Data Science and Engineering
https://doi.org/10.1007/s41019-025-00295-w

RESEARCH PAPERS

Embedding Models for Multivariate Time Series Anomaly Detection in Industry 5.0

Lorenzo Colombi[1] · Michela Vespa[1] · Nicolas Belletti[1] · Matteo Brina[1] · Simon Dahdal[1] · Filippo Tabanelli[1] · Francesco Resca[1] · Elena Bellodi[1] · Mauro Tortonesi[1] · Cesare Stefanelli[1] · Massimiliano Vignoli[2]

Received: 9 December 2024 / Revised: 18 April 2025 / Accepted: 21 April 2025
© The Author(s) 2025

Abstract

Industrial processes often involve the generation—and the analysis—of multivariate time series data, which poses several challenges from the anomaly detection perspective. In addition to the need to detect previously unseen anomalies, the high dimensionality of industrial datasets introduces the complexity of simultaneously analyzing multiple features and their interactions. Finally, industrial datasets are typically highly imbalanced, with minimal information on anomalous processes. To address these issues, we propose a novel anomaly detection framework that introduces two embedding models, based on Time2Vec and Discrete Wavelet Transforms, leveraging their capabilities to represent multivariate time series as vectors while capturing and preserving temporal dependencies and combining them with several classifiers to enhance the overall performance of anomaly detection. We tested our solution using a publicly available benchmark dataset and a real industrial use case, particularly data collected from a Bonfiglioli gear manufacturing plant. The results demonstrate that, unlike traditional reconstruction-based autoencoders, which often struggle with sporadic noise, our embedding-based solutions maintain high performance across various noise conditions.

Keywords Anomaly detection · Multivariate time series · Time2 Vec · DWT · Embedding · Machine learning · Industry 5.0

List of abbreviations

AE	Autoencoder
AD	Anomaly detection
CNN	Convolutional neural network
D-IF	Deep isolation forest
DeepSVDD	Deep support vector data description
DTW	Dynamic time warping
DWT	Discrete wavelet transform
IF	Isolation forest
LOF	Local outlier factor
ML	Machine learning
MRA	MultiResolution analysis
MTS	Multivariate time series
OCSVM	One-class support vector machine

✉ Lorenzo Colombi
lorenzo.colombi@unife.it

Michela Vespa
michela.vespa@unife.it

Nicolas Belletti
nicolas.belletti@unife.it

Matteo Brina
matteo.brina@unife.it

Simon Dahdal
simon.dahdal@unife.it

Filippo Tabanelli
filippo.tabanelli@unife.it

Francesco Resca
francesco.resca@unife.it

Elena Bellodi
elena.bellodi@unife.it

Mauro Tortonesi
mauro.tortonesi@unife.it

Cesare Stefanelli
cesare.stefanelli@unife.it

Massimiliano Vignoli
massimiliano.vignoli@bonfiglioli.com

[1] University of Ferrara, Ferrara, Italy

[2] Bonfiglioli S.P.A., Via Cav. Clementino Bonfiglioli, 1, Calderara di Reno, Italy

Published online: 24 June 2025

pywt	PyWavelets
T2 V	Time2Vec
TS	Time series

1 Introduction

Machine Learning (ML) and MLOps solutions are increasingly being adopted to optimize processes in different application scenarios, including social networks Alom et al. [3], healthcare Tartarisco et al. [42], energy management Bevilacqua et al. [8], smart cities Forkan et al. [20], human assistance and disaster recovery Colombi et al. [13]. In this context, digital manufacturing represents one of the most promising fields for ML innovations that use data-driven insights to refine production methods, improve product quality and increase operational efficiency Dahdal et al. [18].

In industrial applications, significant productivity-boosting and, consequently, competitive advantages can be gained through the timely and accurate identification of anomalies that can indicate equipment malfunctions, process inefficiencies, or potential safety hazards Colombi et al. [12]. Anomaly Detection (AD), therefore, represents a key enabler for maintaining operational efficiency, implementing predictive maintenance, and preventing costly failures. This is even more important as we are rapidly advancing toward the Industry 5.0 paradigm and its challenging Zero Defect Manufacturing (ZDM) and Zero Waste Manufacturing (ZWM) objectives, aiming for more sustainable processes Venanzi et al. [43].

Traditional AD methods, such as Recurrent Neural Networks or reconstruction-based approaches, often struggle with Multivariate Time Series (MTS) characterized by complex temporal dependencies and cross-variable interactions Tank, [41]. For this reason, industrial settings need robust tools that implement AD for MTS.

In particular, anomaly detection for multivariate time series, which involves observing multiple variables over time, is increasingly becoming a critical research challenge in industrial applications. First of all, the nature of industrial processes often involves having to deal with previously unseen anomalies, which makes it difficult to rely on supervised learning methods. On top of that, the high dimensionality of MTS poses additional complexities that emerge from the need for the simultaneous analysis of multiple features - and their interactions. Finally, the industrial environment usually presents issues related to data quality - especially in terms of imbalanced datasets that contain a lot of information for normal processes and a much lower amount of information for anomalous ones.

To address these challenges, various signal processing and ML techniques have been proposed. While embedding techniques Bertrand et al. [10] have shown promise in representing MTS data in a transformed space, advances in Wavelet Transforms Meyer, [30] have created new opportunities for more robust AD Rhif et al. [35]; Pacheco et al. [33], offering an alternative to classical methods like short-time Fourier transforms, which cannot simultaneously achieve high resolution in both time and frequency domains. In particular, Discrete Wavelet Transforms (DWTs) Mallat, [29] provide a powerful tool for analyzing non-stationary signals typical in industrial settings. This multi-resolution analysis capability allows for better detection of anomalies across different time scales and frequencies Zhang et al. [48]; Kanarachos et al. [24]. Autoencoders (AEs) have emerged as another powerful technique for creating embeddings of MTS Ienco and Interdonato, [23]. When combined with DWT, it is possible to leverage both the temporal-scale decomposition properties of DWT and the dimensionality reduction capabilities of AEs. This hybrid approach enables a more effective capture of complex patterns and anomalies in industrial Time Series (TS). Jointly with AEs, various ML algorithms are commonly used for anomaly detection. For example, one-class classification could be used to detect anomalies in the dataset embedding created by the autoencoder.

In this paper, we build on the previous work Colombi et al. [16] and propose a novel framework for AD in MTS within an Industry 5.0 setting, combining DWT-based and Time2 Vec-inspired Peñaloza, [34] autoencoder architectures with many ML algorithms used to perform AD. Our approach integrates DWT decomposition for multi-scale analysis with embeddings that preserve temporal information, significantly improving the AD process. We employ various ML models for AD on the resulting embeddings, including a One-Class Support Vector Machine, Isolation Forest, Deep Isolation Forest, Deep Support Data Description, and Local Outlier Factor. We validated this approach using both a well-known public dataset, KDD Cup [25], and a real industrial case study provided by Bonfiglioli, a global leader in gear motors and drive systems manufacturing. The KDD-99 dataset was used as a standard dataset for anomaly detection, ensuring reproducibility and benchmarking across different models. In contrast, the Bonfiglioli dataset demonstrated the use of the proposed solution in an industrial setting, helping to solve a real-world problem related to anomaly detection in MTS from production processes. In particular, the objective of the Bonfiglioli use case was to detect anomalies in the gear production process by analyzing TS data from various sensors installed on the machinery. We evaluated our novel solution against a traditional reconstruction-based approach and a convolutional AE used to extract embeddings, which served as the baseline. Our results demonstrate that the Time2 Vec-based and DWT-based AEs not only match but, in many cases, significantly surpass the performance of the baseline model.

The primary contributions of this paper are: (i) we develop two AE-based AD solutions, a DWT-based and a Time2 Vec-based one, for generating high-dimensional embeddings from multivariate TS; (ii) we evaluate the effectiveness of these approaches in detecting anomalies using various established ML techniques on the generated embeddings in a real industrial environment and on a publicly available dataset (KDD-99).

The paper is organized as follows: Sect. 2 provides background and related work on MTS AD. Section 3 describes the industrial use case at Bonfiglioli. Section 4 describes our two novel approaches. Section 5 presents the experimental evaluation of our models. Section 6 concludes the paper and outlines future work.

2 Background and Related Work

AD in TS data Schmidl et al. [40] is essential for maintaining efficiency and safety in various contexts. An anomaly can be either an isolated observation or a sequence of observations that substantially differ from the normal distribution. The unpredictable nature of anomalies adds another layer of complexity. Unlike regular predictive modeling where events or outcomes are known and defined, anomalies can often show entirely new behaviors or patterns that have not been observed or labeled before. This unpredictability requires robust and generalizable algorithms to detect deviations that fall outside the range of historical data used for training. Compared to univariate TS, MTS introduces additional complexities that must be accounted for, such as the limited availability of tools and the necessity of considering both spatial dependencies between variables as well as temporal dependencies. In analyzing large sets of MTS, it is also crucial to identify interactions between different features. Traditional linear autoregressive models are often insufficient, necessitating new approaches to handle non-linear and non-stationary TS interactions Tank, [41]. Creating effective AD models is very challenging due to the complexities of TS data. Another important issue is the imbalance in training data, where instances of normal operations greatly outnumber rare occurrences of anomalies. This makes it hard for ML models to accurately learn about anomalous behaviour since there is not enough anomalous data for effective training.

