methods in real-world applications.

In this paper, we aim to close this gap by establishing a thorough comparison between conventional methods, machine learning-based and more recent DNN-based approaches. Our work is motivated by different recent works, which have reported on the limitations and drawbacks of DNN-based methods in different application fields [11, 12, 13, 14]. While some of this works have focused on pointing out to the weaknesses of DNN-based methods [11, 12], other works have been able to demonstrate the superiority of more conventional approaches [13, 14].

The main contributions of this paper are the following:

- We describe sixteen of the most commonly used methods for anomaly detection in multivariate time series grouped into three categories: Conventional, machine learning-based and DNN-based.
- We study and analyze the performance of these techniques over five open real-world data sets.
- Finally, we discuss the need for DNN-based approaches and the importance of conventional methods in future benchmarks for multivariate time series anomaly detection.

The rest of this document is organized as follows. Section 2 briefly reviews other works comparing modern DNN-based methods to previous non-DNN-based works. Section 3 formalizes the problem of anomaly detection in multivariate time series. Section 4 presents the methods compared in this study. Sections 5 describe the experiments and analyze the performance on the data sets.

2. Related Works

Different studies have raised the question about the real gain of DNN-based methods in several application fields.

A first study by Jiao *et al.* [13], showed how conventional linear regression methods outperform DNN-based techniques in two showcased optical imaging problems, i.e. an optical cryptosystem attack and blind reconstruction in single pixel imaging. Autun *et al.* [11] proposed a stability test to demonstrate how DNN-based methods for image reconstruction are very sensitive to tiny perturbations in the input images during training, which leads to unstable results. Furthermore, Heaven [12] showed small changes in a DNN’s input, usually imperceptible to humans, can destabilize the best neural networks, thus pointing to the lack of robustness of DNN-based methods and their dependence on large amounts of data. Most recently, in the context of medical image segmentation, Fu *et al.* [15] showed that simpler DNN configurations have better generalization properties than complex state-of-the-art DNN models, thus challenging the current trend towards continuously increasing the model complexity.

In the specific context of time series analysis, Garg *et al.* [16] raise the fact that there is no benchmark in the literature between multivariate methods for anomaly detection in time series. They propose a benchmark of 15 methods out of which 12 are DNNs methods.

Finally, the results of the M3 challenge on time series forecasting [14] showed that the accuracy of machine learning and DNN models, in general, was lower than that of conventional approaches, while their computational requirements were considerably greater than those of conventional statistical methods. Similarly, one of the main outcomes of the follow-up M4 competition [17] was that none of the pure ML methods participating was able to outperform the combination of DNN models and statistical (i.e. conventional) methods. For instance, only one DNN approach was more accurate than a naive random walk

model that assumed future values will be the same as those of the last known observation [14].

3. Definitions

This section is concerned with providing a clear definition of multivariate time series (Section 3.1), of the types of anomalies that can be found in a multivariate time series (Section 3.2), and of formalizing the problem of anomaly detection in multivariate time series (Section 3.3)

3.1. Multivariate Time Series

A multivariate time series is a sequence of data points $\mathcal{T} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$, $\mathbf{x}_t \in \mathbb{R}^m \forall t$, each being an observation measured at a specific time t in a process, with m the number of time-series. A univariate time series is a special case of multivariate time series with $m = 1$. In this paper we focus on the general context, multivariate time series.

3.2. Types of Anomalies

Anomalies in time series, also called outliers, are points or sequences of points that do not correspond to normal behavior [4]. The concept of normal behaviour is difficult to formalize. Therefore, another possible definition for anomalies could be a pattern in data that is not expected in comparison to what has been seen before [4]. In fact, an implicit assumption is that anomalies are rare events.

Chandola et al. [4] propose to classify time series anomalies into three types, point, contextual and collective anomalies.

- **Point anomalies.** This is the simplest type of anomaly. It corresponds to a point that differs from the rest of the data (Figure 1).
- **Contextual anomalies.** A contextual anomaly can be defined as follows : A data point is anomalous in a specific context, but is considered normal in another context, meaning that observing the same point through different contexts does not always give a sign of abnormal behavior. For example, a temperature of 30C during summer is normal, while the same temperature in winter is abnormal. Figure 1 illustrates a contextual anomaly, where the values of the two time-series are not abnormal taken individually, but, seen in context, the values of the bottom time-series should be close to 0.
- **Collective anomalies** A collective anomaly corresponds to a group of anomaly points. Each individual point might not be an anomaly, but their appearance together is an anomaly (Figure 1).

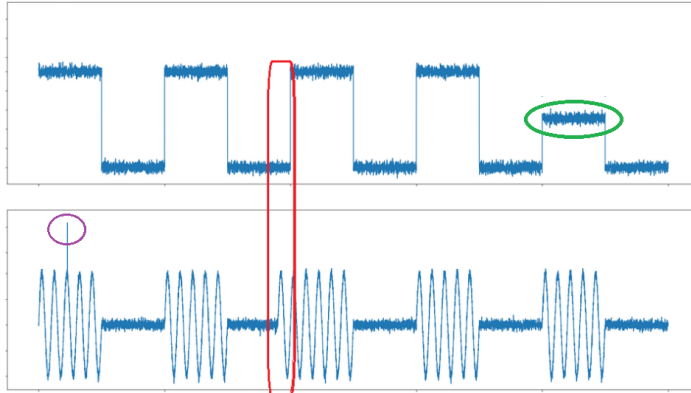


Figure 1: An example of a point anomaly (in purple), a contextual anomaly (in red) and a collective anomaly (in green)

3.3. Problem Formulation

Let us now consider a problem where \mathcal{T} is given, and let us introduce a binary variable $y \in \{0, 1\}$. The goal of anomaly detection methods is to learn a model that assigns to an observation $\hat{\mathbf{x}}_t$, $t > T$, a label y_t to indicate a detected anomaly at time t , *i.e.* $y_t = 1$, or not ($y_t = 0$).

