# Deep Learning for Time Series Anomaly Detection: A Survey

ZAHRA ZAMANZADEH DARBAN, Faculty of IT, Monash University, Clayton, Australia
GEOFFREY I. WEBB, Faculty of IT, Monash University, Clayton, Australia
SHIRUI PAN, School of Information and Communication Technology, Griffith University, Gold Coast, Australia
CHARU AGGARWAL, IBM T. J. Watson Research Center, Yorktown Heights, United States
MAHSA SALEHI, Faculty of IT, Monash University, Clayton, Australia

Time series anomaly detection is important for a wide range of research fields and applications, including financial markets, economics, earth sciences, manufacturing, and healthcare. The presence of anomalies can indicate novel or unexpected events, such as production faults, system defects, and heart palpitations, and is therefore of particular interest. The large size and complexity of patterns in time series data have led researchers to develop specialised deep learning models for detecting anomalous patterns. This survey provides a structured and comprehensive overview of state-of-the-art deep learning for time series anomaly detection. It provides a taxonomy based on anomaly detection strategies and deep learning models. Aside from describing the basic anomaly detection techniques in each category, their advantages and limitations are also discussed. Furthermore, this study includes examples of deep anomaly detection in time series across various application domains in recent years. Finally, it summarises open issues in research and challenges faced while adopting deep anomaly detection models to time series data.

## 1 Introduction

The detection of anomalies, also known as outlier or novelty detection, has been an active research field in numerous application domains since the 1960s [72]. As computational processes evolve, the collection of big data and its use in **artificial intelligence (AI)** is better enabled, contributing

Authors' Contact Information: Zahra Zamanzadeh Darban, Faculty of IT, Monash University, Clayton, Victoria, Australia; e-mail: zahra.zamanzadeh@monash.edu; Geoffrey I. Webb, Faculty of IT, Monash University, Clayton, Victoria, Australia; e-mail: geoff.webb@monash.edu; Shirui Pan, School of Information and Communication Technology, Griffith University, Gold Coast, Queensland, Australia; e-mail: s.pan@griffith.edu.au; Charu Aggarwal, IBM T. J. Watson Research Center, Yorktown Heights, New York, United States; e-mail: charu@us.ibm.com; Mahsa Salehi, Faculty of IT, Monash University, Clayton, Victoria, Australia; e-mail: mahsa.salehi@monash.edu.

to time series analysis including the detection of anomalies. With greater data availability and increasing algorithmic efficiency/computational power, time series analysis is increasingly used to address business applications through forecasting, classification, and anomaly detection [23, 57]. **Time series anomaly detection (TSAD)** has received increasing attention in recent years, because of increasing applicability in a wide variety of domains, including urban management, intrusion detection, medical risk, and natural disasters.

Deep learning has become increasingly capable over the past few years of learning expressive representations of complex time series, like multidimensional data with both spatial (intermetric) and temporal characteristics. In deep anomaly detection, neural networks are used to learn feature representations or anomaly scores in order to detect anomalies. Many deep anomaly detection models have been developed, providing significantly higher performance than traditional time series anomaly detection tasks in different real-world applications.

Although the field of anomaly detection has been explored in several literature surveys [17, 20, 24, 26, 140] and some evaluation review papers exist [101, 153], there is only one survey on deep anomaly detection methods for time series data [37]. However, the mentioned survey [37] has not covered the vast range of TSAD methods that have emerged in recent years, such as DAEMON [33], TranAD [171], DCT-GAN [114], and Interfusion [117]. Additionally, the representation learning methods within the taxonomy of TSAD methodologies has not been addressed in this survey. As a result, there is a need for a survey that enables researchers to identify important future directions of research in TSAD and the methods that are suitable to various application settings. Specifically, this article makes the following contributions:

— **Taxonomy**: We present a novel taxonomy of deep anomaly detection models for time series data. These models are broadly classified into four categories: forecasting-based, reconstruction-based, representation-based and hybrid methods. Each category is further divided into subcategories based on the deep neural network architectures used. This taxonomy helps to characterise the models by their unique structural features and their contribution to anomaly detection capabilities.

— **Comprehensive Review**: Our study provides a thorough review of the current state of the art in time series anomaly detection up to 2024. This review offers a clear picture of the prevailing directions and emerging trends in the field, making it easier for readers to understand the landscape and advancements.

— **Benchmarks and Datasets**: We compile and describe the primary benchmarks and datasets used in this field. Additionally, we categorise the datasets into a set of domains and provide hyperlinks to these datasets, facilitating easy access for researchers and practitioners.

— **Guidelines for Practitioners**: Our survey includes practical guidelines for readers on selecting appropriate deep learning architectures, datasets, and models. These guidelines are designed to assist researchers and practitioners in making informed choices based on their specific needs and the context of their work.

— **Fundamental Principles**: We discuss the fundamental principles underlying the occurrence of different types of anomalies in time series data. This discussion aids in understanding the nature of anomalies and how they can be effectively detected.

— t**Evaluation Metrics and Interpretability**: We provide an extensive discussion on evaluation metrics together with guidelines for metric selection. Additionally, we include a detailed discussion on model interpretability to help practitioners understand and explain the behaviour and decisions of TSAD models.

This article is organised as follows. In Section 2, we start by introducing preliminary definitions, which is followed by a taxonomy of anomalies in time series. Section 3 discusses the application of
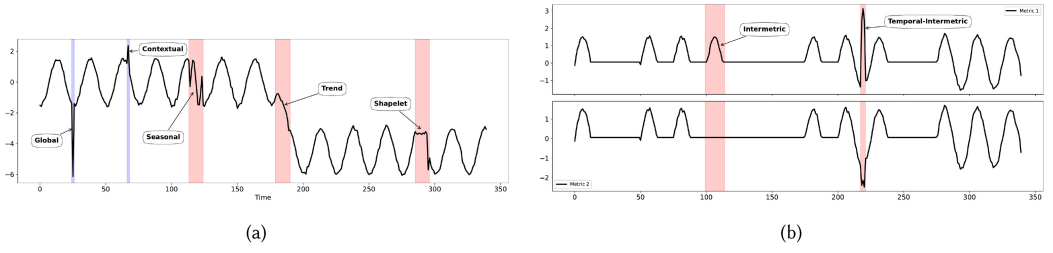
Fig. 1. (a) An overview of different temporal anomalies plotted from the NeurIPS-TS dataset [107]. Global and contextual anomalies occur in a point (coloured in blue). Seasonal, trend and shapelet can occur in a subsequence (coloured in red). (b) Intermetric and temporal-intermetric anomalies in MTS. In this figure, metric 1 (top) is power consumption, and metric 2 (bottom) is CPU usage.

deep anomaly detection models to time series. Different deep models and their capabilities are then presented based on the main approaches (forecasting-based, reconstruction-based, representation-based, and hybrid) and architectures of deep neural networks. Additionally, Section 7 explores the applications of time series deep anomaly detection models in different domains. Finally, Section 8 provides several challenges in this field that can serve as future opportunities. An overview of publicly available and commonly used datasets for the considered anomaly detection models can be found in Section 4.

## 2 Background

A time series is a series of data points indexed sequentially over time. The most common form of time series is a sequence of observations recorded over time [75]. Time series are often divided into *univariate* (one-dimensional) and *multivariate* (multi-dimensional). These two types are defined in the following subsections. Thereafter, decomposable components of the time series are outlined. Following that, we provide a taxonomy of anomaly types based on time series' components and characteristics.

### 2.1 Univariate Time Series

As the name implies, a **univariate time series (UTS)** is a series of data that is based on a single variable that changes over time, as shown in Figure 1(a). Keeping a record of the humidity level every hour of the day would be an example of this. The time series $X$ with $t$ timestamps can be represented as an ordered sequence of data points in the following way:

$$X = (x_1, x_2, \ldots, x_t) \tag{1}$$

Where $x_i$ represents the data at timestamp $i \in T$ and $T = \{1, 2, \ldots, t\}$.

### 2.2 Multivariate Time Series

Additionally, a **multivariate time series (MTS)** represents multiple variables that are dependent on time, each of which is influenced by both past values (stated as "temporal" dependency) and other variables (dimensions) based on their correlation. The correlations between different variables are referred to as spatial or intermetric dependencies in the literature, and they are used interchangeably [117]. In the same example, air pressure and temperature would also be recorded every hour besides humidity level.

An example of an MTS with two dimensions is illustrated in Figure 1(b). Consider an MTS represented as a sequence of vectors over time, each vector at time $i$, $X_i$, consisting of $d$ dimensions:

$$X = (X_1, X_2, \ldots, X_t) = \left( \left( x_1^1, x_1^2, \ldots, x_1^d \right), \left( x_2^1, x_2^2, \ldots, x_2^d \right), \ldots, \left( x_t^1, x_t^2, \ldots, x_t^d \right) \right) \qquad (2)$$

Where $X_i = (x_i^1, x_i^2, \ldots, x_i^d)$ represents a data vector at time $i$, with each $x_i^j$ indicating the observation at time $i$ for the $j^{th}$ dimension, and $j = 1, 2, \ldots, d$, where $d$ is the total number of dimensions.