2.1 Unsupervised Deep Learning for MTS Anomaly Detection

Unsupervised deep learning methods operate without labelled examples of normal or anomalous states, making them ideal for scenarios where anomalies are rare or have not been previously identified Siegel [39]. Several approaches have been developed to address MTS anomaly detection, including forecasting-based and reconstruction-based models, each offering unique strengths in capturing temporal dependencies and identifying deviations from normal patterns.

Forecasting-based models predict future values using historical and current data, identifying anomalies through discrepancies between predicted and actual values. Recurrent Neural Networks (RNNs) and Long-Short Term Memory Networks (LSTMs) are preferred for their efficacy in managing lengthy interrelated sequences and modelling intricated temporal dynamics, attributed to their ability to maintain memory over time. LSTM-based AD for multivariate TS data was first proposed by Malhotra et al. [32] by stacking LSTM networks. Other applications of LSTM network architectures have been explored, such as AD-LTI Wu et al. [44]. However, these models can struggle to detect new or unseen anomalies because they rely heavily on historical data for predictions. While they are effective at identifying anomalies that resemble past patterns, they may fail to recognize novel anomalies that differ significantly from what they have previously encountered and struggle with interdependencies in data Wu et al. [46].

Reconstruction-based deep learning models for AD operate on the principle that normal patterns within the data can be learned and reconstructed accurately, whereas anomalies cannot be reconstructed well and thus will show significant deviations when the model attempts to reconstruct them. In particular, models for normal behaviour are constructed by encoding subsequences of normal training data in latent spaces. In the test phase, the model cannot reconstruct anomalous subsequences since it is only trained on normal data. As a result, anomalies are detected by reconstructing a point/sliding window from test data and comparing them to the actual values, by calculating a reconstruction error.

These methods have gained widespread attention, particularly in scenarios where labeled anomaly data is scarce or unavailable, they have become a crucial approach for unsupervised anomaly detection, enabling models to learn normal patterns and identify deviations without requiring explicit supervision Miao et al. [31].

Convolutional AEs have been extensively adopted in prior literature due to their ability to capture temporal dependencies and provide a robust baseline for anomaly detection tasks. This established effectiveness motivated its selection as a representative state-of-the-art baseline for comparative evaluation. They consist of two main components: an encoder that compresses the input data into a lower-dimensional representation and a decoder that attempts to reconstruct the input data from this compressed form. The goal is to minimize the difference between the original input and its reconstruction, which helps to identify anomalies when the reconstruction error is notably high. Some implementations

of AE-based architectures for detecting anomalies in TS include CAE Chevrot et al. [17] and USAD Audibert et al. [5].

Hybrid methods for AD in TS data involve combining various techniques to enhance the overall performance of the detection system. These methods typically integrate classical statistical approaches, ML models, and deep learning architectures to handle the intricacies of TS data, such as trends, seasonality, and noise. They often blend a forecasting-based model with a reconstruction-based model to obtain improved TS representations. Examples are CAE-M Zhang et al. [47] and Hu et al. [21]. In Zhang et al. [47], authors propose a solution that can model generalized patterns based on normalized data by undertaking reconstruction and prediction simultaneously. This is achieved through the combination of a convolutional AE and LSTMs.

2.2 Embedding-Based Anomaly Detection

Many of the mentioned methods conduct unsupervised AD on MTS data. However, they rely on raw TS input for this task, meaning that AD is performed on the raw data itself rather than on a representation of the data. Methods like Bonifati et al. [6] aim to create a representation for MTS based on feature extraction and selection. Recent advances have focused on developing more sophisticated representations for MTS analysis. Embeddings offer dense vector representations that capture inherent structure and relationships within the data, potentially providing a more refined view of normal versus abnormal patterns. There is currently limited research on AD performed on embeddings, primarily due to the complexities of generating and interpreting embeddings for MTS. In the context of MTS, embeddings not only need to represent the individual data points but also their sequence and temporal context. Higher-dimensional embeddings can be particularly beneficial in MTS data due to the complexity and richness of the data involved Echihabi et al. [19]. Higher-dimensional embeddings can capture complex feature relationships more effectively, preserving the essential characteristics of the data that might be lost in lower-dimensional representations. However, the use of higher-dimensional embeddings also introduces significant challenges. The primary concern is the curse of dimensionality Aremu et al. [4], which arises because the vector space grows substantially with the addition of each dimension. As the number of dimensions increases, the data distribution across the space becomes increasingly sparse. This sparsity is problematic because the data points are spread out over a large volume, and there is less likelihood that any two points are close to each other. Consequently, traditional distance metrics become less meaningful, making it harder for ML models to discern patterns or make accurate predictions. Previous efforts have used AEs to generate general-purpose embeddings for domain-agnostic problems Bertrand et al. [10]. In Ienco and Interdonato, [23], the proposed framework utilizes a two-stage process that combines a recurrent AE with attention and gating mechanisms to generate embedding representations of multivariate time series and subsequently perform clustering.

2.3 Time-Aware Representations of MTS

Recent research in MTS AD has focused on modelling both temporal dependencies across sequential observations and correlation structures between variables, as anomalous patterns can manifest through deviations in either temporal evolution or cross-variable relationships of the system.

Time-aware representation techniques like Time2 Vec (T2 V), presented in Kazemi et al. [26], have been proposed to address the unique characteristics of time-dependent data, w.r.t. Univariate TS. T2 V encodes time into vectors capturing both linear and periodic characteristics, enabling models to recognize temporal dynamics.

For each time instance τ, T2 V of τ, denoted as $T2V(\tau)$, is a vector of size $k+1$, defined as follows:

$$T2V(\tau)[i] = \begin{cases} \omega_i \tau + \phi_i & \text{if } i = 0, \\ F(\omega_i \tau + \phi_i) & \text{if } 1 \leq i \leq k, \end{cases}$$

where $T2V(\tau)[i]$ is the ith element of $T2V(\tau)$, F is a periodic activation function, and ω and ϕ are learnable parameters.

The application of T2 V as an embedding technique allows for a more refined and structured representation of Univariate TS data. Thus, T2 V serves as a bridge between traditional TS analysis and advanced AD strategies that exploit the rich information contained in embeddings.

On the other hand, a significant advancement in MTS analysis comes from the application of Discrete Wavelet Transform to perform AD. DWT provides a mathematically rigorous framework for analyzing TS dependencies by decomposing the multivariate signal into time-scale representations, enabling the characterization of both localized temporal dynamics and scale-dependent variable interactions through their coefficient structures. This enables MultiResolution Analysis (MRA), in which a signal is represented simultaneously at multiple scales or resolutions, where each scale captures different levels of detail in the data.

A particularly interesting application of MRA is the DWT, which recently has gained attention in the field of MTS AD for their ability to decompose signals into multiple frequency scales, helping isolate anomalous patterns within specific bands. From a single “mother wavelet"$\psi(t)$, a family of “daughter wavelets" can be generated through dilations and translations. In the DWT, dilation and translation parameters are typically constrained to powers of two, allowing the

wavelet to be shifted and scaled, providing an analysis of the signal at different resolutions:

$$\psi_{j,k}(t) = 2^{-j/2}\psi\left(\frac{t-k}{2^j}\right)$$

Where j and k are integers that control the scale and translation, respectively. This results in a computationally efficient algorithm, often implemented using the pyramid method. The pyramid algorithm is a structured approach to decomposing a signal into various levels of detail: using a pair of Finite Impulse Response (FIR) filters as a basic decomposition module, it is possible to create a sequence of high-pass and low-pass filters (filter-bank), where the transfer functions of the filters are determined based on the mother wavelet function. The MEGA framework Wang et al. [45] leverages this by integrating DWT with graph-based AEs, enabling effective multiscale anomaly detection that captures temporal and inter-variable dependencies. However, this approach is limited to end-to-end training on specific datasets, potentially resulting in false positives or false negatives in context-sensitive scenarios.

3 Industrial Use Case

Bonfiglioli[1] is a prominent manufacturer with decades of experience in designing and producing a wide array of gear motors, drive systems, planetary gear motors, reducers, and inverters. The company is a leader in power transmission production and is increasingly adopting Industry 5.0 best practices, focusing on implementing efficient and environmentally sustainable processes. One of the main manufacturing plants is located in Forlì (Italy) and is entirely dedicated to the production of axles and gearboxes for electric vehicles.

In this plant, the gear manufacturing process process begins with fusion, followed by stamping, turning, and spline gear cutting. Afterwards, external processing includes hardening and carburizing, which is succeeded by internal processing phases such as diameter grinding and tooth grinding. In this work, we mainly focus on the gear-cutting phase by the Gleason P800 machine, a hobbing machine designed for pinion shafts with external gearing rough-cut using a disk cutter with inserts. The pinions produced are primarily destined for the wind energy sector, specifically for gear reducers.

Data collected from this machine allowed us to train an AD model to identify irregularities in the gear-cutting phase. As a critical component in Bonfiglioli's manufacturing line, this machine requires continuous monitoring to ensure high precision and quality in gear production. To achieve this, Bonfiglioli collected extensive TS data from various sensors installed on the machine. At Bonfiglioli, domain experts prioritize minimizing false positives in anomaly detection systems for industrial processes. This strategic focus stems from maintaining operational efficiency and avoiding unnecessary disruptions. When an anomaly detection system erroneously flags normal operational conditions as anomalous (i.e., as false positives), it can trigger a series of costly interventions that undermine manufacturing productivity. Each false positive can lead to unnecessary production halts, downtime and inspections, which can translate directly into economic losses. A single false alarm might cause line managers to pause critical production, conduct unnecessary equipment inspections, or initiate costly troubleshooting procedures, disrupting the production workflow.