Supervised anomaly detection assumes that, jointly with \mathcal{T} , there are labels that assign to every $\mathbf{x}_t \in \mathcal{T}$ one of two classes, normal or abnormal. During training, the model is thus expected to learn how to classify anomalies by using both \mathcal{T} and the associated labels.

Unsupervised anomaly detection assumes \mathcal{T} contains only normal points. Therefore, the model is trained to learn the distribution of normal data and an anomalous point is expected to be one that differs significantly from \mathcal{T} . The difference between the sample $\hat{\mathbf{x}}_t$ and the normal set \mathcal{T} is measured by an anomaly score, which is then compared to a threshold. If the score is above the threshold the point is considered as an anomaly.

In practice, it is difficult to obtain sets \mathcal{T} with a sufficient number of samples from both classes, a requirement of supervised techniques. As a result, unsupervised methods are the most common approach found in the literature [4]. Therefore, this study focuses on unsupervised techniques.

4. Anomaly detection methods

In this section, we first present the different categories to classify different anomaly detection methods (Section 4.1). Next, we describe the sixteen multivariate time series anomaly detection methods used in our experiments. Each method is presented within one of three categories that we propose: conventional (Section 4.2), machine learning-based (Section 4.3) and DNN-based methods (Section 4.4).

4.1. Taxonomy

We define a taxonomy consisting of three classes of anomaly detection methods for multivariate time series. These are: conventional approaches, machine learning-based and DNN-based methods.

Conventional approaches, rely on the assumption that the observed data is generated by a stochastic model and their aim is to estimate a model's parameters from the data and then use the model for prediction [18]. It is often the case that the model hypothesis is considered linear.

The boundary between conventional and machine learning-based approaches is not fully clear. **Machine learning-based** models produce predictions about the results of complex mechanisms by mining databases of inputs for a given problem, without necessarily having an explicit assumption about a model's hypothesis. In this setup, a method aims to learn a function that operates an input data to predict output responses [18].

Finally, **DNN-based methods** are a subclass of non-linear machine learning-based methods which use neural networks with multiple layers [6].

4.2. Conventional methods

The methods presented in this section offer many different approaches. We have classified them into five categories. Control chart, where the objective is to monitor variations in the statistical characteristics of a process. Prediction methods, where the objective is to predict the next point. Decomposition techniques, based on the search for unusual patterns in time series using decomposition and finally similarity-search model based on the search for similar sub-sequences in the data.

4.2.1. Control Charts methods

Control charts are a statistical process control tool. Their goal is to track the mean and variance of a time series over time and to detect abrupt changes in its statistical behaviour. We consider two methods: Multivariate CUMulated SUMs control chart (MCUSUM) and Multivariate Exponential Weighted Moving Average (MEWMA).

In [19], Woodall and Ncube proposed to monitor the performance of multiple variables in a manufacturing system using a control chart called **Multivariate CUMulated SUMs control chart (MCUSUM)**. The MCUSUM uses the fact that the cumulative sum St of the recursive residuals of a statistic s compared to a normal value (like the mean) : $St = s(\mathbf{x}_{t-k+1}, \dots, \mathbf{x}_t)$ is stable for normal data and increasing after the change. Using this property, the MCUSUM anomaly score g_t is based on the comparison of the increase of S_t with a threshold h : $g_t = \max(0, St - \mu + g_{t-1})$ with $g_0 = 0$ and μ the value of St for normal data. MCUSUM iterates over g_t as long as $g_t < h$. If $g_t \geq h$, an alarm is raised and the iteration over g_t is restarted with $g_0 = 0$ in order to detect another change point. Due to this sequential restarting of the test, MCUSUM detects small but persistent structural changes.

Multivariate Exponential Weighted Moving Average (MEWMA) [20] is based, as its name indicates, on the exponential smoothing of data. For a given constant $\lambda \in [0, 1]$, the successive data are smoothed using: $g_t = \lambda \mathbf{x}_t + (1 - \lambda)g_{t-1}$. Unlike MCUSUM, MEWMA gives more importance to recent history values.

4.2.2. Forecast methods

Control Charts methods are usually based on the hypothesis that the individual data in the time-series are independent and identically distributed. This assumption is rarely satisfied in practice. A generic way to handle this issue is to build a mathematical model, incorporating the known or assumed sequential correlations of the time-series. Using this model, the value \mathbf{x}_t of the data indicator at t is expressed as $\mathbf{x}_t = z_t + e_t$, where z_t is accounting for normal sequential correlations within the data and e_t (the residual) is the noise. Once a mathematical model representing time-series evolution is chosen, a usual approach is to predict at time t the expected value $\hat{\mathbf{x}}_{t+1}$. The anomaly score can then be expressed as the difference between $\hat{\mathbf{x}}_{t+1}$ and the actual value \mathbf{x}_{t+1} .

The most commonly used forecast method in multivariate time series is the **Vector Autoregressive (VAR)** method. VAR is a statistical model that captures the inter-dependencies between several time series. In a VAR model, variables are treated symmetrically so that each variable is explained by its own past values and by the past values of the other variables. For example, in [21], VAR is used to monitor multivariate processes.

4.2.3. Decomposition methods

Methods in this category use basis functions for decomposing the time series. Given \mathbf{x}_t , a point of a multivariate time series, it can be expanded in terms of the eigenfunctions ϕ_j as: $\mathbf{x}_t = \sum_{j=1}^{\infty} \alpha_j \phi_{jt}$, where the coefficients α_j are given by the projection of \mathbf{x}_t on the respective eigenfunctions.