## 2.3 Time Series Decomposition

It is possible to decompose a time series $X$ into four components, each of which express a specific aspect of its movement [52]. The components are as follows:

— Secular trend: This is the long-term trend in the series, such as increasing, decreasing or stable. The secular trend represents the general pattern of the data over time and does not have to be linear. The change in population in a particular region over several years is an example of nonlinear growth or decay depending on various dynamic factors.

— Seasonal variations: Depending on the month, weekday, or duration, a time series may exhibit a seasonal pattern. Seasonality always occurs at a fixed frequency. For instance, a study of gas/electricity consumption shows that the consumption curve does not follow a similar pattern throughout the year. Depending on the season and the locality, the pattern is different.

— Cyclical fluctuations: A cycle is defined as an extended deviation from the underlying series defined by the secular trend and seasonal variations. Unlike seasonal effects, cyclical effects vary in onset and duration. Examples include economic cycles such as booms and recessions.

— Irregular variations: This refers to random, irregular events. It is the residual after all the other components are removed. A disaster such as an earthquake or flood can lead to irregular variations.

A time series can be mathematically described by estimating its four components separately, and each of them may deviate from the normal behaviour.

## 2.4 Anomalies in Time Series

According to [77], the term anomaly refers to a deviation from the general distribution of data, such as a single observation (point) or a series of observations (subsequence) that deviate greatly from the general distribution. A small portion of the dataset contains anomalies, indicating the dataset mostly follows a normal pattern. There may be considerable amounts of noise embedded in real-world data, and such noise may be irrelevant to the researcher [4]. The most meaningful deviations are usually those that are significantly different from the norm. In circumstances where noise is present, the main characteristics of the data are identical. In data domains such as time series, trend analysis and anomaly detection are closely related, but they are not equivalent [4]. It is possible to see changes in time series datasets owing to concept drift, which occurs when values and trends change over time gradually or abruptly [3, 128].

*2.4.1 Types of Anomalies.* Anomalies in UTS and MTS can be classified as temporal, intermetric, or temporal-intermetric anomalies [117]. In a time series, temporal anomalies can be compared with either their neighbours (local) or the whole time series (global), and they present different forms depending on their behaviour [107]. There are several types of temporal anomalies that commonly occur in UTS, all of which are shown in Figure 1(a). Temporal anomalies can also occur in the MTS and affect multiple dimensions or all dimensions. A subsequent anomaly may appear when an unusual pattern of behaviour emerges over time; however, each observation may not be considered an outlier by itself. As a result of a point anomaly, an unexpected event occurs at one

point in time, and it is assumed to be a short sequence. Different types of temporal anomalies are as follows:

— Global: They are spikes in the series, which are point(s) with extreme values compared to the rest of the series. A global anomaly, for instance, is an unusually large payment by a customer on a typical day. Considering a threshold, it can be described as Equation (3).

$$|x_t - \hat{x}_t| > threshold \tag{3}$$

Where $x$ is the output of the model and $\hat{x}$ is the desired output. If the difference between the output and actual point value is greater than a threshold, then it has been recognised as an anomaly. An example of a global anomaly is shown on the left side of Figure 1(a) where the point with value of -6 shows a large deviation from the time series.

— Contextual: A deviation from a given context is defined as a deviation from a neighbouring time point, defined here as one that lies within a certain range of proximity. These types of anomalies are small glitches in sequential data, which are deviated values from their neighbours. It is possible for a point to be normal in one context while an anomaly in another. For example, large interactions, such as those on boxing day, are considered normal, but not so on other days. The formula is the same as that of a global anomaly, but the threshold for finding anomalies differs. The threshold is determined by taking into account the context of neighbours:

$$threshold \approx \lambda \times \mathbf{var}(X_{t-w:t}) \tag{4}$$

Where $X_{t-w:t}$ refers to the context of the data point $x_t$ within a window of size $w$, $\mathbf{var}$ is the variance of the context of data point and $\lambda$ controlling coefficient for the threshold. The second blue highlight in Figure 1(a) is a contextual anomaly that occurs locally in a specific context.

— Seasonal: In spite of normal shapes and trends of the time series, their seasonality is unusual compared to the overall seasonality. An example is the number of customers in a restaurant during a week. Such a series has a clear weekly seasonality, so it makes sense to look for deviations in this seasonality and process the anomalous periods individually.

$$diss_s(S, \hat{S}) > threshold \tag{5}$$

Where $diss_s$ is a function measuring the dissimilarity between two subsequences, $S$ denotes the actual seasonality, and $\hat{S}$ is the expected seasonality of the subsequences. As demonstrated in the first red highlight of Figure 1(a), the seasonal anomaly changes the frequency of a rise and drop of data in the particular segment.

— Trend: The event that causes a permanent shift in the data to its mean and produces a transition in the trend of the time series. While this anomaly preserves its cycle and seasonality of normality, it drastically alters its slope. Trends can occasionally change direction, meaning they may go from increasing to decreasing and vice versa. As an example, when a new song comes out, it becomes popular for a while, then it disappears from the charts like the segment in Figure 1(a) where the trend is changed and is assumed as a trend anomaly. It is likely that the trend will restart in the future.

$$diss_t(T, \hat{T}) > threshold \tag{6}$$

Where $T$ is the actual trend and $\hat{T}$ is the normal trend.

— Shapelet: Shapelet means a distinctive, time series subsequence pattern. There is a subsequence whose time series pattern or cycle differs from the usual pattern found in the rest of the sequence. Variations in economic conditions, like the total demand for and

supply of goods and services, are often the cause of these fluctuations. In the short-run, these changes lead to periods of expansion and recession.

$$diss_c(C, \hat{C}) > threshold \qquad (7)$$

Where $C$ is the actual cycle or shape and $\hat{C}$ specifies the expected cycle or shape of the subsequences. For example, the last highlight in Figure 1(a) where the shape of the segment changed due to some fluctuations.

Having discussed various types of anomalies, we understand that these can often be characterised by the distance between the actual subsequence observed and the expected subsequence. In this context, **dynamic time warping (DTW)** [134], which optimally aligns two time series, is a valuable method for measuring this dissimilarity. Consequently, DTW's ability to accurately calculate temporal alignments makes it a suitable tool for anomaly detection applications, as evidenced in several studies [15, 161]. Moreover, MTS is composed of multiple dimensions (a.k.a. metrics [117, 163]) that each describe a different aspect of a complex entity. Spatial dependencies (correlations) among dimensions within an entity, also known as intermetric dependencies, can be linear or nonlinear. MTS would exhibit a wide range of anomalous behaviour if these correlations were broken. An example is shown in the left part of Figure 1(b). The correlation between power consumption in the first dimension (metric 1) and CPU usage in the second dimension (metric 2) usage is positive, but it breaks about 100th of a second after it begins. Such an anomaly is named the intermetric anomaly in this study:

$$\max_{\forall j, k \in D, j \neq k} \text{diss}_{\text{corr}} \left( \text{Corr}\left( X^j, X^k \right), \text{Corr}\left( X^j_{t+\delta t_j:t+w+\delta t_j}, X^k_{t+\delta t_k:t+w+\delta t_k} \right) \right) > \text{threshold} \qquad (8)$$

where $X^j$ and $X^k$ are different dimensions of the MTS, **Corr** denotes the correlation function that measures the relationship between two dimensions, $\delta t_j$ and $\delta t_k$ are time shifts that adjust the comparison windows for dimensions $j$ and $k$, accommodating asynchronous events or delays between observations, $t$ is the starting point of the time window, $w$ is the width of the time window, indicating the duration over which correlations are assessed, $\text{diss}_{\text{corr}}$ is a function that quantifies the divergence in correlation between the standard, long-term measurement and the dynamic, short-term measurement within the specified window, threshold is a predefined limit that determines when the divergence in correlations signifies an anomaly, and $D$ is the set of all dimensions within the MTS, with the comparison conducted between every unique pair $(j, k)$ where $j \neq k$.

Dimensionality reduction techniques, such as selecting a subset of critical dimensions based on domain knowledge or preliminary analysis, help manage the computational complexity that increases with the number of dimensions.

Where $X^j$ and $X^k$ are two different dimensions of MTS that are correlated, and *corr* measures the correlations between two dimensions. When this correlation deteriorates in the window $t : t + w$, it means that the coefficient deviates more than *threshold* from the normal coefficient.

Intermetric-temporal anomalies introduce added complexity and challenges in anomaly detection; however, they occasionally facilitate easier detection across temporal or various dimensional perspectives due to their simultaneous violation of intermetric and temporal dependencies, as illustrated on the right side of Figure 1(b).