Each monitored Bonfiglioli manufacturing process was recorded in a CSV file for a total of one million rows. Each row represents a specific time instance, and for each second, 104 different measurements were taken. Therefore, each MTS was originally made of 104 dimensions. The selection of 6 key features from the original 104 measurements in the Bonfiglioli dataset was driven by domain experts' requirements. However, unsupervised feature selection techniques could be applied as an alternative to reduce dependency on manual expert input Solorio-Fernández et al. [38]. These specific measurements capture the load on various movable components within the system, including the piece holder table, the spindle, and the radial, tangential, and axial slides, which are responsible for the movement along different axes. Concentrating on this subset of the available data allowed us to tackle the inherent challenges of MTS analysis.

Next, we checked the presence of NaN values and deleted duplicated rows and outliers. The forward-filling method was employed to substitute NaN values with the value from the previous row. Regarding outliers, it was important to remove them due to the unlabeled nature of the dataset, which may contain noise or anomalies. To reach this goal, we deleted rows where at least one value was below the first or above the third quantile, calculated on the single column. Some files contained only 90 rows, while others contained over 10,000 due to the varying duration of the manufacturing process. Since data are collected each second, we had to manage observation files spanning from a minimum of 90 s to a maximum of 160 min.

Since splitting a single registered process without losing fundamental information was not possible, to shorten the MTSs in the longer files, we resampled the MTSs aggregating rows belonging to the same time window into a single value, calculating the average and preserving only the time index of the last row in the time window. The dataset was split into multivariate fixed-size TS of 100 s. The examples shorter than 100 s were extended by applying constant

[1] https://bonfiglioli.com.

padding, specifically prolonging the last value. After preprocessing, the final dataset consisted of 2950 6-feature MTS, with each one spanning 100 s. This limited number of examples raised a problem of data scarcity, which can jeopardize the model training process.

To recap, the task involved navigating several data-related challenges inherent to the aforementioned dataset. Firstly, we report data scarcity, as the dataset contained, after processing, only 2950 MTSs. Secondly, the dataset was completely unlabeled, which likely included some anomalous instances. This reflects a common characteristic of real-world manufacturing datasets, where data annotation often requires manual effort for this task. Finally, the multivariate nature of TS introduced additional complexity in developing an effective model. These challenges highlighted key considerations in addressing the detection of production process anomalies.

4 A Novel Approach for Embedding Extraction from MTS

To address the challenges presented in Sect. 3 related to the real industrial scenario, we developed two alternative AE-based architectures capable of extracting informative embeddings from MTS.

Our goal was to design models that maximize precision in anomaly detection. However, experimental results, presented in Sec. 5, showed that neither architecture consistently outperformed the other across every dataset.

The first architecture is called "**Ad-hoc Time2 Vec-based Autoencoder**" and is a specifically designed AE with a Time2 Vec-inspired layer capable of embedding spatial and temporal dependencies into the vector, as described in Sect. 4.1.

The second solution, presented in Sect. 4.2, designed in two variants, is called "**Ad-hoc DWT-based Autoencoder**" and involves an additional step to apply the representation change to the input data, encoding the signal by the coefficients of its DWT.

For both solutions, the initial step involved designing an embedding neural network capable of capturing a detailed and comprehensive representation of the data. This model was specifically designed to preserve both the spatial (cross-variable) and temporal characteristics of the MTS. Subsequently, AD was performed using a range of one-class classification machine learning algorithms.

4.1 Ad-hoc Time2 Vec-Based Autoencoder

The T2 V-based AE architecture is shown in Fig. 1 and aims to maintain a detailed and informative representation where both spatial and temporal attributes are preserved.

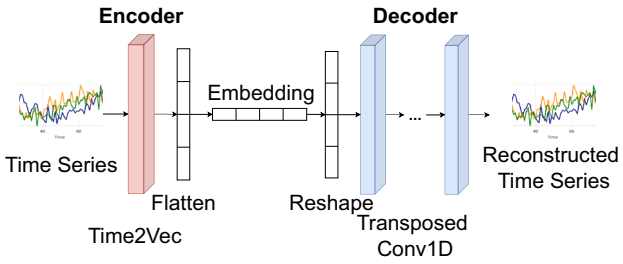

Fig. 1 Ad-hoc T2 V-based AE architecture

Furthermore, as shown in Kazemi et al. [26], vector representations of time must capture both periodic and non-periodic patterns, invariant to time rescaling and simple enough to be used in different architectures. Therefore, drawing inspiration from Peñaloza [34], where T2 V is utilized as an embedding layer within a larger architectural framework, we have developed a novel AE-based architecture designed to extract vector representations from MTS data.

As highlighted in Sect. 2, T2 V was originally proposed to produce a better representation of time by computing a vector using the scalar notion of time as input and is limited to the univariate case. However, we can assume that our TS includes an implicit notion of time, enabling us to use the TS values rather than the time. Building upon this, we sought to extend the T2 V approach beyond its original single-dimensional limitation, developing a method to generate embeddings for MTS. We implemented this AE-based architecture in Python, using the TensorFlow[2] framework. We created a custom layer called Time2 Vec, designed to be integrated into more complex models. This layer combines linear and sinusoidal transformations to effectively capture temporal patterns, enabling a rich and adaptable representation of temporal information, as shown in Eq. 1.

To adapt the original T2 V architecture to the MTS problem, we developed a novel approach for embedding generation. Consider a MTS X of dimension N×F, where N is the number of steps and F is the number of features, and an embedding dimension K, where K is a hyper-parameter: we use function $T2V_{MTS}(X)$ to create a matrix representation, which is subsequently flattened into a vector, where each column i is defined as follows:

$$T2V_{MTS}(X)[i] = \begin{cases} X\,\omega_0 + b_0 & \text{if } i = 0 \\ \sin\left(X\,\omega_i + b_i\right) & \text{if } 1 \leq I < K \end{cases} \quad (1)$$

where the sin (·) function operates element-wise, w_0 is a vector of weights (F×1) and b_0 is a vector of biases (N×1). w_i are matrices of weights of dimension F×(K-1), b_i are matrices of

[2] https://www.tensorflow.org/.

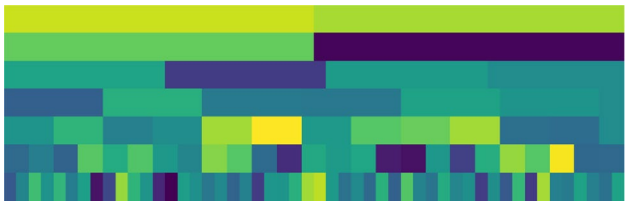

Fig. 2 Scalogram representation of a signal of 100 samples. In each row, each array of coefficients is arranged to reach the length of the longest one

biases (N×(K-1)). By creating the embedding in this way, it is possible to capture both non-periodic patterns ($Xw_0 + b_0$) and periodic patterns ($\sin(Xw + b)$). Also by using the sine function, we make the embedding invariant to time rescaling Kazemi et al. [26].

The output of this layer is flattened to obtain a 1-D vector representing the original TS. Therefore, the Decoder comprises a reshaping layer and a series of 1-D convolutional layers that reconstruct the original TS starting from the previously computed embedding.

4.2 Ad-hoc DWT-Based Autoencoder

We decided to preprocess the MTS using the DWT algorithm's efficiency, fastening the representation change. Two different variants of the DWT-based AE were implemented, depending on the arrangement of the DWT coefficients: a one-dimensional variant using the flattened coefficients and a two-dimensional one, obtained by arranging them as an image called "scalogram". An example of a scalogram is shown in Fig. 2.

As discussed in Sect. 2, the DWT is efficiently implemented using a filter bank approach, where the input signal is passed through a sequence of high-pass and low-pass filters to decompose it into multiple frequency bands, both cutting on/off at half the sampling frequency and followed by downsampling to half the original rate. This method efficiently computes the DWT by approximating the signal at different decomposition levels, as shown in Fig. 3. Recursively adding this module to the low-frequency component, we obtain a filter bank able to apply DWT to the input signal, resulting in N+1 output signals, where N is the "decomposition level" or number of decomposition modules. Each output sampling rate determines its number of coefficients.

Our approach aims to extend the use of AEs beyond their traditional reconstruction tasks to establish a more flexible framework for learning generalizable MTS representations. Rather than limiting the architecture to specific AD methods, we develop an embedding space that can serve as input for various downstream ML techniques. This generalization allows the learned representations to be easily adaptable in different AD approaches by simply interfacing with their

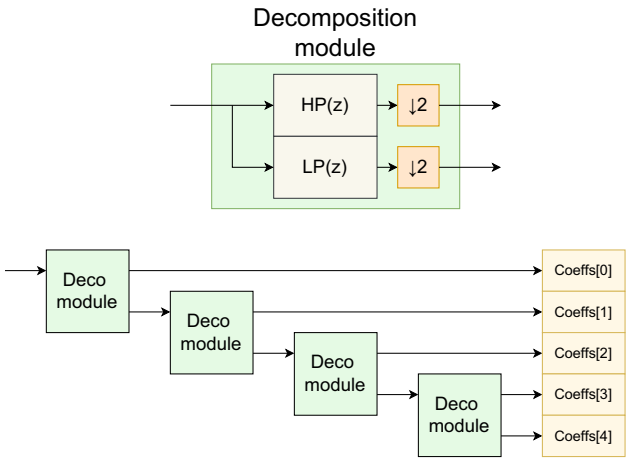

Fig. 3 Pyramid algorithm, performing a DWT using a bank of filters. High-pass (HP) and low-pass (LP) are the transfer functions for the FIR filters. The number of coefficient sets is equal to one plus the decomposition level

respective detection mechanisms without requiring architectural modifications of the core representation learning component.