Anomaly detection is then performed by computing the projection of each new point onto the eigenvectors, and a normalized reconstruction error. The normalized error is used as the anomaly score. We discuss three methods under this sub-category: Principal Component Analysis (PCA), Singular Spectrum Analysis (SSA) and Independent Component Analysis (ICA).

Principal Component Analysis (PCA). The idea of PCA[22] is to group data points into clusters defined as an ellipsoid in \mathbb{R}^m containing the normal data. The idea is that the length of the principal axes of the ellipsoid represents the direction of the variability of the data instances.

The (always positive) eigenvalues λ_i with $i = 1, \dots, k$ corresponding to the principal component v_i characterize the variability in the dataset captured by the eigenvector v_i . A subspace $S \subset \mathbb{R}^m$ is selected using the r eigenvectors v_i corresponding the r largest eigenvalues.

In [23], the Q-statistic that characterizes the PCA projection residual statistics is used to define the threshold h_α , such that the anomaly score is: $|\tilde{\mathbf{x}}_t| > h_\alpha$ then \mathbf{x}_t is an anomaly with $1 - \alpha$ confidence where $\alpha \in [0, 1]$ is a manually defined parameter.

Singular Spectrum Analysis (SSA) considers a time series as a projection of the trajectory of a dynamical system in a vector space $V = \mathbb{R}^m$ where \mathbf{x}_t is the position at time t of a state of the system. From dynamical systems theory, it is known that physically observable states of a dynamical system in \mathbb{R}^m are lying on attractors of the dynamics (i.e. subsets of the \mathbb{R}^m capturing all the long term evolution of the system). Future observable states should normally continue to be located on these attractors. As such, $\hat{\mathbf{x}}_t$ are assumed to be located on (or close to) an attractor of the dynamics. So, the distance between $\hat{\mathbf{x}}_t$ and the attractor computed at time t is evaluated and used as anomaly score [24].

Independent Component Analysis (ICA) [25] assumes that the different physical processes that generate multivariate time series are statistically independent of each other.

ICA decomposes a multivariate time series into “independent” components by orthogonal rotation and by maximizing the statistical independence between the components assuming their non-Gaussianity. In [26], the authors use Kurtosis, a classical measure of non-Gaussianity, as an anomaly score. A negative Kurtosis indicates a uniformly distributed factor, indicating a clustered structure and a high positive Kurtosis identifies a multivariate anomaly.

4.2.4. Similarity-search approach

Similarity search methods [27] are designed to identify patterns in a multivariate time series. These methods compute the distance between all the sub-sequences of the time series. The minimum distance between a sub-sequence and all others is used as an anomaly score.

The **Matrix-Profile (MP)** [28] is a data structure for time series analysis. It consists of three elements. The first element is the distance profile, which is a vector of Euclidean distances between a given sub-sequence W and each sub-sequence in the set A of all sub-sequences of the multivariate time series. The distance is measured using the z-normalized Euclidean distance between sub-sequences. The distance profiles are arranged into a data structure denoted the distance matrix, which corresponds to the second element in MP. The distance matrix corresponds to all the distance profiles that have been computed for each reference sub-sequence. At the same time, the matrix profile is the simplification of the distance matrix by considering only the nearest neighbor for each sub-sequence allowing a memory complexity of $O(|A|)$. The vector obtained corresponds to the smallest values of each row in the matrix. It is defined as : $MP(i) = \min(d(W^i, W^j))$ with $W^j \in A \setminus \{W^i\}$.

A low value in the matrix profile indicates that the sub-sequence has at least one relatively similar sub-sequence located somewhere in the original series. In [29], it is shown that a high value indicates that the original series must have an abnormal sub-sequence. Therefore the matrix profile can be used as an anomaly score, with a high value indicating an anomaly.

4.3. Machine learning-based methods

The methods presented in this section fall into three categories : Isolation, Neighbourhood-based and Domain-based methods, which have been proposed in the survey by Domingues *et al.* [5]. Isolation algorithms consider a point as an anomaly when it is easy to isolate from others. Neighbourhood-based models look at the neighbourhood of each data point to identify outliers. Domain-based methods rely on the construction of a boundary separating the nominal data from the rest of the input space.

A common characteristic of machine learning-based techniques is that they typically model the dependency between a current time point and previous ones by transforming the multivariate time series \mathcal{T} into a sequence of windows $\mathcal{W} = \{W_1, \dots, W_T\}$, where W_t is a time window of length K at a given time t : $W_t = \{\mathbf{x}_{t-K+1}, \dots, \mathbf{x}_{t-1}, \mathbf{x}_t\}$. In this setup, learning-based anomaly detection methods assign to a window \widehat{W}_t , $t > T$, a label y_t to indicate a detected anomaly at time t , *i.e.* $y_t = 1$, or not ($y_t = 0$) based on the window’s anomaly score. We will use this notation in the following.

4.3.1. Isolation methods

Isolation methods focus on separating outliers from the rest of the data points. These methods attempt to isolate the anomalies rather than mapping the normal points.

The **Isolation Forest (IF) algorithm** [30] is based on decision trees. IF calculates, for each time window an anomaly score. To calculate this score, the algorithm isolates the sample recursively: it chooses a feature and a “cut-off point” at random, then evaluates whether this isolates the sample; if so, the algorithm stops, otherwise, it chooses another feature and another cut-off point at random, and so on until the data is isolated from the rest. The number of cut-off point defines the anomaly score: a sample with a very short path, *i.e.* sample that is easy to isolate, is also likely to be an anomaly since it is very far from the other samples in the dataset.