## 3   Time Series Anomaly Detection Methods

Traditional methods offer varied approaches to time series anomaly detection. *Statistical-based* methods [186] aim to learn a statistical model of the normal behaviour of time series. In *clustering-based* approaches [133], a normal profile of time series windows is learned, and the distance to the centroid of the normal clusters is considered as an anomaly score, or clusters with a small number
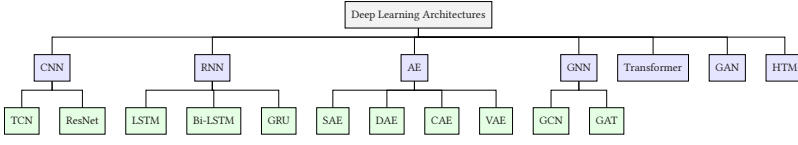
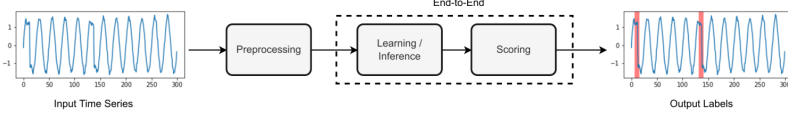Fig. 2. Deep learning architectures used in time series anomaly detection.



Fig. 3. General components of deep anomaly detection models in time series.

of members are considered as anomaly clusters. *Distances-based* approaches are extensively studied [188], in which the distance of a window of time series to its nearest neighbours is considered as an anomaly score. *Density-based* approaches [50] estimate the density of data points and time series windows with low density are detected as anomalies.

In data with complex structures, deep neural networks are powerful for modelling temporal and spatial dependencies in time series. A number of scholars have explored their application to anomaly detection using various deep architectures, as illustrated in Figure 2.

## 3.1 Deep Models for Time Series Anomaly Detection

An overview of deep anomaly detection models in time series is shown in Figure 3. In our study, deep models for anomaly detection in time series are categorised based on their main approach and architectures. There are three main approaches in the TSAD literature: *forecasting-based*, *reconstruction-based*, and *representation-based*. A categorisation of deep learning architectures in TSAD is shown in Figure 2.

The TSAD models are summarised in Tables 1 and 2 based on the input dimensions they process, which are UTS and MTS, respectively. These tables give an overview of the following aspects of the models: Temporal/Spatial, Learning scheme, Input, Interpretability, Point/Sub-sequence anomaly, Stochasticity, Incremental, and Univariate support. However, Table 1 excludes columns for Temporal/Spatial, Interpretability, and Univariate support as these features pertain solely to MTS. Additionally, it lacks an Incremental column because no univariate models incorporate an incremental approach.

*3.1.1 Temporal/Spatial.* With a UTS as input, a model can capture temporal information (i.e., pattern), while with an MTS as input, it can learn normality through both temporal and spatial dependencies. Moreover, if the model input is an MTS in which spatial dependencies are captured, the model can also detect intermetric anomalies (shown in Figure 1(b)).

*3.1.2 Learning Schemes.* In practice, training data tends to have a very small number of anomalies that are labelled. As a consequence, most of the models attempt to learn the representation or features of normal data. Based on anomaly definitions, anomalies are then detected by finding deviations from normal data. There are four learning schemes in the recent deep models for anomaly detection: unsupervised, supervised, semi-supervised, and self-supervised. These are based on the availability (or lack) of labelled data points. *Supervised* method employs a distinct method of learning the boundaries between anomalous and normal data that is based on all the labels in the training set. It can determine an appropriate threshold value that will be used for classifying all timestamps as anomalous if the anomaly score (Section 3.1) assigned to those

timestamps exceeds the threshold. The problem with this method is that it is not applicable to many real-world applications because anomalies are often unknown or improperly labelled. In contrast, *Unsupervised* approach uses no labels and makes no distinction between training and testing datasets. These techniques are the most flexible since they rely exclusively on intrinsic features of the data. They are useful in streaming applications because they do not require labels for training and testing. Despite these advantages, researchers may encounter difficulties evaluating anomaly detection models using unsupervised methods. The anomaly detection problem is typically treated as an unsupervised learning problem due to the inherently unlabelled nature of historical data and the unpredictable nature of anomalies. *Semi-supervised* anomaly detection in time series data may be utilised in cases where the dataset only consists of labelled normal data, unlike supervised methods that require a fully labelled dataset of both normal and anomalous points. Unlike unsupervised methods, which detect anomalies without any labelled data, semi-supervised TSAD relies on labelled normal data to define normal patterns and detect deviations as anomalies. This approach is distinct from *self-supervised* learning, where the model generates its own supervisory signal from the input data without needing explicit labels.

*3.1.3 Input.* A model may take an individual point (i.e., a time step) or a window (i.e., a sequence of time steps containing historical information) as an input. Windows can be used in order, called sliding windows, or shuffled without regard to the order, depending on the application. To overcome the challenges of comparing subsequences rather than points, many models use representations of subsequences (windows) instead of raw data and employ sliding windows that contain the history of previous time steps that rely on the order of subsequences within the time series data. A sliding window extraction is performed in the preprocessing phase after other operations have been implemented, such as imputing missing values, downsampling or upsampling of the data, and data normalisation.

*3.1.4 Interpretability.* In interpretation, the cause of an anomalous observation is given. Interpretability is essential when anomaly detection is used as a diagnostic tool since it facilitates troubleshooting and analysing anomalies. MTS are challenging to interpret, and stochastic deep learning complicates the process even further. A typical procedure to troubleshoot entity anomalies involves searching for the top dimension that differs most from previously observed behaviour. In light of that, it is, therefore, possible to interpret a detected entity anomaly by analysing several dimensions with the highest anomaly scores.

*3.1.5 Point/Subsequence Anomaly.* The model can detect either point anomalies or subsequence anomalies. A point anomaly is a point that is unusual when compared with the rest of the dataset. Subsequence anomalies occur when consecutive observations have unusual cooperative behaviour, although each observation is not necessarily an outlier on its own. Different types of anomalies are described in Section 2.4 and illustrated in Figures 1(a) and 1(b)

*3.1.6 Stochasticity.* As shown in Tables 1 and 2, we investigate the stochasticity of anomaly detection models as well. Deterministic models can accurately predict future events without relying on randomness. Predicting something that is deterministic is easy because you have all the necessary data at hand. The models will produce the same exact results for a given set of inputs in this circumstance. Stochastic models can handle uncertainties in the inputs. Through the use of a random component as an input, you can account for certain levels of unpredictability or randomness.

*3.1.7 Incremental.* This is a machine learning paradigm in which the model's knowledge extends whenever one or more new observations appear. It specifies a dynamic learning strategy

Table 1. Univariate Deep Anomaly Detection Models in Time Series

| A[1] | MA[2] | Model | Year | Su/Un[3] | Input[4] | P/S[5] | Stc[6] |
|---|---|---|---|---|---|---|---|
| Forecasting | RNN (3.2.1) | LSTM-AD [126] | 2015 | Un | P | Point | |
| | | DeepLSTM [28] | 2015 | Semi | P | Point | |
| | | LSTM RNN [19] | 2016 | Semi | P | Subseq | |
| | | LSTM-based [56] | 2019 | Un | W | – | |
| | | TCQSA [118] | 2020 | Su | P | – | |
| | HTM (3.2.4) | Numenta HTM [5] | 2017 | Un | – | – | |
| | | Multi HTM [182] | 2018 | Un | – | – | |
| Reconstruction | CNN (3.2.2) | SR-CNN [147] | 2019 | Un | W | Point + Subseq | |
| | VAE (3.3.2) | Donut [184] | 2018 | Un | W | Subseq | ✓ |
| | | Bagel [115] | 2018 | Un | W | Subseq | ✓ |
| | | Buzz [32] | 2019 | Un | W | Subseq | ✓ |
| | AE (3.3.1) | EncDec-AD [125] | 2016 | Semi | W | Point | |

[1]A: *Approach*, [2]MA: *Main Architecture*, [3]Su/Un: *Supervised/Unsupervised* | Values: [Su: *Supervised*, Un: *Unsupervised*, Semi: *Semi-supervised*, Self: *Self-supervised*], Input: *P: point / W: window*, [5]P/S: *Point/Sub-sequence*, [6]Stc: *Stochastic*, "−" *indicates a feature is not defined or mentioned.*

that can be used if training data becomes available gradually. The goal of incremental learning is to adapt a model to new data while preserving its past knowledge.

Moreover, the deep model processes the input in a step-by-step or end-to-end fashion (see Figure 3). In the first category (step-by-step), there is a learning module followed by an anomaly scoring module. It is possible to combine the two modules in the second category to learn anomaly scores using neural networks as an end-to-end process. An output of these models may be anomaly scores or binary labels for inputs. Contrary to algorithms whose objective is to improve representations, DevNet [141], for example, introduces deviation networks to detect anomalies by leveraging a few labelled anomalies to achieve end-to-end learning for optimising anomaly scores. End-to-end models in anomaly detection are designed to directly output the final classification of data points or subsequences as normal or anomalous, which includes the explicit labelling of these points. In contrast, step-by-step models typically generate intermediate outputs at each stage of the analysis, such as anomaly scores for each subsequence or point. These scores then require additional post-processing, such as thresholding, to determine if an input is anomalous. Common methods for establishing these thresholds include **Nonparametric Dynamic Thresholding (NDT)** [92] and **Peaks-Over-Threshold (POT)** [158], which help convert scores into final labels.