Once the coefficients have been calculated, they must be normalized and arranged in a form that can be fed to a Convolutional Neural Network (CNN). The whole AE is illustrated in Fig. 4, where the two types of encodings can be used as input for the AE: both the one-dimensional concatenation of the coefficient arrays and the two-dimensional scalogram.

In both cases, the DWT representations are produced for each feature of each multivariate signal, leading to multichannel representations.

Each representation of the signal, as concatenated coefficients or scalograms, is fed as input to a CNN. The two networks share both processes and general structure. The encoder consists of repeated convolution and batch normalization layer pairs, followed by a flattening and a dense layer, resulting in the final embedding. The decoder is symmetrical, except for the absence of the batch normalization layers. The only difference between the two architectures lies in the convolution employed: since the concatenated representation is sequence-like, one-dimensional convolutions and their transposes have been used. For scalograms, instead, two-dimensional convolutions were applied.

The implementation relies on the PyWavelets (pywt)[3] Python module and its `wavedec` method to produce the coefficient arrays. The mother wavelet function chosen is `db1`, from the Daubechies wavelets family. The chosen level of decomposition is the integer part of the base 2

[3] https://pywavelets.readthedocs.io.

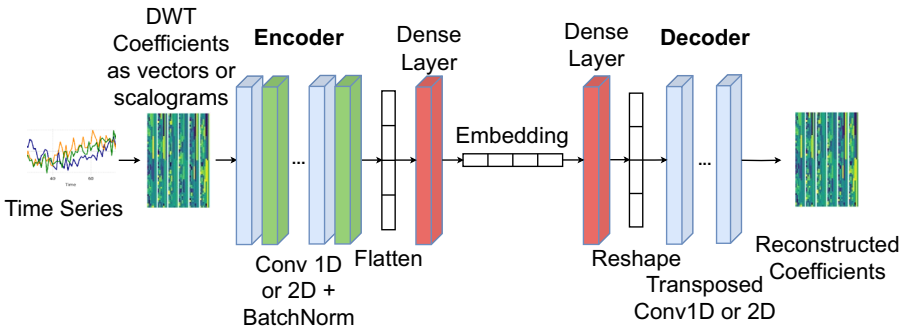

Fig. 4 Architectures for the DWT-based autoencoders with coefficients arranged as a vector or a scalogram

logarithm of the number of samples in the sequence. The method used to select the correct decomposition level is based on the PyWavelet documentation and guarantees a faithful signal representation and the possibility of reconstruction.

5 Experimental Evaluation

We conducted several experiments to evaluate the scalability and performance, measuring training and inference time, of our custom-designed AEs in detecting anomalies.

We used two datasets: an open-source dataset with naturally occurring anomalies and a real-world industrial dataset with synthetic anomalies introduced since the latter was unlabeled and it was assumed that it did not include them. The focus of our evaluation was on the precision of the AE models in detecting these anomalies across both datasets.

The results are obtained by using different AEs to extract MTS embeddings. We then used these embeddings jointly with ML algorithms to perform binary classification. In detail, we chose different state-of-the-art one-class classification algorithms to detect anomalies within the embedding test sets Seliya et al. [36]; Khan and Madden, [28]: Local Outlier Factor (LOF), Deep Isolation Forest (D-IF), Isolation Forest (IF), One-Class Support Vector Machine (One-Class SVM), and Deep Support Vector Data Description (Deep SVDD).

Additionally, for comparison purposes, we compared such results with a widely used non-embedding-based method, where AE reconstruction error has been used to identify anomalies. In this case, the reconstruction error threshold is set to the maximum value in the training set, measured using three different loss functions: Dynamic Time Warping (DTW),, MSE and L2. Then, if an MTS in the test set has one of the three reconstruction errors above the threshold, it is classified as an anomaly. Those functions were chosen since our first experiments demonstrated superior performance than using a single function to measure the reconstruction error.

5.1 Datasets and Preprocessing

In this section, we introduce the datasets utilized and provide an overview of the data processing phase. To ensure a meaningful comparison, we processed each dataset to obtain MTSs with 100 samples per channel overlapping by 20 samples with the previous TS. All the values were normalized between -1 and $+1$, and the different kinds of anomalies were united under a single label.

KDD-99 Dataset The "Knowledge Discovery and Data Mining 1999 (KDD-99)" KDD Cup [25] is a benchmark dataset for evaluating intrusion detection systems. It comprises network traffic data with 42 features describing connections in a computer network. In this dataset, anomalies are represented as various kinds of attacks on the network, so detecting anomalies corresponds to identifying bad connections. Random bad connections were dropped until they constituted 10% of the test set. We selected this dataset for our experiments to ensure both reproducibility and comparability of the results.

Bonfiglioli Dataset Initially, this dataset was split into two parts: training and test set, the latter containing 10% of the total examples. We generated four additional test sets from the original test set by including modified MTS, altered by adding noise and synthetic anomalies. Specifically, we altered the MTS adding two kinds of anomalies. The first one is described as a step function, and the second one is a series of periodic spikes sampled from a 3-component Gaussian Mixture Model. In collaboration with domain experts, we made this choice, assuming anomalies in industrial machinery typically appear as step and periodic disturbances within the MTS data.

Moreover, to further test and validate the AEs' robustness to noise and precision, we introduced random noise, including single-point and salt-and-pepper noise into 10% of the test set, which represents the 10% of the original dataset. This type of noise, representing sporadic, sharp disturbances, does not align with the expected anomaly in industrial machinery. The goal was to assess whether the AEs could effectively distinguish and ignore irrelevant

noise, concentrating instead on significant disturbances relevant to machinery operation.

After injecting these anomalies and noise, we created a total of four different test set versions. In the first test set (A-6 F) we injected **A**nomalies in all **6 F**eatures, while the second one contains synthetic **N**oise too (AN-6 F). The third (A-4 F) and fourth (AN-4 F) ones are the same as the first and second, containing anomalies or anomalies+noise, but these were injected only in **4** features. This approach was taken because two of the features were almost flatlined, with minimal fluctuations, and we wanted to assess whether excluding these features would impact the AD tool's sensitivity to them.

5.2 Hyperparameter Optimization

Given the extensive number of hyperparameters and their significant impact on the AEs performance Berahmand et al. [9], we conducted a systematic search for the optimal hyperparameters using Optuna,[4] a comprehensive hyperparameters optimization tool.

Since hyperparameters are inherently domain-specific, their optimal values are intimately tied to the statistical and structural characteristics of the data, the nature and frequency of anomalies, and the ultimate objectives of the detection or prediction task. This dependency arises because each domain presents a unique set of challenges and data patterns that influence model behavior in distinct ways. Domains also vary in terms of noise levels, feature redundancy, temporal dependencies, and degrees of sparsity or missingness, all of which affect the sensitivity of models to hyperparameter settings. Furthermore, the choice of objective-whether it is maximising precision, recall, F1-score, or minimising detection latency-also influences the optimal hyperparameter configuration.

Given these domain-specific variations, it becomes essential to adopt a systematic and contextual approach to hyperparameter tuning. Therefore, we searched for the optimal configuration, performing 200 Optuna trials, for each model-dataset pair. Each Optuna trial comprises a training process of 100–2000 epochs with batches of 16–512 MTS, where the AEs are trained minimising the MTS reconstruction error using the L2 measure as a loss function.

At the end of the optimisation cycle, the best model is the one with the lowest DTW reconstruction error and is used for the subsequent performance evaluation. DTW measures the similarity between two temporal sequences that may vary in speed or timing. DTW is widely used in tasks involving TS data where sequences may not be perfectly aligned Senin [37].

[4] https://optuna.org/.

Table 1 Optimized hyperparameters for the baseline AE on each dataset

Baseline AE				
Dataset	Embedding dimension	Depth	Kernels	Channels
KDD-99	43	2	[3, 5]	[41, 34]
Bonfiglioli	18	1	[4]	[6]

Table 2 Optimized hyperparameters for the Ad-hoc Time2 Vec-based AE on each dataset

Ad-hoc Time2 Vec based AE				
Dataset	Embedding dimension	Depth	Kernels	Channels
KDD-99	74	1	[8]	[41]
Bonfiglioli	13	2	[7, 3]	[6, 35]

The searched parameters include the embedding size for each parameter (K), the number of convolutional layers (network depth), and the respective kernel size and number of channels. Regarding the K parameter, shown in Eq. 1 but also used with the same meaning on the DWT-based AE, it is worth noting how the embedding has an actual length of K*100, where 100 is the number of time instances since K represents the embedding size for each time instance. To avoid overfitting, we empirically set an upper bound for K so that K can be as big as the number of examples in the training set divided by 100. The regularization rate was fixed at 0.005. Lastly, the depth parameter varies between 1 and 5, referring to the number of repetitions of the fundamental element, which could be a convolution on its own or paired with a batch normalization layer depending on the architecture of the encoder and the decoder, as they are symmetric in most proposed models.