4.3.2. Neighbourhood-based methods

Among neighborhood-based methods, which study the neighborhoods of every point to identify anomalies, the **Local Outlier Factor (LOF)** [31] measures the local deviation of a given data point with respect to its neighbours. Based on the K-nearest neighbors, the local density of an observation is evaluated by considering the closest K observations in its neighbourhood. The anomaly score is then calculated by contrasting its local density with those of its k-nearest neighbors. A high score indicates a lower density than its neighbors and therefore potentially an anomaly. It has been applied to multivariate time series [32], demonstrating its ability to detect anomalies in long-term data.

Density-based spatial clustering of applications with noise (DBSCAN) [33] is a clustering method that groups data points in high density areas (many nearby neighbors) and marks points in low-density regions (few

neighbors) as anomalous. To handle multivariate time series, DBSCAN considers each time window as a point with the anomaly score being the distance from the point to the nearest cluster.

4.3.3. Domain-based methods

Domain-based methods aim to construct a boundary between normal samples and the rest of the input space. The distance of a points to this boundary is used as the anomaly score. Among these, the **One-Class Support Vector Machine (OC-SVM)** [34] method learns the smallest hypersphere containing of all the training data points. The learned model classifies points inside the hypersphere as normal and labels those in the rest of the space as anomalous. An anomaly score can also be obtained by taking the signed distance from the hyper-sphere. The signed distance is positive for a normal value and negative for an abnormal. As other machine learning methods One-class SVM has been used for time series anomaly detection by using time windows rather than the raw series. J.Ma and S.Perkins [35] use this approach, while proposing to combine the one-class SVM output for different time windows to produce more robust detection results.

4.4. DNN-based methods

DNN-based methods are a are sub-category of machine learning-based approaches, which rely on deep neural networks. Given the explosion of DNN-based methods over the last years, we present them as a separate category. We selected the five methods that report a good performance in the literature on the five datasets used in this paper. These are: Auto-Encoder (AE), UnSupervised Anomaly Detection (USAD), Long Short-Term Memory Variational Auto-Encoders (LSTM-VAE), Deep Autoencoding Gaussian Mixture Model (DAGMM) and OmniAnomaly (OA).

An **Auto-Encoder (AE)** [36] is a neural network architecture consisting of a combined encoder and decoder. The encoder maps the input windows into a set of latent variables, while the decoder maps the latent variables back into the input space as a reconstruction. The difference between the input window and its reconstruction is the reconstruction error. The AE learns to minimize this error. The anomaly score of a window is the corresponding reconstruction error. A window with a high score is considered abnormal.

UnSupervised Anomaly Detection (USAD) [1] extends the AE concept and constructs two AEs sharing the same encoder. The architecture is driven in two phases. In the first phase, the two AEs learn to reproduce the normal windows. TIn the second phase, an adversarial training teaches the first AE to fool the second one, while the second one learns to recognize the data coming from the input or the reconstructed by the first AE. The anomaly score is the difference between the input data and the data reconstructed by the concatenated AEs.

Long Short-Term Memory Variational Auto-Encoders (LSTM-VAE) [37] uses an LSTM to model temporal dependency, whereas the VAE projects

the input data and its temporal dependencies into a latent space. During decoding, the latent space representation allows to estimate the output distribution. An anomaly is detected when the log-likelihood of the current data is below a threshold. S.Lin et al. [38] have shown LSTM-VAE’s capacity to identify anomalies that span over multiple time scales.

The **Deep Autoencoding Gaussian Mixture Model (DAGMM)** [7] considers a Gaussian Mixture Model (GMM) and a Deep Auto-encoder together to model the distribution of multidimensional data. The purpose of the Deep Auto-encoder is to generate a low-dimensional representation and a reconstruction error for each input data time window. This representation is used as input of a GMM. The parameters of the Deep Auto-encoder and the mixture model are optimized simultaneously from end to end, taking advantage of a separate estimation network to facilitate the learning-based of the parameters of the mixture model. The DAGMM then uses the likelihood to observe the input samples as an anomaly score.

Finally, **OmniAnomaly (OA)** [9] is a recurrent neural network that learns robust representations of time series with a planar normalizing flow and a stochastic variable connection. Then the reconstruction probabilities are used to determine anomalies. OmniAnomaly uses the Gated Recurrent Unit (GRU) to model the time dependencies of multivariate time series. The method also uses a VAE to learn a mapping of the input data W to a representation in a latent space. To model time dependencies in the latent space, OmniAnomaly uses a stochastic variable connection technique. As suggested by [39], the reconstruction can be evaluated by a conditional probability. The anomaly score used is then the probability of reconstruction. A high score means that the input can be well reconstructed. If an observation follows normal time series patterns, it can be reconstructed with high confidence. On the other hand, the lower the score, the less well the observation can be reconstructed and the more likely it is to be anomalous.

5. Experiments and results

We present the experimental setup (Sec. 5.1). Then, we study the performance of the sixteen methods from the different categories (Sec. 4) on the five benchmark datasets. We analyze these results by comparing the anomalies detected by the conventional methods, the machine learning and the DNN-based approaches (Sec. 5.3). We investigate the impact of the size of the training set (Sec. 5.4). Finally, we summarize the results and discuss them (Sec. 5.5).

5.1. Experimental setup

We first describe the benchmark datasets and the performance metrics used in our study.

Table 1: Benchmarked Datasets. (%) is the percentage of anomalous data points in the dataset.