An anomaly score is mostly defined based on a loss function. In most of the reconstruction-based approaches, reconstruction probability is used, and in forecasting-based approaches, the prediction error is used to define an anomaly score. An anomaly score indicates the degree of an anomaly in each data point. Anomaly detection can be accomplished by ranking data points according to anomaly scores ($A_S$) and a decision score based on a *threshold* value:

$$|A_S| > threshold \tag{9}$$

Evaluation metrics that are used in these papers are introduced in Appendix 5.

Table 2. Multivariate Deep Anomaly Detection Models in Time Series

| A[1] | MA[2] | Model | Year | T/S[3] | Su/Un[4] | Input[5] | Int[6] | P/S[7] | Stc[8] | Inc[9] | US[10] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Forecasting | RNN (3.2.1) | LSTM-PRED [66] | 2017 | T | Un | W | ✓ | – | | | |
| | | LSTM-NDT [92] | 2018 | T | Un | W | ✓ | Subseq | | | |
| | | LGMAD [49] | 2019 | T | Semi | P | | Point | | | ✓ |
| | | THOC [156] | 2020 | T | Self | W | | Subseq | | | ✓ |
| | | AD-LTI [183] | 2020 | T | Un | P | | Point (frame) | | | |
| | CNN (3.2.2) | DeepAnt [135] | 2018 | T | Un | W | | Point + Subseq | | | |
| | | TCN-ms [78] | 2019 | T | Semi | W | | Subseq | | | ✓ |
| | | TimesNet [181] | 2023 | T | Un | W | | – | | | ✓ |
| | GNN (3.2.3) | GDN [45] | 2021 | S | Un | W | ✓ | – | | | |
| | | GTA* [34] | 2021 | ST | Semi | – | | – | | | |
| | | GANF [40] | 2022 | ST | Un | W | | | | | |
| | HTM (3.2.4) | RADM [48] | 2018 | T | Un | W | | – | | | |
| | Transformer (3.2.5) | SAND [160] | 2018 | T | Semi | W | | – | | | |
| | | GTA* [34] | 2021 | ST | Semi | – | | – | | | |
| Reconstruction | AE (3.3.1) | AE/DAE [150] | 2014 | T | Semi | P | | Point | | | |
| | | DAGMM [203] | 2018 | S | Un | P | | Point | ✓ | | |
| | | MSCRED [192] | 2019 | ST | Un | W | ✓ | Subseq | | | |
| | | USAD [10] | 2020 | T | Un | W | | Point | | | |
| | | APAE [70] | 2020 | T | Un | W | | – | | | |
| | | RANSynCoders [1] | 2021 | ST | Un | P | ✓ | Point | | ✓ | |
| | | CAE-Ensemble [22] | 2021 | T | Un | W | | Subseq | | | |
| | | AMSL [198] | 2022 | T | Self | W | | – | | | |
| | | ContextDA [106] | 2023 | T | Un | W | | Point + Subseq | | | |
| | VAE (3.3.2) | STORN [159] | 2016 | ST | Un | P | | Point | ✓ | | |
| | | GGM-VAE [74] | 2018 | T | Un | W | | Subseq | ✓ | | |
| | | LSTM-VAE [143] | 2018 | T | Semi | P | | – | ✓ | | |
| | | OmniAnomaly [163] | 2019 | T | Un | W | ✓ | Point + Subseq | ✓ | | |
| | | VELC [191] | 2019 | T | Un | – | | – | ✓ | | |
| | | SISVAE [112] | 2020 | T | Un | W | | Point | ✓ | | ✓ |
| | | VAE-GAN [138] | 2020 | T | Semi | W | | Point | ✓ | | |
| | | TopoMAD [79] | 2020 | ST | Un | W | | Subseq | ✓ | | |
| | | PAD [30] | 2021 | T | Un | W | | Subseq | ✓ | | ✓ |
| | | InterFusion [117] | 2021 | ST | Un | W | ✓ | Subseq | ✓ | | |
| | | MT-RVAE* [177] | 2022 | ST | Un | W | | – | ✓ | | |
| | | RDSMM [113] | 2022 | T | Un | W | | Point + Subseq | ✓ | | ✓ |
| | GAN (3.3.3) | MAD-GAN [111] | 2019 | ST | Un | W | | Subseq | | | |
| | | BeatGAN [200] | 2019 | T | Un | W | | Subseq | | | ✓ |
| | | DAEMON [33] | 2021 | T | Un | W | ✓ | Subseq | | | |
| | | FGANomaly [54] | 2021 | T | Un | W | | Point + Subseq | | | |
| | | DCT-GAN* [114] | 2021 | T | Un | W | | – | | | ✓ |
| | Transformer (3.3.4) | Anomaly Transformer [185] | 2021 | T | Un | W | | Subseq | | | |
| | | DCT-GAN* [114] | 2021 | T | Un | W | | – | | | ✓ |
| | | TranAD [171] | 2022 | T | Un | W | ✓ | Subseq | | | ✓ |
| | | MT-RVAE* [177] | 2022 | ST | Un | W | | – | | | ✓ |
| | | Dual-TF [136] | 2024 | T | Un | W | | Point + Subseq | | | ✓ |
| Representation | Transformer (3.4.1) | TS2Vec [190] | 2022 | T | Self | P | | Point | | | ✓ |
| | CNN (3.4.2) | TF-C [196] | 2022 | T | Self | W | | – | | | ✓ |
| | | DCdetector [187] | 2023 | ST | Self | W | | Point + Subseq | | | ✓ |
| | | CARLA [42] | 2023 | ST | Self | W | | Point + Subseq | | | ✓ |
| | | DACAD [43] | 2024 | ST | Self | W | | Point + Subseq | | | |
| Hybrid | AE (3.5.1) | CAE-M [197] | 2021 | ST | Un | W | | Subseq | | | |
| | | NSIBF* [60] | 2021 | T | Un | W | | Subseq | | | |
| | RNN (3.5.2) | TAnoGAN [13] | 2020 | T | Un | W | | Subseq | | | |
| | | NSIBF* [60] | 2021 | T | Un | W | | Subseq | | | |
| | GNN (3.5.3) | MTAD-GAT [199] | 2020 | ST | Self | W | ✓ | Subseq | | | |
| | | FuSAGNet [76] | 2022 | ST | Semi | W | | Subseq | | | |

[1]A: *Approach*, [2]MA: *Main Architecture*, [3]T/S: *Temporal/Spatial* | Values: [S:*Spatial*, T:*Temporal*, ST:*Spatio-Temporal*], [4]Su/Un: *Supervised/Unsupervised* | Values: [Su: *Supervised*, Un: *Unsupervised*, Semi: *Semi-supervised*, Self: *Self-supervised*], [5]Input: *P: point / W: window*, [6]Int: *Interpretability*, [7]P/S: *Point/Sub-sequence*, [8]Stc: *Stochastic*, [9]Inc: *Incremental*, [10]US: *Univarite support*, *Models with more than one main architecture.*, "−" indicates a feature is not defined or mentioned.
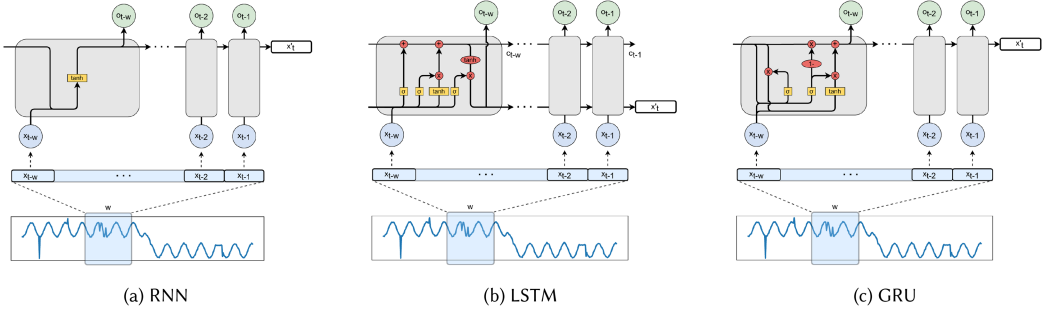
Fig. 4. An overview of (a) Recurrent neural network (RNN), (b) Long short-term memory unit (LSTM), and (c) Gated recurrent unit (GRU). These models can predict $x'_t$ by capturing the temporal information of a window of $w$ samples prior to $x_t$ in the time series. Using the error $|x_t - x'_t|$, an anomaly score can be computed.