The kernel dimension is an integer between 3 and 9 for each encoder layer, with the decoder using the same kernels in reverse order. The number of filters is an integer between 6 and 150, with the first layer containing as many filters as the MTS channels to ensure proper reconstruction.

The ad-hoc T2 V AE is a special case, having an asymmetrical architecture where the depth parameter defines the number of layers only for the decoder.

5.3 Training of the Ad-hoc Autoencoders

We observed that the training loss for every architecture appears to converge quite smoothly to an error close to 0, suggesting that the model-related hyperparameters were well-optimized and allowed the models to correctly fit the training dataset.

Table 3 Optimized hyperparameters for the Ad-hoc DWT-based AE with coefficients arranged as scalograms on each dataset

Ad-hoc DWT-based scalograms AE				
Dataset	Embedding dimension	Depth	Kernels	Channels
KDD-99	68	1	[3]	[41]
Bonfiglioli	7	1	[5]	[6]

Table 4 Optimized hyperparameters for the Ad-hoc DWT-based AE with coefficients arranged as concatenated vectors on each dataset

Ad-hoc DWT-based vectors AE				
Dataset	Embedding Dimension	Depth	Kernels	Channels
KDD-99	57	1	[5]	[41]
Bonfiglioli	9	1	[4]	[6]

The optimal hyperparameters identified through Optuna are presented in Table 1 for the baseline AE, in Table 2 for the T2 V-based AE. Lastly, Table 3 shows the hyperparameters for DWT coefficients represented as scalograms, while Table 4 displays those where DWT coefficients are treated as vectors.

To establish a baseline for comparing the performance of our custom-designed AE models, we applied the same optimization, training, and evaluation processes to a baseline AE architecture. This AE has a mirrored structure between the encoder and the decoder. Both comprise a series of Conv1D layers, a flattening, and a dense layer.

Our analysis focused on two main aspects: firstly, we wanted to directly compare the performance of the reconstruction error-based AD method against the embedding-based approach. Secondly, we aimed to compare the quality of the embeddings produced by our convolutional AE model against those generated by other AE architectures. The baseline convolutional AE architecture was chosen for its prevalence and demonstrated performance in existing works, ensuring a fair comparison against our proposed T2 V- and DWT-based architectures Miao et al. [31]. This approach aligns with common practices in AD literature, where convolutional autoencoders serve as a standard benchmark for evaluating novel methods.

5.4 Results

Results are presented in Tables 5, 6, 7 and 8. To measure the AD performance of our solutions, we relied on the following performance metrics: Precision, Recall, F1 and F0.5-Score. By simultaneously evaluating Precision and Recall, it is possible to assess a model's ability

Table 5 Comparison of anomaly detection performance between the baseline autoencoder jointly with different ML algorithms (DIF, OCSVM, Deep SVDD, IF, LOF) and a reconstruction error-based method

Baseline Autoencoder								
	Reconstruction error				Deep isolation forest			
Dataset	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.70	0.59	0.54	1.00	0.54	0.45	0.41	0.79
AN-6 F	0.40	0.29	0.25	1.00	0.47	0.35	0.31	0.74
A-4 F	0.26	0.41	0.67	0.16	0.73	0.63	0.58	1.00
AN-4 F	0.12	0.10	0.09	0.16	0.62	0.50	0.45	1.00
KDD99	0.09	0.20	0.90	0.05	0.08	0.17	0.79	0.04
	One-class support vector machine				Deep support vector data description			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.45	0.34	0.29	1.00	0.55	0.43	0.38	1.00
AN-6 F	0.36	0.26	0.22	1.00	0.47	0.35	0.30	1.00
A-4 F	0.54	0.42	0.37	1.00	0.68	0.57	0.51	1.00
AN-4 F	0.41	0.30	0.26	1.00	0.56	0.45	0.39	1.00
KDD99	0.92	0.89	0.87	0.98	0.08	0.14	0.28	0.05
	Isolation forest				Local outlier factor			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.48	0.37	0.32	1.00	0.43	0.41	0.40	0.47
AN-6 F	0.39	0.28	0.24	1.00	0.39	0.35	0.33	0.47
A-4 F	0.59	0.47	0.42	1.00	0.66	0.71	0.76	0.58
AN-4 F	0.46	0.34	0.29	1.00	0.63	0.66	0.69	0.58
KDD99	0.93	0.90	0.88	0.98	0.00	0.00	0.00	0.00

Table 6 Comparison of anomaly detection performance between the Time2 Vec-based autoencoder jointly with many ML algorithms (DIF, OCSVM, Deep SVDD, IF, LOF) and a reconstruction error-based method

T2 V-based Autoencoder								
	Reconstruction error				Deep isolation forest			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	1.00	1.00	1.00	1.00	0.78	0.69	0.64	0.99
AN-6 F	0.63	0.52	0.46	1.00	0.71	0.62	0.57	0.96
A-4 F	1.00	1.00	1.00	1.00	0.84	0.77	0.72	1.00
AN-4 F	0.78	0.69	0.64	1.00	0.77	0.69	0.64	1.00
KDD99	0.98	0.99	0.96	0.99	0.78	0.69	0.65	0.98
	One-class support vector machine				Deep support vector data description			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.88	0.87	0.86	0.90	1.00	1.00	1.00	1.00
AN-6 F	0.74	0.67	0.64	0.90	0.96	0.94	0.93	1.00
A-4 F	0.93	0.89	0.87	1.00	0.92	0.88	0.85	1.00
AN-4 F	0.84	0.77	0.72	1.00	0.73	0.63	0.57	1.00
KDD99	0.94	0.96	0.99	0.91	0.01	0.02	0.5	0.00
	Isolation forest				Local outlier factor			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.66	0.60	0.57	0.78	0.84	0.97	0.89	0.81
AN-6 F	0.58	0.51	0.47	0.76	0.71	0.62	0.57	0.96
A-4 F	0.90	0.85	0.81	1.00	0.84	0.77	0.72	1.00
AN-4 F	0.64	0.76	0.87	0.50	0.77	0.69	0.64	1.00
KDD99	0.98	0.99	1.00	0.97	0.78	0.69	0.65	0.98

Table 7 Comparison of anomaly detection performance between the Discrete Wavelet Transform-based autoencoder with coefficients arranged as vectors jointly with many ML algorithms (DIF, OCSVM, Deep SVDD, IF, LOF) and a reconstruction error-based method

DWT-based autoencoder with coefficients as vectors								
	Reconstruction error				Deep isolation forest			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.71	0.61	0.55	1.00	0.44	0.46	0.48	0.40
AN-6 F	0.60	0.48	0.43	1.00	0.41	0.44	0.47	0.37
A-4 F	0.93	0.96	0.97	0.89	0.38	0.61	1.00	0.24
AN-4 F	0.84	0.81	0.79	0.89	0.38	0.58	0.90	0.24
KDD99	0.09	0.19	0.91	0.05	0.01	0.02	1.00	0.00
	One-class support vector machine				Deep support vector data description			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.36	0.26	0.22	1.00	0.63	0.52	0.46	1.00
AN-6 F	0.30	0.21	0.18	1.00	0.56	0.44	0.39	1.00
A-4 F	0.36	0.26	0.23	0.84	0.67	0.64	0.63	0.71
AN-4 F	0.30	0.21	0.18	0.84	0.61	0.57	0.54	0.71
KDD99	0.93	0.91	0.90	0.97	0.09	0.17	0.35	0.05
	Isolation forest				Local outlier factor			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.50	0.38	0.33	1.00	0.71	0.61	0.55	1.00
AN-6 F	0.42	0.31	0.26	1.00	0.59	0.48	0.42	1.00
A-4 F	0.61	0.50	0.44	1.00	0.73	0.83	0.92	0.61
AN-4 F	0.52	0.40	0.35	1.00	0.62	0.63	0.64	0.61
KDD99	0.94	0.92	0.90	0.97	0.01	0.02	0.25	0.00

to discriminate anomalous patterns from normal behaviour while minimizing, in particular, the number of false positives. These represent normal behaviour classified as anomalous and as greatly described in Sect. 3 they can

Table 8 Comparison of anomaly detection performance between the Discrete Wavelet Transform-based autoencoder with coefficients arranged as scalograms jointly with many ML algorithms (DIF, OCSVM, Deep SVDD, IF, LOF) and a reconstruction error-based method

DWT-based autoencoder with coefficients as scalograms								
	Reconstruction error				Deep isolation forest			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.70	0.60	0.54	1.00	0.58	0.47	0.42	0.95
AN-6 F	0.57	0.45	0.40	1.00	0.57	0.46	0.41	0.94
A-4 F	0.97	0.96	0.95	1.00	0.66	0.57	0.52	0.89
AN-4 F	0.79	0.70	0.66	1.00	0.66	0.57	0.52	0.89
KDD99	0.11	0.24	0.94	0.06	0.97	0.98	0.98	0.96
	One-class support vector machine				Deep support vector data description			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.37	0.27	0.23	0.89	0.73	0.70	0.69	0.76
AN-6 F	0.35	0.26	0.22	0.89	0.72	0.70	0.69	0.76
A-4 F	0.38	0.28	0.24	0.89	0.91	0.91	0.92	0.89
AN-4 F	0.37	0.28	0.23	0.89	0.89	0.89	0.89	0.89
KDD99	0.93	0.91	0.89	0.98	0.96	0.97	0.97	0.95
	Isolation forest				Local outlier factor			
Test set	F1-Score	F0.5-Score	Precision	Recall	F1-Score	F0.5-Score	Precision	Recall
A-6 F	0.56	0.45	0.40	0.94	0.72	0.66	0.63	0.84
AN-6 F	0.55	0.44	0.39	0.94	0.72	0.66	0.63	0.84
A-4 F	0.61	0.50	0.44	1.00	0.73	0.83	0.92	0.61
AN-4 F	0.60	0.50	0.44	0.94	0.71	0.79	0.85	0.61
KDD99	0.93	0.90	0.89	0.97	0.02	0.05	0.55	0.01

undermine manufacturing productivity. F1-Score is the harmonic mean of Precision and Recall, providing a balanced evaluation of a model's performance. F0.5-Score places more emphasis on Precision than Recall, which can be useful for AD tasks where false positives are more costly than false negatives, like the Bonfiglioli use case. Therefore, the goal was to obtain an AD model that could strike the right balance between accurately detecting true anomalies and keeping the rate of false positives low.