Dataset	Train	Test	Dimensions	Anomalies (%)	Anomaly
SWaT	496800	449919	51	11.98	P, C, Col
WADI	1209601	172801	123	5.99	P, C, Col
SMD	708405	708420	28*38	4.16	P, Col
SMAP	135183	427617	55*25	13.13	P, Col
MSL	58317	73729	27*55	10.72	P, Col

P: Point Anomaly, C: Contextual anomaly, Col: Collective anomaly

Benchmark Datasets. Table 1 summarizes the five datasets used in this study and reports the train and test splits used during the experiments. The **SWaT** dataset¹ is a scaled down version of a real-world industrial water treatment plant. The **WADI** dataset² is an extension of the SWaT testbed. **SMD** is a 5-week-long dataset from an Internet company made publicly available³. **SMAP** and **MSL** are real-world public datasets from NASA⁴. We highlight the fact that, while these datasets are among the most common benchmarks for multi-variate time-series anomaly detection validation, **SMD**, **SMAP** and **MSL** datasets may contain flaws, as pointed out in [10].

Evaluation Metrics. We use the F1 score (F1) and the average precision (AP) from anomaly scores to evaluate anomaly detection performance. The F1 is estimated as $F1 = 2 \cdot P \cdot R / (P + R)$, with $P = (TP) / (TP + FP)$, the precision, $R = TP / (TP + FN)$, the recall, TP the True Positives window, FP the False Positives, and FN the False negatives. In our experiments, we consider a window is labeled as an anomaly as soon as one of the points it contains is detected as anomalous. For methods that do not use a time window, an anomaly is considered detected if the detection occurs within the length of sliding window time points on either side of the anomaly.

The AP summarizes the P-R curve as the weighted mean of precision achieved at every given anomaly score threshold, with the increase in recall from the previous threshold used as the weight. As such, it allows to obtain a score that takes into account the ability of the methods to rank the anomalies without having to define a threshold, thus providing a complementary information w.r.t. the F1 score. It is estimated as: $AP = \sum_n (R_n - R_{n-1}) P_n$ where P_n and R_n are the precision and recall at the n-th threshold. The AP is particularly sensitive to the positive class, i.e. the anomalies, so it is well-suited to highly unbalanced data, as it is the case in anomaly detection.

¹https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#swat

²https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/#wadi

³<https://github.com/smallcowbaby/OmniAnomaly>

⁴<https://s3-us-west-2.amazonaws.com/telemanom/data.zip>

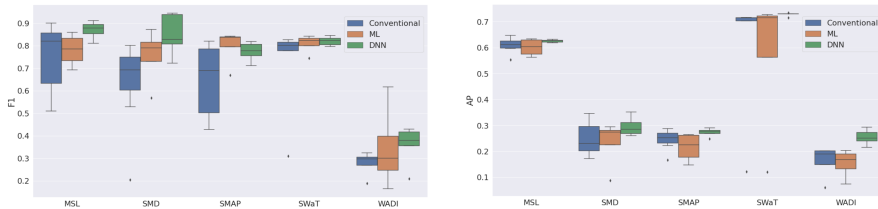


Figure 2: Boxplots of F1 (left) and AP (right) for each category of methods on the five benchmark dataset.

5.2. Benchmark Performance

We assessed the performance of the sixteen anomaly detection methods on the five benchmark datasets in terms of F1 score and AP. As not all of the anomaly detection approaches provide a mechanism for selecting anomaly thresholds, which are required for F1 estimation, we tested a thousand possible anomaly thresholds for each model. In particular, we normalized the anomaly score between 0 and 1 and then tested one thousand thresholds in steps of 0.001. We report the results associated to the highest achieved F1 score among the thresholds. Figure 2 presents boxplots summarizing the performance of each family of methods. Table 2 lists the best and worst performing method for each family of methods.

In terms of F1 score (Fig. 2 left), DNN methods perform best on average four out of five datasets. In the remaining dataset, SMAP, ML methods obtain the best average performance. A detailed comparison of the performance of the different methods in terms of AP (Fig. 2 right), indicated that in terms of mean values, conventional methods outperform all families of methods in the MSL dataset and they are the second best ranked after DNN methods for the remaining datasets. Most importantly, for both F1 and AP the boxplots suggest that the differences among families of methods are relatively subtle.

To confirm this hypothesis, we performed a KruskalWallis test on both measurements for each group of methods to determine if the medians of the three groups of methods were equal (null hypothesis). The test was performed on each dataset. For F1, the null hypothesis was rejected only for SMD ($p < 0.05$). This indicated that there are no significant differences among the families of methods in four of the datasets, which is consistent with what is observed in Fig. 2 (left).

For AP, the null hypothesis was rejected on SWaT and WADI ($p < 0.05$), being consistent with what is observed in Fig. 2 (right) and indicating no significant differences among the families of methods in three out of the five datasets. We performed a post hoc analysis using a Dunn’s test on SWaT and WADI every category pair. The pairwise analysis showed a significant difference ($p < 0.05$) between DNN and conventional methods in WADI and provided weak evidence to reject the null hypothesis in SWaT between DNN and conventional methods ($0.1 < p < 0.05$). In all other pairwise tests, no significant differences were encountered.

Table 2: Best and worst performing methods, in terms of F1 and AP, for each family of methods F1 and AP per each category of methods.

		Conventional		ML		DNN	
		Worst	Best	Worst	Best	Worst	Best
SWaT	AP F1	VAR/SSA*	PCA MP	DBscan	SVM LOF	DAGMM	USAD
WADI	AP F1	VAR/SSA/MP*	PCA MEWMA	DBscan SVM	LOF IF	DAGMM	USAD
SMD	AP F1	MCUSUM	ICA	DBscan	SVM IF	DAGMM	OA
SMAP	AP F1	MP VAR	MCUSUM ICA	DBscan	SVM LOF	DAGMM	OA USAD
MSL	AP F1	VAR MP	SSA ICA	DBscan SVM	SVM IF	LSTM-VAE DAGMM	USAD

*: No convergence after 10 days of execution; SVM: OC-SVM; OA: Omnianomaly

5.3. Analysis of WADI

We perform a more detailed analysis on the performance of each family of methods. We focus on the WADI dataset, as it is the only dataset where the post hoc analysis identified significant differences between the performance of DNN and the other methods. Overall, WADI contains 14 anomalies distributed over the test set (Figure 3 bottom).