## 3.2 Forecasting-based Models

The forecasting-based approach uses a learned model to predict a point or subsequence based on a point or a recent window. In order to determine how anomalous the incoming values are, the predicted values are compared to their actual values and their deviations are considered as anomalous values. Most forecasting methods use a sliding window to forecast one point at a time. This is especially helpful in real-world anomaly detection situations where normal behaviour is in abundance, but anomalous behaviour is rare.

It is worth mentioning that some previous works such as [124] use prediction error as a *novelty* quantification rather than an anomaly score. In the following subsections, different forecasting-based architectures are explained.

*3.2.1 Recurrent Neural Networks (RNN).* RNNs have internal memory, allowing them to process variable-length input sequences and retain temporal dynamics [2, 167]. An example of a simple RNN architecture is shown in Figure 4(a). Recurrent units take the points of the input window $X_{t-w:t-1}$ and forecast the next timestamp $x'_t$. The input sequence is processed iteratively, timestamp by timestamp. Given input $x_{t-1}$ to the recurrent unit $o_{t-2}$ and an activation function like tanh, the output $x'_t$ is calculated as follows:

$$x'_t = \sigma\left(W_{x'}.o_{t-1} + b_{x'}\right) ,$$
$$o_{t-1} = \tanh\left(W_o.x_{t-1} + U_o.o_{t-2} + b_h\right) \tag{10}$$

where $W_{x'}$, $W_o$, $U_o$, and $b_h$ are the network parameters. The network learns long-term and short-term temporal dependencies using previous outputs as inputs.

LSTM networks extend RNNs with memory lasting thousands of steps [82], enabling superior predictions through long-term dependencies. An LSTM unit, illustrated in Figure 4(b), comprises cells, input gates, output gates, and forget gates. The cell remembers values for variable time periods, while the gates control the flow of information.

In LSTM processing, the forget gate $f_{t-1}$ is calculated as:

$$f_{t-1} = \sigma\left(W_f.x_{t-1} + U_f.o_{t-2}\right) \tag{11}$$

$$i_{t-1} = \sigma\left(W_i.x_{t-1} + U_i.o_{t-2}\right) \tag{12}$$

$$s_{t-1} = \sigma\left(W_s.x_{t-1} + U_s.o_{t-2}\right) \tag{13}$$

Next, the candidate cell state $\tilde{c_{t-1}}$ is updated as:

$$\tilde{c_{t-1}} = \tanh\left(W_c.x_{t-1} + U_c.o_{t-2}\right) ,$$
$$c_{t-1} = i_{t-1}.\tilde{c_{t-1}} + f_{t-1}.c_{t-2} \tag{14}$$

Finally, the hidden state $o_{t-1}$ or output is:

$$o_{t-1} = \tanh(c_{t-1}) . s_{t-1} \tag{15}$$

where $W$ and $U$ are the parameters of the LSTM cell. $x'_t$ is finally calculated using Equation (10).

Experience with LSTM has shown that stacking recurrent hidden layers with sigmoidal activation units effectively captures the structure of time series data, allowing for processing at different time scales compared to other deep learning architectures [80]. LSTM-AD [126] possesses long-term memory capabilities and combines hierarchical recurrent layers to detect anomalies in UTS without using labelled data for training. This stacking helps learn higher-order temporal patterns without needing prior knowledge of their duration. The network predicts several future time steps to capture the sequence's temporal structure, resulting in multiple error values for each point in the sequence. These prediction errors are modelled as a multivariate Gaussian distribution to assess the likelihood of anomalies. LSTM-AD's results suggest that LSTM-based models are more effective than RNN-based models, especially when it's unclear whether normal behaviour involves long-term dependencies.

As opposed to the stacked LSTM used in LSTM-AD, Bontemps et al. [19] propose a simpler LSTM RNN model for collective anomaly detection based on its predictive abilities for UTS. First, an LSTM RNN is trained with normal time series data to make predictions, considering both current states and historical data. By introducing a circular array, the model detects collective anomalies by identifying prediction errors that exceed a certain threshold within a sequence.

Motivated by promising results in LSTM models for UTS anomaly detection, a number of methods attempt to detect anomalies in MTS based on LSTM architectures. In DeepLSTM [28], stacked LSTM recurrent networks are trained on normal time series data. The prediction errors are then fitted to a multivariate Gaussian using maximum likelihood estimation. This model predicts both normal and anomalous data, recording the **Probability Density Function (PDF)** values of the errors. This approach has the advantage of not requiring preprocessing, and it works directly on raw time series. LSTM-PRED [66] utilises three LSTM stacks with 100 hidden units each, processing data sequences of 100 seconds to learn temporal dependencies. Instead of setting thresholds for each sensor, it uses the **Cumulative Sum (CUSUM)** method to detect anomalies. CUSUM calculates the cumulative sum of the sequence predictions to identify small deviations, reducing false positives. It computes the positive and negative differences between predicted and actual values, setting **Upper Control Limits (UCL)** and **Lower Control Limits (LCL)** from the validation data to determine anomalies. Moreover, this model can pinpoint the specific sensor showing abnormal behaviour.

In all three above-mentioned models, LSTMs are stacked to improve prediction accuracy by analysing historical data from MTS; however, LSTM-NDT [92] combines various techniques. LSTM-NDT model introduces a technique that automatically adjusts thresholds for data changes, addressing issues like diversity and instability in evolving data. Another model, called LGMAD [49], enhances LSTM's structure for better anomaly detection in time series. Additionally, a method combines LSTM with a **Gaussian Mixture Model (GMM)** for detecting anomalies in both simple and complex systems, with a focus on assessing the system's health status through a health factor. This model can only be applied in low-dimensional applications. For high-dimensional data, it's suggested to use dimension reduction methods like PCA for effective anomaly detection [88].

Ergen and Kozat [56] present LSTM-based anomaly detection algorithms in an unsupervised framework, as well as semi-supervised and fully supervised frameworks. To detect anomalies, it uses scoring functions implemented by **One Class-SVM (OC-SVM)** and **Support Vector Data Description (SVDD)** algorithms. In this framework, LSTM and OC-SVM (or SVDD) architecture parameters are jointly trained with well-defined objective functions, utilising two joint optimisation approaches. The gradient-based joint optimisation method uses revised OC-SVM and
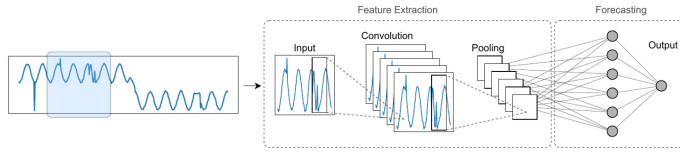
Fig. 5. Structure of a Convolutional Neural Network (CNN) predicting the next values of an input time series based on a previous data window. Time series dependency dictates that predictions rely solely on previously observed inputs.

SVDD formulations, illustrating their convergence to the original formulations. As a result of the LSTM-based structure, methods are able to process data sequences of variable length. Aside from that, the model is effective at detecting anomalies in time series data without preprocessing. Moreover, since the approach is generic, the LSTM architecture in this model can be replaced by a **GRU (gated recurrent neural networks)** architecture [38].

GRU was proposed by Cho et al. [36] in 2014, similar to LSTM but incorporating a more straightforward structure that leads to less computing time (see Figure 4(c)). Both LSTM and GRU use gated architectures to control information flow. However, GRU has gating units that inflate the information flow inside the unit without having any separate memory unit, unlike LSTM [47]. There is no output gate but an update gate and a reset gate. Figure 4(c) shows the GRU cell that integrates the new input with the previous memory using its reset gate. The update gate defines how much of the last memory to keep [73]. The issue is that LSTMs and GRUs are limited in learning complex seasonal patterns in multi-seasonal time series. As more hidden layers are stacked and the backpropagation distance (through time) is increased, accuracy can be improved. However, training may be costly.

In this regard, the AD-LTI model is a forecasting tool that combines a GRU network with a method called Prophet to learn seasonal time series data without needing labelled data. It starts by breaking down the time series to highlight seasonal trends, which are then specifically fed into the GRU network for more effective learning. When making predictions, the model considers both the overall trends and specific seasonal patterns like weekly and daily changes. However, since it uses past data that might include anomalies, the projections might not always be reliable. To address this, it introduces a new measure called **Local Trend Inconsistency (LTI)**, which assesses the likelihood of anomalies by comparing recent predictions against the probability of them being normal, overcoming the issue that there might be anomalous frames in history.

Traditional one-class classifiers are developed for fixed-dimension data and struggle with capturing temporal dependencies in time series data [149]. A recent model, called THOC [156], addresses this by using a complex network that includes a multilayer dilated RNN [27] and hierarchical SVDD [165]. This setup allows it to capture detailed temporal features at multiple scales (resolution) and efficiently recognise complex patterns in time series data. It improves upon older models by using information from various layers, not just the simplest features, and it detects anomalies by comparing current data against its normal pattern representation. In spite of the accomplishments of RNNs, they still face challenges in processing very long sequences due to their fixed window size.