The aforementioned metrics have been measured on the KDD-99 and the Bonfiglioli dataset. Regarding the latter, as further detailed in Sect. 5.1 we experimented with four variations of the test set, with anomalies (A) and noise (N) injected, affecting either six (6 F) or four (4 F) features.

Moreover, other than comparing our two ad-hoc solutions to a standard convolutional AE, we evaluated the performance of each approach in two ways: first, using the AE reconstruction error, and second, leveraging the AE to generate embeddings, which were then used for AD with various ML algorithms.

Table 5 shows the performance using the baseline AE described in Sect. 5.3. Results show how this AE performs better in noiseless conditions, as in the A-6 F test, and when AD is performed measuring the reconstruction error. On the other hand, AD performed on the embeddings paired with one-class classification algorithms on the Bonfiglioli datasets performed worse than doing the same with the other developed AE-based solutions. This highlights the low quality of the computed embeddings, which can not preserve the MTS relation. Conversely, on the KDD-99 dataset, results are the opposite, with acceptable performance using only IF and One-Class Support Vector Machine (OCSVM).

Finally, it is worth highlighting that the LOF algorithm applied to the embeddings generated by the baseline convolutional AE fails to identify any anomalies.

Results obtained using the T2 V-based AE are presented in Table 6. This AE generally shows a good F1-score on the noisy Bonfiglioli dataset (AN-6 F and AN-4 F). Results are also very good on the benchmark dataset in most of the cases. Despite the good performance measured using the reconstruction-based method, the performance is better across the test sets with noise injected when performing AD using the ML algorithms on the embeddings. Combining the T2 V-based AE with all the tested algorithms-except for IF-consistently demonstrated high and stable performance, both precision and recall, across all datasets. Furthermore, both the reconstruction error and the ML algorithms, with some variations, can effectively distinguish anomalies from normal MTS data in the KDD-99 dataset (except in one case).

Regarding the DWT-based model, with coefficients as vectors, Table 7 shows that the best results are obtained on the 4 F version of the Bonfiglioli dataset by the reconstruction error-based method and by LOF. On the other hand, on

Table 9 Mean and standard deviation of single-example embedding generation across different model architectures and datasets and anomaly detection algorithms inference times (s)

Baseline autoencoder			
Dataset	Embedding computation	IF	One-Class SVM
Bonfiglioli	0.2400 ± 0.1190	0.2000 ± 0.0720	0.0022 ± 0.0040
KDD	0.1400 ± 0.0290	0.4200 ± 0.0680	0.0012 ± 0.0004
Dataset	LOF	Deep IF	Deep SVDD
Bonfiglioli	0.0240 ± 0.0078	1.1550 ± 0.4100	0.0006 ± 0.0003
KDD	0.0250 ± 0.0002	1.0400 ± 0.2200	0.0007 ± 0.0002
T2 V-based autoencoder			
Dataset	Embedding computation	IF	One-Class SVM
Bonfiglioli	0.1440 ± 0.0360	0.1120 ± 0.0551	0.0011 ± 0.0004
KDD	0.1400 ± 0.0340	0.3486 ± 0.0750	0.0016 ± 0.0005
Dataset	LOF	Deep IF	Deep SVDD
Bonfiglioli	0.0244 ± 0.0086	1.1145 ± 0.3010	0.0012 ± 0.0003
KDD	0.0370 ± 0.0073	1.0166 ± 0.1778	0.0011 ± 0.0058
DWT-based Autoencoder with coefficients as vectors			
Dataset	Embedding computation	IF	One-Class SVM
Bonfiglioli	0.2340 ± 0.0100	0.5090 ± 0.1990	0.0042 ± 0.0034
KDD	0.0244 ± 0.1258	0.3813 ± 0.2000	0.0012 ± 0.0006
Dataset	LOF	Deep IF	Deep SVDD
Bonfiglioli	0.0092 ± 0.0023	4.1626 ± 0.6000	0.0011 ± 0.0003
KDD	0.0500 ± 0.0238	1.0569 ± 0.2230	0.0010 ± 0.0005
DWT-based Autoencoder with coefficients as scalograms			
Dataset	Embedding computation	IF	One-Class SVM
Bonfiglioli	0.1633 ± 0.0840	0.5368 ± 0.1778	0.0010 ± 0.0005
KDD	0.3000 ± 0.1718	0.2300 ± 0.1110	0.0014 ± 0.0008
Dataset	LOF	Deep IF	Deep SVDD
Bonfiglioli	0.0059 ± 0.0007	4.3050 ± 0.9300	0.0009 ± 0.0004
KDD	0.0250 ± 0.0050	1.0730 ± 0.2100	0.0019 ± 0.0013

the KDD-99, dataset performance is very good only when AD is executed on the embedding together with IF and One-Class SVM.

Lastly, the results of the DWT-based model, where the coefficients are represented as scalograms, are presented in Table 8. This model performed better than the other version of the DWT-based AE and the baseline, with a particularly high F1-score when used together with Deep SVDD on every dataset, and with Deep IF on KDD99. However, performance is generally worse in this case than the T2 V-based AE.

With this embedding model, the best performance on the noisy test sets (AN-6 F and AN-4 F) is reached using an ML algorithm on the embeddings, specifically Deep SVDD and LOF. The highest precision is reached in the AN-4 F test set, with the baseline, the reconstruction-based method and the T2 V-based AE. On the other hand, in the KDD-99 dataset, the best results are obtained using IF, Deep SVDD, One-Class SVM and D-IF.

Table 10 Mean and standard deviation of a single training for different model architectures and datasets times (s)

Baseline autoencoder	
Dataset	Training time
Bonfiglioli	91.384 ± 52.333
KDD	152.712 ± 138.135
T2 V-based Autoencoder	
Dataset	Training time
Bonfiglioli	94.579 ± 51.781
KDD	314.389 ± 172.661
DWT-based Autoencoder with coefficients as vectors	
Dataset	Training time
Bonfiglioli	60.561 ± 41.504
KDD	182.720 ± 319.971
DWT-based Autoencoder with coefficients as scalograms	
Dataset	Training Time
Bonfiglioli	245.813 ± 371.495
KDD	2237.752 ± 1508.083

5.5 Scalability Analysis

In this section, we show the scalability of all the proposed solutions in terms of computational time to assess applicability in a real industrial environment, where strict time constraints are usually present. We measured the inference and training time for each embedding model and one-class classification algorithm on both datasets, KDD-99 and Bonfiglioli.

To ensure meaningful results, we simulated an industrial environment with a cloud-edge architecture, where generally, the models are trained in the cloud and then deployed on the edge. We simulated the cloud environment using an HPC cluster node where GPU resources are partitioned, allowing multiple users to share the same hardware. For our tests, we ran the experiments on 2 AMD® EPYC 9124 16-core CPUs, 64Gb of RAM, and 25% of a single Nvidia H100 GPU equipped with 96GB of VRAM. The edge node was simulated with a machine equipped with an AMD® Ryzen 9 pro 7940 hs CPU and 32GB of RAM. We conducted all the experiments with Python 3.9.16, CUDA 12.2, Tensorflow 2.15, and scikit-learn 1.6.1.

The average and standard deviation of the inference times calculated on single examples (with over 100 inferences) are shown in Table 9. Inference times demonstrate how all the developed embedding models and most classification algorithms can meet strict near-real-time constraints, typical of industrial environments, with an average embedding

computation time of less than 0.24 s. Most one-class classification algorithms perform inference in less than 0.3 s on all embeddings, with IF and D-IF being the only exceptions. The latter, when applied to the DWT-based embedding, takes more than 4 s to compute the prediction. On the other hand, One-Class SVM and Deep SVDD show the best performance, with inference times always ≤ 0.043 s. On the other hand, while our approach can meet near-real-time constraints, it may not be suitable for use cases with stricter time requirements. In such cases, where AD must be performed in real-time and inference is executed at high frequency, other techniques may be more appropriate. It is also possible to combine multiple AD tools to improve performance as well as detection accuracy and precision, which are fundamental in smart manufacturing environments.