Figure 3 presents the false negatives of all the conventional methods (top), the ML methods (second row) and the DNN approaches (third row). In other words, a value of 1.0 represents an anomaly labeled as normal by a given category of methods. The fourth series (Fig. 3 fourth) presents the false negatives of the conventional approaches which are true positives for the DNNs methods. In other words, a value of 1.0 corresponds to an anomaly detected by at least one DNN algorithm but by no conventional method.

An inspection of Figure 3 shows that four anomalies are not detected by any of the DNNs approaches, while seven anomalies are not detected by the conventional and ML models. Thus, these three anomalies explain the performance gap between the DNNs and conventional methods, which are the second best performing family in terms of AP on this dataset.

Figure 4 presents a detailed view of the first anomaly from the green series. It is caused by the variate 1_MV_001 increasing its value to 2.0 before the variate 1_LT_001 has reached its minimal expected value (40). It is a contextual anomaly, as the separate behavior of each variate does not constitute an anomalous behavior on its own. The remaining two anomalies also shows that they are contextual anomalies. This suggests that the performance gap between DNNs and conventional methods comes from a better detection of contextual anomalies by DNNs approaches.

5.4. Impact of training set size

We use WADI and SWaT datasets to study the impact of the training set size on the performance of the methods. We reduce the size of the training set by keeping the most recent points, i.e. the closest to the test set. We keep 10%

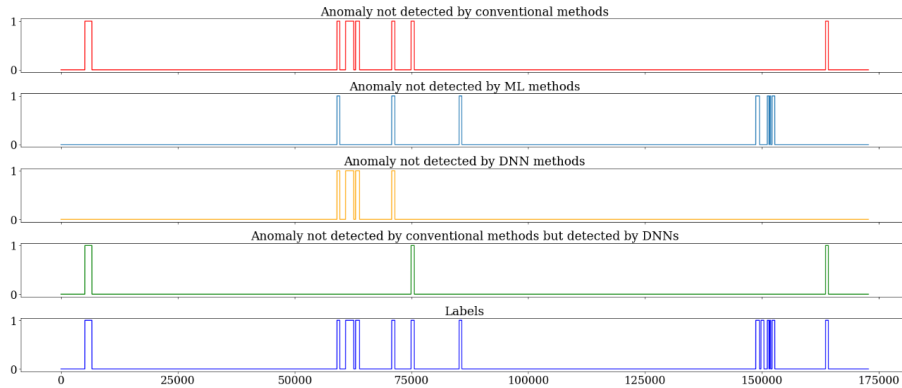


Figure 3: Analysis of WADI. False negatives of conventional methods (top row); false negatives of ML (second); false negatives of DNNs (third); false negatives of conventional methods that are predicted by DNN (fourth) and ground truth labels (last row).

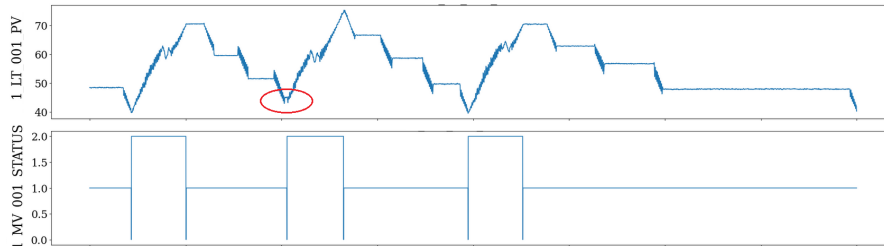


Figure 4: Contextual anomaly in red due to the activation of a motorized valve (1.MV_001) which causes the filling of the tank (1.LT_001) before it reaches the switching threshold, located at 40.

, 25% , 50% and 75% of the original training points. For every set, we train a model per method and assess its performance.

Figure 5 presents the AP obtained on the SWaT dataset and the WADI dataset. We observe that conventional methods globally obtain better results when the dataset size is less than 50% of the original one. For example, on the SwaT dataset, MP, PCA and ICA outperform in terms of AP all ML and DNN methods when the training set is 50% or less. Their performance is matched by OC-SVM at 50%, and it is outperformed only when three quarters of the training dataset is retained. We also observe that the performance of conventional approaches remains relatively constant, despite changes in the training set size, meaning this has little impact in their performance. Instead, ML and DNN methods perform better when increasing the size of the training dataset, except for IF on the WADI dataset which performs slightly better when only 50% of the dataset is kept. This can be explained by the fact that the isolation of anomalous points can be more complex when the training set is larger. In general, it is only above 50% that DNNs approaches seem to perform better

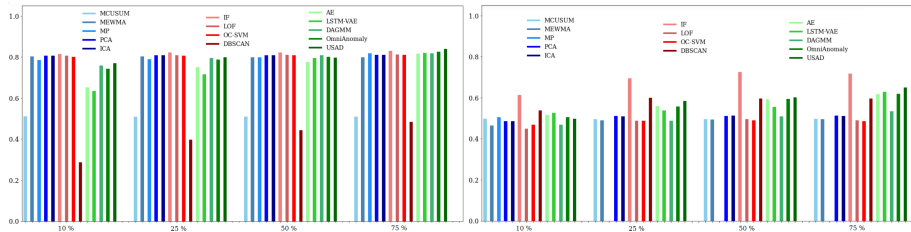


Figure 5: Average precision (AP) on SWaT (left) dataset and WADI (right) datasets.

than conventional methods.