*3.2.2 Convolutional Neural Networks (CNN).* **Convolutional Neural Networks (CNNs)** are adaptations of multilayer perceptrons designed to identify hierarchical patterns in data. These networks employ convolutional, pooling, and fully connected layers, as depicted in Figure 5. Convolutional layers utilise a set of learnable filters that are applied across the entire input to produce 2D activation maps through dot products. Pooling layers summarise these outputs statistically.

The CNN-based DeepAnt model [135] efficiently detects small deviations in time series patterns with minimal training data and can handle data contamination under 5% in an unsupervised setup. DeepAnt is applicable to both UTS and MTS and detects various anomaly types, including point, contextual anomalies, and discords.

Despite their effectiveness, traditional CNNs struggle with sequential data due to their inherent design. This limitation has been addressed by the development of **Temporal Convolutional Networks (TCN)** [11], which use dilated convolutions to accommodate time series data. TCNs ensure that outputs are the same length as inputs without future data leakage. This is achieved using a 1D fully convolutional network and dilated convolutions, ensuring all computations for a timestamp $t$ use only historical data. The dilated convolution operation is defined as:

$$x'(t) = (x *_l f)(t) = \sum_{i=0}^{k-1} f(i) \cdot x_{t-l \cdot i} \tag{16}$$

where $f$ is a filter of size $k$, $*_l$ denotes convolution with dilation factor $l$, and $x_{t-l \cdot i}$ represents past data points.

He and Zhao [78] use different methods to predict and detect anomalies in data over time. They use a TCN trained on normal data to forecast trends and calculate anomaly scores using multivariate Gaussian distribution fitted to prediction errors. It includes a skipping connection to blend multi-scale features, accommodating different pattern sizes. Ren et al. [147] combine a Spectral Residual model, originally for visual saliency detection [83], with a CNN to enhance accuracy. This method, used by over 200 Microsoft teams, can rapidly detect anomalies in millions of time series per minute. The **TCN Autoencoder (TCN-AE)**, developed by Thill et al. [169] (2020), modifies the standard AE by using CNNs instead of dense layers, making it more effective and adaptable. It uses two TCNs for encoding and decoding, with layers that respectively downsample and upsample data.

Many real-world scenarios produce **quasi-periodic time series (QTS)**, like the patterns seen in **ECGs (electrocardiograms)**. A new automated system for spotting anomalies in these QTS called AQADF [118], uses a two-part method. First, it segments the QTS into consistent periods using an algorithm (TCQSA) that uses a hierarchical clustering technique and groups similar data points without needing manual help, even filtering out errors to make it more reliable. Second, it analyses these segments with an attention-based hybrid LSTM-CNN model (HALCM), which looks at both broad trends and detailed features in the data. Furthermore, HALCM is further enhanced by three attention mechanisms, allowing it to capture more precise details of the fluctuation patterns in QTS. Specifically, TAGs are embedded in LSTMs in order to fine-tune variations extracted from different parts of QTS. A feature attention mechanism and a location attention mechanism are embedded into a CNN in order to enhance the effects of key features extracted from QTSs.

TimesNet [181] is a versatile deep learning model designed for comprehensive time series analysis. It transforms 1D time series data into 2D tensors to effectively capture complex temporal patterns. By using a modular structure called TimesBlock, which incorporates a parameter-efficient inception block, TimesNet excels in a variety of tasks, including forecasting, classification, and anomaly detection. This innovative approach allows it to handle intricate variations in time series data, making it suitable for applications across different domains.

*3.2.3 Graph Neural Networks (GNN).* In recent years, researchers have proposed extracting spatial information from MTS to form a graph structure, converting TSAD into a problem of detecting anomalies based on these graphs using GNNs. As shown in Figure 6, GNNs use pairwise message passing, where graph nodes iteratively update their representations by exchanging information. In MTS anomaly detection, each dimension is a node in the graph, represented as $V = \{1, \ldots, d\}$.
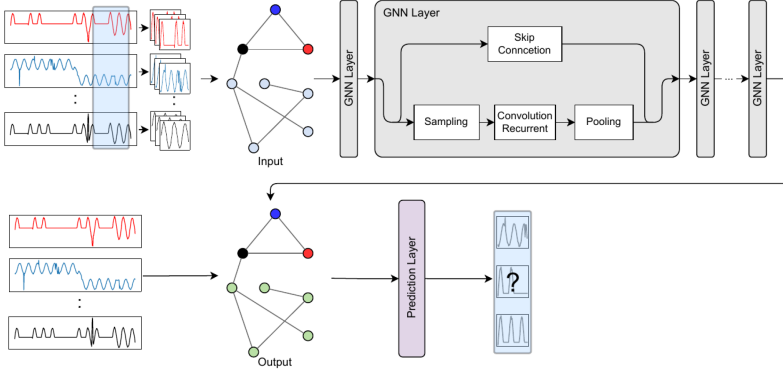
Fig. 6. The basic structure of Graph Neural Network (GNN) for MTS anomaly detection that can learn the relationships (correlations) between metrics and predict the expected behaviour of time series.

Edges $E$ indicate correlations learned from MTS. For node $u \in V$, the message passing layer outputs for iteration $k + 1$:

$$h_u^{k+1} = \text{UPDATE}^k \left( h_u^k, \ m_{N(u)}^k \right),$$
$$m_{N(u)}^k = \text{AGGREGATE}^k \left( h_i^k, \forall i \in N(u) \right)$$

(17)

Where $h_u^k$ is the embedding for each node and $N(u)$ is the neighbourhood of node $u$. GNNs enhance MTS modelling by learning spatial structures [151]. Various GNN architectures exist, such as **Graph Convolution Networks (GCN)** [103], which aggregate one-step neighbours, and **Graph Attention Networks (GAT)** [173], which use attention functions to compute different weights for each neighbour.

Incorporating relationships between features is beneficial. Deng and Hooi [45] introduced GDN, a GNN attention-based model that captures sensor characteristics as nodes and their correlations as edges, predicting behaviour based on adjacent sensors. Anomaly detection framework **GANF (Graph-Augmented Normalizing Flow)** [40] augments normalizing flow with graph structure learning, detecting anomalies by identifying low-density instances. GANF represents time series as a Bayesian network, learning conditional densities with a graph-based dependency encoder and using graph adjacency matrix optimisation [189].

In conclusion, extracting graph structures from time series and modelling them with GNNs enables the detection of spatial changes over time, representing a promising research direction.

*3.2.4 Hierarchical Temporal Memory (HTM).* **Hierarchical Temporal Memory (HTM)** mimics the hierarchical processing of the neocortex for anomaly detection [65]. Figure 7(a) shows the typical components of the HTM. The input $x_t$ is encoded and then processed through sparse spatial pooling [39], resulting in $a(x_t)$, a sparse binary vector. Sequence memory models temporal patterns in $a(x_t)$ and returns a sparse vector prediction $\pi(x_t)$. The prediction error is defined as:

$$err_t = 1 - \frac{\pi(x_{t-1}) \cdot a(x_t)}{|a(x_t)|}$$

(18)

Where $|a(x_t)|$ is the number of 1s in $a(x_t)$. Anomaly likelihood, based on the model's prediction history and error distribution, indicates whether the current state is anomalous.

HTM neurons are organised in columns within a layer (Figure 7(b)). Multiple regions exist within each hierarchical level, with fewer regions at higher levels combining patterns from lower

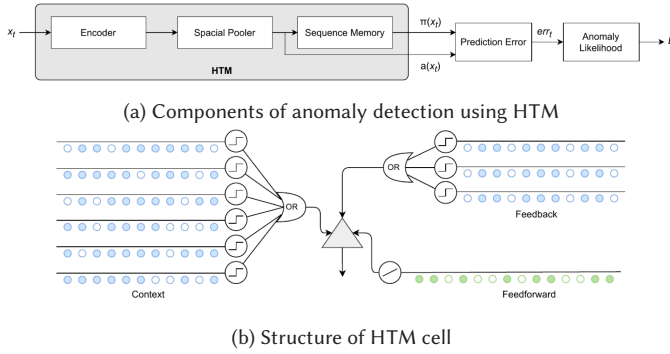(a) Components of anomaly detection using HTM



(b) Structure of HTM cell

Fig. 7. (a) Components of an HTM-based (Hierarchical Temporal Memory) anomaly detection system calculating prediction error and anomaly likelihood. (b) An HTM cell internal structure. Dendrites act as detectors with synapses. Context dendrites receive lateral input from other neurons. Sufficient lateral activity puts the cell in a predicted state.

levels to recognise more complex patterns. Sensory data enters lower-level regions during learning and generates patterns for higher levels. HTM is robust to noise, has high capacity, and can learn multiple patterns simultaneously. It recognises and memorises frequent spatial input patterns and identifies sequences likely to occur in succession.