Furthermore, Table 10 presents the average training time and standard deviation for each model-dataset pair. In our experiments, the T2 V-based AE showed the lowest inference times in the Bonfiglioli dataset, while the baseline was the fastest in the KDD-99 dataset. On the other hand, the scalogram version of the DWT-based AE was the slowest. Notably, all the developed models showed that their performance enables fast model retraining. This is particularly useful to meet the precision requirements of industrial scenarios, which usually present data/concept drift phenomena requiring frequent model re-deployment and re-training.

Lastly, although the present analysis confirms the scalability of the model under the tested conditions, additional investigation may be required to evaluate its performance in more demanding, large-scale scenarios comprehensively. Nevertheless, the low training and inference times observed in our experiments indicate that the proposed solution remains a feasible option even for large-scale industrial datasets.

5.6 Discussion

Regarding the Bonfiglioli dataset, our experimental evaluation showed how in a relatively simple and not noisy environment many solutions proved to be effective. In A-6 F and A-4 F, where anomalies have been injected in 6 or 4 features without synthetic noise, the T2 V-based and the scalogram version of the DWT-based AEs demonstrated good performance when their reconstruction error is used and when they are used to compute MTS embeddings. However, embeddings computed with the T2 V-based AEs proved to be high quality since good performance has been reached with all the ML models.

Moreover, all our developed solutions showed good performance in more realistic and noisy environments, where it is harder to spot anomalies (AN-6 F and AN-4 F cases). In detail, the best performance on the AN-6 F data set was reached using the T2 V-based AE together with the Deep SVDD used to discriminate anomalies, while on the AN-4 F test set was reached using the scalogram version of the DWT-based autoencoder together with Deep SVDD. In these datasets, it is important to note how the embedding-based solution significantly outperformed the reconstruction-based one and our ad-hoc AEs outperformed the baseline. Therefore, we can conclude that in this real industrial scenario the embedding-based solution, with some difference between the three ad-hoc models, is more suitable. Our experiments also suggest that, in terms of the F1-score, the best results are obtained with the T2 V-based AE. However, we can observe how these differences could be domain-specific therefore it is hard to find a single solution for every use case.

The experiment with the KDD-99 dataset validates the effectiveness of our embedding-based approach using a publicly available dataset. All the developed embedding models consistently achieved optimal performance across a variety of ML algorithms. Even here, our solution outperformed the baseline AE, demonstrating once again our approach's effectiveness and our solutions' ability to create high-quality embeddings. However, in this specific experiment, we did not notice a substantial difference between the reconstruction-based and the embedding-based methods.

Finally, we can observe how T2 V-based AE demonstrated a great ability in embedding MTSs since all the ML algorithms achieved high performance. Instead, DWT-based AEs reach good performance only when combined with some of the ML algorithms.

These results demonstrate that performing anomaly detection on embeddings, generated by specifically designed models, can effectively replicate-and often significantly surpass-the anomaly detection performance of reconstruction-based methods. At the same time, we demonstrated how our solution can effectively embed MTS, mostly preserving the relations. This underscores our solution's advantage in handling various AD scenarios, making it a versatile choice for diverse and unpredictable environments, like the Bonfiglioli one.

Let us conclude with some considerations toward the practical adoption of our solution. Deploying and managing ML models is fundamental to ensure the reliability, scalability, and efficiency of AI-driven solutions. More specifically, unlike traditional software, ML models require continuous validation, adaptation to context changes, and (if necessary) automatic retraining and redeployment Kreuzberger et al. [27]. In fact, in industrial environments, plant reconfigurations following evolving production demands, changes in sensorization, and equipment wear could result in changes in the manufacturing process that the ML model is applied to, leading to data and concept drift phenomena Hegedűs and Varga, [22].

To meet these requirements, we have explored the application of MLOps techniques in real industrial environments

in several works. In Colombi et al. [14], we designed and implemented, using open source technologies, an MLOps platform to manage the entire ML lifecycle in a complex multi-cluster edge-to-cloud compute continuum scenario, enabling seamless deployment across different serving runtimes, continuous monitoring, and automated retraining Colombi et al. [11]. Moreover, we designed an automated, self-optimizing MLOps pipeline that seamlessly integrates these generative models into the end-to-end ML lifecycle, ensuring more robust, reliable, and accurate industrial applications. In addition, in Dahdal et al. [18], we tackled data imbalance challenges in smart manufacturing by leveraging Deep Generative Models, specifically Generative Adversarial Networks (GANs), to generate synthetic data and enhance ML model training.

The integration of our framework into an MLOps platform, such as the one discussed above, would allow the automatic detection of both data and concept drift, triggering model retraining and redeployment as needed, and ultimately ensuring that the detection accuracy remains robust as normal and anomalous patterns evolve over time.

6 Conclusions, Limitations and Future work

In this paper, we presented an AD framework that leverages three different ad-hoc AEs to extract vector representation from MTS. Specifically, these include T2 V-based and two DWT-based models, where the DWT coefficients are represented as a vector or a scalogram.

This brings several advantages, such as the flexibility of using many ML techniques that provide consistent and high-performance metrics across varying conditions. We demonstrated how our solution, compared with a reconstruction-based method, results in higher performance, stability, and robustness to noise in a freely accessible dataset and a real industrial use case.

However, our work is primarily applicative rather than theoretical in nature, aiming to demonstrate how embedding-based approaches can enhance anomaly detection in real-world Industry 5.0 applications. While our focus is on practical implementation and performance improvements, we recognize the value of further analyzing the learned feature space and embedding characteristics. Applicative studies like ours can be complemented by theoretical investigations, particularly in areas such as post-hoc interpretability Cascione et al. [15], to provide deeper insights into the underlying mechanisms of these models.

Specifically, future research on classification algorithms should prioritize ML interpretability and explainability, which are essential to Industry 5.0 scenarios. Recent advances in explainable AI (XAI) techniques, such as SHAP values and attention mechanisms, have shown promise in fault diagnosis and industrial applications, and their integration could further enhance the transparency and trustworthiness of our framework Brusa et al. [7]. These efforts aim to address a significant limitation of current embedding-based anomaly detection methods by enhancing both robustness Anisetti et al. [1] and trustworthiness Anisetti et al. [2]. To address this issue, ML algorithms that preserve interpretability when applied to embeddings could hold significant potential.

Finally, we plan to apply our embedding solution in other industrial use cases and to other types of data, such as visual and tabular ones. We also expect to use our embedding models to go beyond AD applications, for example, to accurately estimate industrial machinery's Remaining Useful Lifetime (RUL). In addition, we intend to study other tools and methodologies to compute embeddings of complex data types. For instance, we are considering adopting self-supervised techniques, specifically contrastive learning, that are gaining momentum across multiple domains but remain relatively unexplored in industrial settings.

Author Contributions This section has been anonymized for double-blind peer review.

Funding This section has been anonymized for double-blind peer review.

Data Availability Kdd99 is publicly available at https://kdd.ics.uci.edu/databases/kddcup99. The Bonfiglioli dataset is proprietary and cannot be shared. Documented code that can be used to reproduce or build upon the reported experiments is available online: https://github.com/DSG-UniFE/Embedding_Models_for_MTS_Anomaly_Detection_in_I5.0.

Declarations

Conflict of interest Not applicable

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Anisetti M, Ardagna CA, Balestrucci A, Bena N, Damiani E, Yeun CY (2023) On the robustness of random forest against untargeted data poisoning: an ensemble-based approach. IEEE Trans Sustain Comput 8(4):540–554. https://doi.org/10.1109/TSUSC.2023.3293269