5.5. Discussion

The multivariate anomaly detection performance benchmark across conventional, ML-based and DNN methods and the subsequent statistical analysis of the results (Section 5.2) suggests that the performances across families of methods do not show significant differences in most of the datasets that were considered. In particular, the statistical analyses suggest that DNN methods had a clear superiority in the WADI dataset, which is consistent with our visual analysis of the obtained results (Figure 2).

Given the properties of WADI, a dataset containing contextual anomalies, these results suggest that in the particular case of contextual anomalies, DNN-based methods perform better and may be necessary (Section 5.3). Indeed, contextual anomalies are by definition difficult to detect visually by experts since taken alone the series seem normal, which may justify the need of more complex methods, such as DNNs, capable of extracting and inferring patterns from complex data. However, it is now well established that DNN techniques require large amounts of data to achieve good results. For instance, when we evaluated the performance of the different categories of methods as a function of the training set size (Section 5.4), we observed that DNN techniques observe an important drop in performance, even in those datasets where they are expected to perform best, such as WADI.

In this work we have focused on some aspects of the performance of different types of methods, while leaving aside a detailed analysis of the computational times required by each technique. It is now relatively well established that ML and DNN-based approaches are more computationally demanding than conventional methods. This is mainly due to the fact that training time can be very expensive, especially for DNN-based methods. Conventional methods have the advantage of not requiring a training phase, which is where ML and DNN-based methods consume most of the computational resources. However, at inference time, DNN-based methods can be much faster. As an example, MP is a fast conventional approach [40], which, however, requires to compute the distance of every new sub-sequence w.r.t. previous available time points.

Another important observation from our study regards scalability. Some conventional methods failed to converge when the data set was too large (Table 2). Therefore, the size of the datasets is an important criterion in the choice

of a methods to use. While DNN-based methods are a more suitable choice for larger sets of data, despite their overhead in computational time during training, conventional approaches seem to be a better choice in a small data regime (Section 5.4).

Finally, an important point to consider is also the difficulty to reproduce the results of DNN-based methods compared to the other two categories of methods. Indeed, there is a plethora of open implementations of conventional⁵ and machine learning based methods⁶, while some DNN-based approaches can be difficult to implement, sometimes no implementation is available or it is difficult to set up.

On the basis of these observations, we consider that it is not currently possible to conclude that complex DNN methods should be the de facto choice for multivariate time-series anomaly detection. Instead, we encourage the community to reintegrate ML and conventional methods in the benchmarks to ensure that the new methods proposed improve the performance in the detection of anomalies in time series. This recommendation is inline with the observations of similar studies [17].

6. Conclusion

In this study, we provided a comparative analysis of conventional, ML-based and DNN-based methods. We evaluated the performance of sixteen algorithms on five publicly available benchmark datasets to understand whether the complexity provided by DNN-based approaches is necessary for anomaly detection in multivariate time series. The performance analysis did not allow us to observe a clear and consistent superiority of one category of methods over the others. The DNN based methods seem to perform better when the dataset contains contextual anomalies. However, this finding could only be made on one of the five datasets, so more experimentation is needed to confirm that DNN methods outperform the other categories in terms of contextual anomaly detection. We also studied the impact of the training set size on the performance of these methods. The results show that if the training set is not large enough, the conventional methods outperform the other two categories. Thus, the size of the training set is an important criterion in the choice of the category of methods to detect anomalies in multivariate time series. Some conventional methods have failed to scale on large data sets.

In view of all these results, we have provided the first independent evidence for the hypothesis that much of the recent progress in time series anomaly detection may be illusory [10]. In [10], the authors offer “negative” evidence, noting that most benchmark datasets are unsuitable for making distinctions between algorithms. Our work offers more “positive” evidence for the claim that deep learning has yet to prove real outperformance. We therefore encourage the

⁵PYOD : <https://github.com/yzhao062/pyod>

⁶Scikit Learn : <https://scikit-learn.org/>

community to reincorporate the three categories of methods in the benchmarks of anomaly detection in multivariate time series. Moreover, it seems essential to us to multiply the number of datasets compared in the benchmarks in order to ensure that all eventualities are covered. For this, the community will have to obtain new real world datasets containing contextual anomalies. Indeed, the difficulty for experts to visually label contextual anomalies in multivariate time series makes it difficult to obtain test sets covering this criterion. It is then complicated to assert that DNN methods are necessary although it seems that they are able to outperform conventional approaches when there are contextual anomalies, as long as the data set is large enough. We hope that this study will help guide researchers in their choice and the assessment of methods for detecting anomalies in multivariate time series.

Acknowledgments

Maria A. Zuluaga is partially funded through the 3IA Cte d'Azur Investments in the Future project managed by the National Research Agency (ANR) (ANR-19-P3IA-0002).

References

- [1] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M. A. Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404.
- [2] T. Yin, C. Liu, F. Ding, Z. Feng, B. Yuan, N. Zhang, Graph-based stock correlation and prediction for high-frequency trading systems, *Pattern Recognition* 122 (2022) 108–209.
- [3] B. Garca-Martnez, A. Fernandez-Caballero, R. Alcaraz, A. Martinez-Rodrigo, Assessment of dispersion patterns for negative stress detection from electroencephalographic signals, *Pattern Recognition* 119 (2021) 94–108.
- [4] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, *ACM Computing Surveys (CSUR)* 41 (2009) 1–58.
- [5] R. Domingues, M. Filippone, P. Michiardi, J. Zouaoui, A comparative evaluation of outlier detection algorithms: Experiments and analyses, *Pattern Recognition* 74 (2018) 406–421.
- [6] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (2015) 436–444.
- [7] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep autoencoding gaussian mixture model for unsupervised anomaly detection, in: Proceedings of the International Conference on Learning Representations (ICLR), 2018.