Numenta HTM [5] detects temporal anomalies of UTS in predictable and noisy environments. It effectively handles extremely noisy data, adapts continuously to changes, and can identify small anomalies without false alarms. Multi-HTM [182] learns context over time, making it noise-tolerant and capable of real-time predictions for various anomaly detection challenges, so it can be used as an adaptive model. In particular, it is used for univariate problems and applied efficiently to MTS. RADM [48] proposes a real-time, unsupervised framework for detecting anomalies in MTS by combining HTM with a naive Bayesian network. Initially, HTM efficiently identifies anomalies in UTS with excellent results in terms of detection and response times. Then, it pairs with a Bayesian network to improve MTS anomaly detection without needing to reduce data dimensions, catching anomalies missed in UTS analyses. Bayesian networks help refine observations due to their adaptability and ease in calculating probabilities.

*3.2.5 Transformers.* Transformers [172] are deep learning models that weigh input data differently depending on the significance of different parts. In contrast to RNNs, transformers process the entire data simultaneously. Due to its architecture based solely on attention mechanisms, illustrated in Figure 8, it can capture long-term dependencies while being computationally efficient. Recent studies utilise them to detect time series anomalies as they process sequential data for translation in text data.

The original transformer architecture is encoder-decoder-based. An essential part of the transformer's functionality is its multi-head self-attention mechanism, stated in the following equation:

$$Q, K, V = softmax\left(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}}\right)V \tag{19}$$

where $Q$, $K$ and $V$ are defined as the matrices and $d_k$ is for normalisation of attention map.

A semantic correlation is identified in a long sequence, filtering out unimportant elements. Since transformers lack recurrence or convolution, they need positional encoding for token positions (i.e., relative or absolute positions). GTA [34] uses transformers for sequence modelling and a
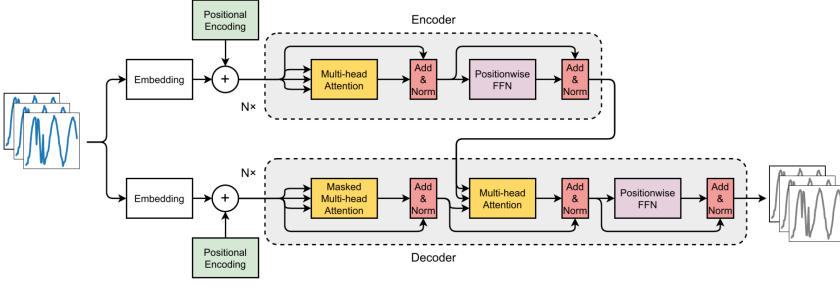
Fig. 8. Transformer network structure for anomaly detection. The Transformer uses an encoder-decoder structure with multiple identical blocks. Each encoder block includes a multi-head self-attention module and a feedforward network. During decoding, cross-attention is added between the self-attention module and the feedforward network.



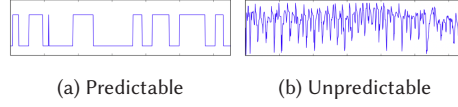(a) Predictable      (b) Unpredictable

Fig. 9. A time series may be unknown at any given moment or may change rapidly like (b), which illustrates sensor readings for manual control [125]. Such a time series cannot be predicted in advance, making prediction-based anomaly detection ineffective.

bidirectional graph to learn relationships among multiple IoT sensors. It introduces an **Influence Propagation (IP)** graph convolution for semi-supervised learning of sensor dependencies. To boost efficiency, each node's neighbourhood is constrained, and then graph convolution layers model information flow. As a next step, a multiscale dilated convolution and graph convolution are fused for hierarchical temporal context encoding. They use transformers for parallelism and contextual understanding and propose multi-branch attention to reduce attention complexity. In another recent work, SAnD [160] uses a transformer with stacked encoder-decoder structures, relying solely on attention mechanisms to model clinical time series. The architecture utilises self-attention to capture dependencies with multiple heads, positional encoding, and dense interpolation embedding for temporal order. It was also extended for multitask diagnoses.

## 3.3 Reconstruction-based Models

Many complex TSAD methods are designed around modelling the time series to predict future values, using prediction errors as indicators of anomalies. However, forecasting-based models often struggle with rapidly and continuously changing time series, as seen in Figure 9, where the future states of a series may be unpredictable due to rapid changes or unknown elements [68]. In such cases, these models tend to generate increased prediction errors as the number of time points grows [126], limiting their utility primarily to very short-term predictions. For example, in financial markets, forecasting-based methods might predict only the next immediate step, which is insufficient in anticipating or mitigating a potential financial crisis.

In contrast, reconstruction-based models can offer more accurate anomaly detection because they have access to current time series data, which is not available to forecasting-based models. This access allows them to effectively reconstruct a complete scenario and identify deviations. While these models might cause some delay in detection, they are preferred when high accuracy is paramount, and some delay is acceptable. Thus, reconstruction-based models are better suited for applications where precision is critical, even if it results in a minor delay in response.

Models for normal behaviour are constructed by encoding subsequences of normal training data in latent spaces (low dimensions). Model inputs are sliding windows (see Section 3) that provide the temporal context. We presume that the anomalous subsequences are less likely to be reconstructed compared to normal subsequences in the test phase since anomalies are rare. As a result, anomalies are detected by reconstructing a point/sliding window from test data and comparing them to the actual values, which is called reconstruction error. In some models, the detection of anomalies is triggered when the reconstruction probability is below a specified threshold since anomalous points/subsequences have a low reconstruction probability.

*3.3.1 Autoencoder (AE).* **Autoencoders (AEs)**, also known as auto-associative neural networks [105], are widely used in MTS anomaly detection for their nonlinear dimensionality reduction capabilities [150, 203]. Recent advancements in deep learning have focused on learning low-dimensional representations (encoding) using AEs [16, 81].

AEs consist of an encoder and a decoder (see Figure 10(a)). The encoder converts input into a low-dimensional representation, and the decoder reconstructs the input from this representation. The goal is to achieve accurate reconstruction and minimise reconstruction error. This process is summarised as follows:

$$Z_{t-w:t} = Enc\left(X_{t-w:t}, \phi\right), \quad \hat{X}_{t-w:t} = Dec\left(Z_{t-w:t}, \theta\right) \tag{20}$$

Where $X_{t-w:t}$ is a sliding window of input data, $x_t \in \mathbb{R}^d$, $Enc$ is the encoder with parameters $\phi$, and $Dec$ is the decoder with parameters $\theta$. $Z$ represents the latent space (encoded representation). The encoder and decoder parameters are optimised during training to minimise reconstruction error:

$$(\phi^*, \theta^*) = \arg\min_{\phi, \theta} Err\left(X_{t-w:t}, Dec\left(Enc\left(X_{t-w:t}, \phi\right), \theta\right)\right) \tag{21}$$

To improve representation, techniques such as **Sparse Autoencoder (SAE)** [137], **Denoising Autoencoder (DAE)** [174], and **Convolutional Autoencoder (CAE)** [139] are used. The anomaly score of a window in an AE-based model is defined based on the reconstruction error:

$$AS_w = ||X_{t-w:t} - Dec(Enc(X_{t-w:t}, \phi), \theta)||^2 \tag{22}$$

There are several papers in this category in our study. Sakurada and Yairi [150] show how AEs can be used for dimensionality reduction in MTS as a preprocessing step for anomaly detection. They treat each data sample at each time index as independent, disregarding the time sequence. Even though AEs already perform well without temporal information, they can be further boosted by providing current and past samples. The authors compare linear PCA, Denoising Autoencoders (DAEs), and kernel PCA, finding that AEs can detect anomalies that linear PCA is incapable of detecting. DAEs further enhance AEs. Additionally, AEs avoid the complex computations of kernel PCA without losing quality in detection. **DAGMM (Deep Autoencoding Gaussian Mixture Model)** [203] estimates the probability of MTS input samples using a Gaussian mixture prior to the latent space. It has two major components: a compression network for dimensionality reduction and an estimation network for anomaly detection using Gaussian Mixture Modelling to calculate anomaly scores in low-dimensional representations. However, DAGMM only considers spatial dependencies and lacks temporal information. The estimation network introduced a regularisation term that helps the compression network avoid local optima and reduce reconstruction errors through end-to-end training.

EncDec-AD [125] model detects anomalies from unpredictable UTS by using the first principal component of the MTS. It can handle time series up to 500 points long but faces issues with error accumulation for longer sequences. [98] proposes two AEs ensemble frameworks based on sparsely connected RNNs: one with independent AEs and another with multiple AEs trained

simultaneously, sharing features and using median reconstruction errors to detect outliers. Audibert et al. [10] propose **Unsupervised Anomaly Detection (USAD)** using AEs in which adversarially trained AEs are utilised to amplify reconstruction errors in MTS, distinguishing anomalies and facilitating quick learning. The input to USAD for either training or testing is in a temporal order. Goodge et al. [70] determine whether AEs are vulnerable to adversarial attacks in anomaly detection by analyzing the effects of various adversarial attacks. **APAE (Approximate Projection Autoencoder)** improves robustness against adversarial attacks by using gradient descent on latent representations and feature-weighting normalisation to account for variable reconstruction errors across features.