2. Anisetti M, Ardagna CA, Bena N, Damiani E (2023) Rethinking certification for trustworthy machine-learning-based applications. IEEE Internet Comput 27(6):22–28. https://doi.org/10.1109/MIC.2023.3322327
3. Alom Z, Carminati B, Ferrari E (2020) A deep learning model for twitter spam detection. Online Soc Netw Media 18:100079. https://doi.org/10.1016/j.osnem.2020.100079
4. Aremu OO, Hyland-Wood D, McAree PR (2020) A machine learning approach to circumventing the curse of dimensionality in discontinuous time series machine data. Reliabil Eng Syst Saf 195:106706. https://doi.org/10.1016/j.ress.2019.106706
5. Audibert J, Michiardi P, Guyard F, Marti S, Zuluaga MA (2020) Usad: unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining. KDD '20, pp. 3395–3404. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3394486.3403392
6. Bonifati A, Buono FD, Guerra F, Tiano D (2022) Time2feat: learning interpretable representations for multivariate time series clustering. Proc VLDB Endow 16(2):193–201. https://doi.org/10.14778/3565816.3565822
7. Brusa E, Cibrario L, Delprete C, Di Maggio LG (2023) Explainable ai for machine fault diagnosis: understanding features' contribution in machine learning models for industrial condition monitoring. Appl Sci. https://doi.org/10.3390/app13042038
8. Bevilacqua M, Ciarapica FE, Diamantini C, Potena D (2017) Big data analytics methodologies applied at energy management in industrial sector: a case study. Int J RF Technol 8(3):105–122
9. Berahmand K, Daneshfar F, Salehi ES, Li Y, Xu Y (2024) Autoencoders and their applications in machine learning: a survey. Artif Intell Rev 57(2):28
10. Bertrand JH, Gargano JP, Mombaerts L, Taws J (2024) Autoencoder-based general purpose representation learning for customer embedding
11. Colombi L, Boleac I, Brina M, Dahdal S, Tortonesi M, Vignoli M, Stefanelli C (2025) Multi-cluster mlops platform for industry 5.0. In: 2025 IEEE symposium on computers and communications (ISCC)
12. Colombi L, Caro ED, Dahdal S, Poltronieri F, Tabanelli F, Tortonesi M, Stefanelli C, Vignoli M (2025) Fededge-learn: a semi-supervised federated learning framework for industry 5.0. In: 2025 IEEE symposium on computers and communications (ISCC)
13. Colombi L, Dahdal S, Di Caro E, Fronteddu R, Gilli A, Morelli A, Poltronieri F, Tortonesi M, Suri N, Stefanelli C (2024) Efficient data dissemination via semantic filtering at the tactical edge. In: MILCOM 2024 - 2024 IEEE military communications conference (MILCOM), pp 457–462 . https://doi.org/10.1109/MILCOM61039.2024.10773700
14. Colombi L, Gilli A, Dahdal S, Boleac I, Tortonesi M, Stefanelli C, Vignoli M (2024) A machine learning operations platform for streamlined model serving in industry 5.0. In: NOMS 2024-2024 IEEE Network Operations and Management Symposium, pp. 1–6 . https://doi.org/10.1109/NOMS59830.2024.10575103
15. Cascione A, Setzu M, Guidotti R (2024) Data-agnostic pivotal instances selection for decision-making models. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 367–386
16. Colombi L, Vespa M, Belletti N, Brina M, Dahdal S, Tabanelli F, Bellodi E, Tortonesi M, Stefanelli C, Vignoli M (2025) Multivariate time series anomaly detection in industry 5.0. arXiv:2503.15946
17. Chevrot A, Vernotte A, Legeard B (2022) Cae: contextual autoencoder for multivariate time-series anomaly detection in air transportation. Comput Secur 116:102652. https://doi.org/10.1016/j.cose.2022.102652
18. Dahdal S, Colombi L, Brina M, Gilli A, Tortonesi M, Vignoli M, Stefanelli C (2024) An mlops framework for gan-based fault detection in Bonfiglioli's evo plant. Infocommun J 16(2):2–10. https://doi.org/10.36244/ICJ.2024.2.1
19. Echihabi K, Zoumpatianos K, Palpanas T (2020) Scalable machine learning on high-dimensional vectors: From data series to deep network embeddings. In: Proceedings of the 10th international conference on web intelligence, mining and semantics. WIMS 2020. Association for Computing Machinery, New York, NY, USA, pp 1–6. https://doi.org/10.1145/3405962.3405989
20. Forkan ARM, Kang YB, Marti F et al (2024) Aiot-citysense: Ai and iot-driven city-scale sensing for roadside infrastructure maintenance. Data Sci Eng 9:26–40. https://doi.org/10.1007/s41019-023-00236-5
21. Hu Y, Chen C, Deng B et al (2024) Decoupling anomaly discrimination and representation learning: self-supervised learning for anomaly detection on attributed graph. Data Sci Eng 9:264–277. https://doi.org/10.1007/s41019-024-00249-8
22. Hegedűs C, Varga P (2023) Tailoring MLOps techniques for industry 5.0 needs. In: 2023 19th international conference on network and service management (CNSM), pp 1–7 . https://doi.org/10.23919/CNSM59352.2023.10327814
23. Ienco D, Interdonato R (2020) Deep multivariate time series embedding clustering via attentive-gated autoencoder. In: Lauw HW, Wong RC-W, Ntoulas A, Lim E-P, Ng S-K, Pan SJ (eds) Advances in knowledge discovery and data mining. Springer, Cham, pp 318–329
24. Kanarachos S, Christopoulos S-RG, Chroneos A, Fitzpatrick ME (2017) Detecting anomalies in time series data via a deep learning algorithm combining wavelets, neural networks and hilbert transform. Expert Syst Appl 85:292–304. https://doi.org/10.1016/j.eswa.2017.04.028
25. KDD Cup 1999: KDD Cup 1999 Data. https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
26. Kazemi SM, Goel R, Eghbali S, Ramanan J, Sahota J, Thakur S, Wu S, Smyth C, Poupart P, Brubaker MA (2019) Time2vec: learning a vector representation of time. CoRR arXiv:1907.05321
27. Kreuzberger D, Kühl N, Hirschl S (2023) Machine learning operations (MLOps): overview, definition, and architecture. IEEE Access 11:31866–31879. https://doi.org/10.1109/ACCESS.2023.3262138
28. Khan SS, Madden MG (2014) One-class classification: taxonomy of study and review of techniques. Knowl Eng Rev 29(3):345–374. https://doi.org/10.1017/S026988891300043X
29. Mallat SG (1989) A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans Pattern Anal Mach Intell 11(7):674–693. https://doi.org/10.1109/34.192463
30. Meyer Y (1993) Wavelets and operators. Cambridge studies in advanced mathematics. Cambridge University Press, Cambridge
31. Miao J, Tao H, Xie H, Sun J, Cao J (2024) Reconstruction-based anomaly detection for multivariate time series using contrastive generative adversarial networks. Inf Process Manag 61(1):103569. https://doi.org/10.1016/j.ipm.2023.103569
32. Malhotra P, Vig L, Shroff G, Agarwal P (2015) Long short term memory networks for anomaly detection in time series. In: ESANN 2015 Proceedings, European Symposium on Artificial Neural Networks
33. Pacheco J, Benitez VH, Pérez G, Brau A (2024) Wavelet-based computational intelligence for real-time anomaly detection and fault isolation in embedded systems. Machines. https://doi.org/10.3390/machines12090664

34. Peñaloza V (2020) Time2vec embedding on a seq2seq bi-directional lstm network for pedestrian trajectory prediction. Res Comput Sci 149:249–260
35. Rhif M, Ben Abbes A, Farah IR, Martínez B, Sang Y (2019) Wavelet transform application for/in non-stationary time-series analysis: a review. Appl Sci. https://doi.org/10.3390/app9071345
36. Seliya N, Abdollah Zadeh A, Khoshgoftaar TM (2021) A literature review on one-class classification and its potential applications in big data. J Big Data 8(1):122. https://doi.org/10.1186/s40537-021-00514-x
37. Senin P (2008) Dynamic time warping algorithm review, vol 855(1–23). Information and Computer Science Department University of Hawaii, Manoa Honolulu, 40
38. Solorio-Fernández S, Carrasco-Ochoa JA, Martínez-Trinidad JF (2020) A review of unsupervised feature selection methods. Artif Intell Rev 53(2):907–948. https://doi.org/10.1007/s10462-019-09682-y
39. Siegel B (2020) Industrial anomaly detection: a comparison of unsupervised neural network architectures. IEEE Sensors Lett 4(8):1–4. https://doi.org/10.1109/LSENS.2020.3007880
40. Schmidl S, Wenig P, Papenbrock T (2022) Anomaly detection in time series: a comprehensive evaluation. Proc VLDB Endow 15:1779–1797
41. Tank A (2018) Discovering interactions in multivariate time series. https://api.semanticscholar.org/CorpusID:127719466
42. Tartarisco G, Cicceri G, Bruschetta R, Tonacci A, Campisi S, Vitabile S, Cerasa A, Distefano S, Pellegrino A, Modesti PA, Pioggia G (2024) An intelligent medical cyber-physical system to support heart valve disease screening and diagnosis. Expert Syst Appl 238:121772. https://doi.org/10.1016/j.eswa.2023.121772
43. Venanzi R, Dahdal S, Solimando M, Campioni L, Cavalucci A, Govoni M, Tortonesi M, Foschini L, Attana L, Tellarini M, Stefanelli C (2023) Enabling adaptive analytics at the edge with the bi-rex big data platform. Comput Ind 147:103876. https://doi.org/10.1016/j.compind.2023.103876
44. Wu W, He L, Lin W, Su Y, Cui Y, Maple C, Jarvis S (2022) Developing an unsupervised real-time anomaly detection scheme for time series with multi-seasonality. IEEE Trans Knowl Data Eng 34(9):4147–4160. https://doi.org/10.1109/TKDE.2020.3035685
45. Wang J, Shao S, Bai Y, Deng J, Lin Y (2023) Multiscale wavelet graph autoencoder for multivariate time-series anomaly detection. IEEE Trans Instrum Meas 72:1–11. https://doi.org/10.1109/TIM.2022.3223142
46. Wu J, Yao L, Liu B, Ding Z, Zhang L (2020) Combining oc-svms with lstm for detecting anomalies in telemetry data with irregular intervals. IEEE Access 8:106648–106659. https://doi.org/10.1109/ACCESS.2020.3000859
47. Zhang Y, Chen Y, Wang J, Pan Z (2023) Unsupervised deep anomaly detection for multi-sensor time-series signals. IEEE Trans Knowl Data Eng 35(2):2118–2132. https://doi.org/10.1109/TKDE.2021.3102110
48. Zhang L, Zhu Z, Jeffay K, Marron JS, Smith FD (2008) Multi-resolution anomaly detection for the internet. In: IEEE INFOCOM workshops 2008, pp 1–6 . https://doi.org/10.1109/INFOCOM.2008.4544618