- [8] J. Audibert, Peer-reviewed deep learning-based methods for anomaly detection in multivariate time series from 2018 to 2021 (2021). URL: <https://www.doi.org/10.6084/m9.figshare.16536144.v1>.
- [9] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837.
- [10] R. Wu, E. J. Keogh, Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress, 2020.
- [11] V. Antun, F. Renna, C. Poon, B. Adcock, A. C. Hansen, On instabilities of deep learning in image reconstruction and the potential costs of AI, National Academy of Sciences 117 (2020) 30088–30095.
- [12] D. Heaven, Why deep-learning ais are so easy to fool, Nature 574 (2019) 163–166.
- [13] S. Jiao, Y. Gao, J. Feng, T. Lei, X. Yuan, Does deep learning always outperform simple linear regression in optical imaging?, Optics Express 28 (2020) 3717–3731.
- [14] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The m4 competition: Results, findings, conclusion and way forward, International Journal of Forecasting 34 (2018) 802–808.
- [15] W. Fu, K. Breininger, R. Schaffert, Z. Pan, A. Maier, keep it simple, scholar: an experimental analysis of few-parameter segmentation networks for retinal vessels in fundus imaging, International Journal of Computer Assisted Radiology and Surgery 16 (2021) 967–978.
- [16] A. Garg, W. Zhang, J. Samaran, R. Savitha, C.-S. Foo, An evaluation of anomaly detection and diagnosis in multivariate time series, IEEE Transactions on Neural Networks and Learning Systems (2021) 1–10.
- [17] S. Makridakis, E. Spiliotis, V. Assimakopoulos, The m4 competition: 100,000 time series and 61 forecasting methods, International Journal of Forecasting 36 (2020) 54–74.
- [18] L. Breiman, et al., Statistical modeling: The two cultures, Statistical Science 16 (2001) 199–231.
- [19] W. H. Woodall, M. M. Ncube, Multivariate cusum quality-control procedures, Technometrics 27 (1985) 285–292.
- [20] C. A. Lowry, W. H. Woodall, C. W. Champ, S. E. Rigdon, A multivariate exponentially weighted moving average control chart, Technometrics 34 (1992) 46–53.

- [21] X. Pan, J. Jarrett, Using vector autoregressive residuals to monitor multivariate processes in the presence of serial correlation, *International Journal of Production Economics* 106 (2007) 204–216.
- [22] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, L. Chang, A novel anomaly detection scheme based on principal component classifier, Technical Report, University of Miami, 2003.
- [23] J. E. Jackson, G. S. Mudholkar, Control procedures for residuals associated with principal component analysis, *Technometrics* 21 (1979) 341–349.
- [24] N. Golyandina, A. Zhigljavsky, *Singular Spectrum Analysis for Time Series*, 2013.
- [25] L. Zonglin, H. Guangmin, Y. Xingmiao, Multi-dimensional traffic anomaly detection based on ica, in: *Proceedings of the IEEE Symposium on Computers and Communications*, 2009, pp. 333–336.
- [26] M. Reza, Multivariate outlier detection using independent component analysis, *Science Journal of Applied Mathematics and Statistics* 3 (2015) 171–176.
- [27] B. Xia, *Similarity search in time series data sets*, Ph.D. thesis, Simon Fraser University, 1997.
- [28] C.-C. M. Yeh, Y. Zhu, L. Ulanova, N. Begum, Y. Ding, H. A. Dau, D. F. Silva, A. Mueen, E. Keogh, Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets, in: *Proceedings of the 2016 IEEE International Conference on Data Mining (ICDM)*, 2016, pp. 1317–1322.
- [29] M. Linardi, Y. Zhu, T. Palpanas, E. Keogh, Matrix profile goes mad: variable-length motif and discord discovery in data series, *Data Mining and Knowledge Discovery* 34 (2020) 1022–1071.
- [30] F. T. Liu, K. M. Ting, Z.-H. Zhou, Isolation forest, in: *Proceedings of the 2008 IEEE International Conference on Data Mining (ICDM)*, 2008, pp. 413–422.
- [31] M. M. Breunig, H.-P. Kriegel, R. T. Ng, 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.
- [32] S. Oehmcke, O. Zielinski, O. Kramer, Event detection in marine time series data, in: *Proceedings of the Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, 2015, pp. 279–286.
- [33] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: *Proceedings of the Second ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1996, p. 226231.

- [34] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. Smola, R. Williamson, Estimating support of a high-dimensional distribution, *Neural Computation* 13 (2001) 1443–1471.
- [35] J. Ma, S. Perkins, Time-series novelty detection using one-class support vector machines, in: *Proceedings of the International Joint Conference on Neural Networks*, 2003, pp. 1741–1745.
- [36] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Learning Internal Representations by Error Propagation*, MIT Press, 1986, p. 318362.
- [37] D. Park, Y. Hoshi, C. C. Kemp, A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder, *IEEE Robotics and Automation Letters* 3 (2018) 1544–1551.
- [38] S. Lin, R. Clark, R. Birke, S. Schnborn, N. Trigoni, S. Roberts, Anomaly detection for time series using vae-lstm hybrid model, in: *Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4322–4326.
- [39] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, et al., Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications, in: *Proceedings of the 2018 World Wide Web Conference*, 2018, pp. 187–196.
- [40] Y. Zhu, C.-C. M. Yeh, Z. Zimmerman, K. Kamgar, E. Keogh, Matrix profile xi: Scrimp++: time series motif discovery at interactive speeds, in: *Proceedings of the 2018 IEEE International Conference on Data Mining (ICDM)*, 2018, pp. 837–846.