In MSCRED [192], attention-based ConvLSTM networks capture temporal trends, and a **convolutional autoencoder (CAE)** reconstructs a signature matrix, representing inter-sensor correlations instead of relying on the time series explicitly. The matrix length is 16, with a step interval of 5. An anomaly score is derived from the reconstruction error, aiding in anomaly detection, root cause identification, and anomaly duration interpretation. In CAE-Ensemble [22], a convolutional sequence-to-sequence autoencoder captures temporal dependencies with high parallelism. **Gated Linear Units (GLU)** with convolution layers and attention capture local patterns, recognising recurring subsequences like periodicity. The ensemble combines outputs from diverse models based on CAEs and uses a parameter-transfer training strategy, which enhances accuracy and reduces training time and error. In order to ensure diversity, the objective function also considers the differences between basic models rather than simply assessing their accuracy.

RANSysCoders [1] outlines a real-time anomaly detection system used by eBay. The authors propose an architecture with multiple encoders and decoders, using random feature selection and majority voting to infer and localise anomalies. The decoders set reconstruction bounds, functioning as bootstrapped AE for feature-bounds construction. The authors also recommend using spectral analysis of the latent space representation to extract priors for MTS synchronisation. Improved accuracy comes from feature synchronisation, bootstrapping, quantile loss, and majority voting. This method addresses issues with previous approaches, such as threshold identification, time window selection, downsampling, and inconsistent performance for large feature dimensions.

A novel **Adaptive Memory Network with Self-supervised Learning (AMSL)** [198] is designed to increase the generalisation of unsupervised anomaly detection. AMSL uses an AE framework with convolutions for end-to-end training. It combines self-supervised learning and memory networks to handle limited normal data. The encoder maps the raw time series and its six transformations into a feature space. A multi-class classifier is then used to classify these features and improve generalisation. The features are also processed through global and local memory networks, which learn common and specific features. Finally, an adaptive fusion module merges these features into a new reconstruction representation. Recently, ContextDA [106] utilises deep reinforcement learning to optimise domain adaptation for TSAD. It frames context sampling as a Markov decision process, focusing on aligning windows from the source and target domains. The model uses a discriminator to align these domains without leveraging label information in the source domain, which may lead to ineffective alignment when anomaly classes differ. ContextDA addresses this by leveraging source labels, enhancing the alignment of normal samples and improving detection accuracy.

*3.3.2 Variational Autoencoder (VAE).* Figure 10(b) shows a typical configuration of the **variational autoencoder (VAE)**, a directional probabilistic graph model which combines neural network autoencoders with mean-field variational Bayes [102]. The VAE works similarly to AE, but instead of encoding inputs as single points, it encodes them as a distribution using inference
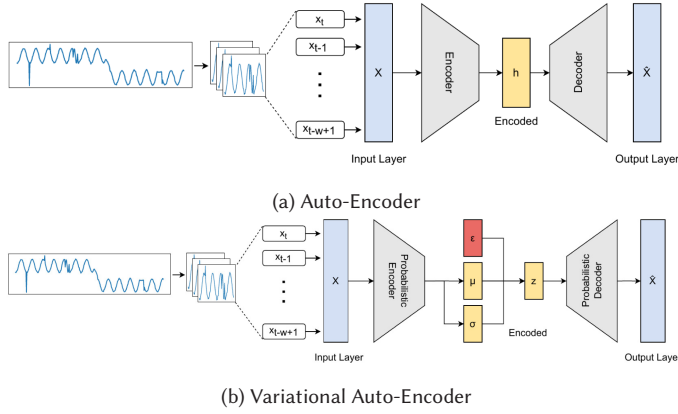
(a) Auto-Encoder



(b) Variational Auto-Encoder

Fig. 10. Structure of (a) Auto-Encoder that compresses an input window into a lower-dimensional representation ($h$) and then reconstructs the output $\hat{X}$ from this representation, and (b) Variational Auto-Encoder that its encoder compresses an input window of size $w$ into a latent distribution. The decoder uses sampled data from this distribution to produce $\hat{X}$, closely matching $X$.

network $q_\phi(Z_{t-w+1:t}|X_{t-w+1:t})$ where $\phi$ is its parameters. It represents a $d$ dimensional input $X_{t-w+1:t}$ to a latent representation $Z_{t-w+1:t}$ with a lower dimension $k < d$. A sampling layer takes a sample from a latent distribution and feeds it to the generative network $p_\theta(X_{t-w+1:t}|Z_{t-w+1:t})$ with parameters $\theta$, and its output is $g(Z_{t-w+1:t})$, reconstruction of the input. There are two components of the loss function, as stated in Equation (23) that are minimised in a VAE: a reconstruction error that aims to improve the process of encoding and decoding and a regularisation factor, which aims to regularise the latent space by making the encoder's distribution as close to the preferred distribution as possible.

$$loss = ||X_{t-w+1:t} - g(Z_{t-w+1:t})||^2 + KL(N(\mu_x, \sigma_x), \ N(0,1)) \tag{23}$$

Where $KL$ is the Kullback–Leibler divergence. By using regularised training, it avoids overfitting and ensures that the latent space is appropriate for a generative process.

LSTM-VAE [143] represents a variation of the VAE that uses LSTM instead of a feed-forward network. This model is trained with a denoising autoencoding method for better representation. It detects anomalies when the log-likelihood of a data point is below a dynamic, state-based threshold to reduce false alarms. Xu et al. [184] found that training on both normal and abnormal data is crucial for VAE anomaly detection. Their model, Donut, uses a VAE trained on shuffled data for unsupervised anomaly detection. Donut's Modified ELBO, Missing Data Injection, and MCMC Imputation make it excellent at detecting anomalies in the seasonal KPI dataset. However, due to VAE's nonsequential nature and sliding window format, Donut struggles with temporal anomalies. Later on, Bagel [115] is introduced to handle temporal anomalies robustly and unsupervised. Instead of using VAE in Donut, Bagel employs **conditional variational autoencoder (CVAE)** [109] and considers temporal information. VAE models the relationship between two random variables, $x$ and $z$. CVAE models the relationship between $x$ and $z$, conditioned on $y$, i.e., it models $p(x, z|y)$.

**STORNs** [159], or **stochastic recurrent networks**, use variational inference to model high-dimensional time series data. The algorithm is flexible and generic and doesn't need domain knowledge for structured time series. OmniAnomaly [163] uses a VAE with stochastic RNNs for robust representations of multivariate data and planar normalizing flow for non-Gaussian latent space distributions. It detects anomalies based on reconstruction probability and uses POT for thresholding.

InterFusion [117] uses a **hierarchical Variational Autoencoder (HVAE)** with two stochastic latent variables for intermetric and temporal representations, along with a two-view embedding. To prevent overfitting anomalies in training data, InterFusion employs prefiltering temporal anomalies. The paper also introduces MCMC imputation, MTS for anomaly interpretation, and IPS for assessing results.

There are a few studies on anomaly detection in noisy time series data. Buzz [32] uses an adversarial training method to capture patterns in univariate KPI with non-Gaussian noises and complex data distributions. This model links Bayesian networks with optimal transport theory using Wasserstein distance. **SISVAE (smoothness-inducing sequential VAE)** [112] detects point-level anomalies by smoothing before training a deep generative model using a Bayesian method. As a result, it benefits from the efficiency of classical optimisation models as well as the ability to model uncertainty with deep generative models. This model adjusts thresholds dynamically based on noise estimates, crucial for changing time series. Other studies have used VAE for anomaly detection, assuming a unimodal Gaussian distribution as a prior. Existing studies have struggled to learn the complex distribution of time series due to its inherent multimodality. The GRU-based Gaussian Mixture VAE [74] addresses this challenge of learning complex distributions by using GRU cells to discover time sequence correlations and represent multimodal data with a Gaussian Mixture.

In [191], a VAE with two extra modules is introduced: a Re-Encoder and a **Latent Constraint network (VELC)**. The Re-Encoder generates new latent vectors, and this complex setup maximises the anomaly score (reconstruction error) in both the original and latent spaces to accurately model normal samples. The VELC network prevents the reconstruction of untrained anomalies, leading to latent variables similar to the training data, which helps distinguish normal from anomalous data. The VAE and LSTM are integrated as a single component in PAD [30] to support unsupervised anomaly detection and robust prediction. The VAE minimises noise impact on predictions, while LSTMs help VAE capture long-term sequences. **Spectral residuals (SR)** [83] are also used to improve performance by assigning weights to each subsequence, indicating their normality.

**TopoMAD (topology-aware multivariate time series anomaly detector)** [79] is an anomaly detector in cloud systems that uses GNN, LSTM, and VAE for spatiotemporal learning. It's a stochastic seq2seq model that leverages topological information to identify anomalies using graph-based representations. The model replaces standard LSTM cells with graph neural networks (GCN and GAT) to capture spatial dependencies. To improve anomaly detection, models like VAE-GAN [138] use partially labelled data. This semi-supervised model integrates LSTMs into a VAE, training an encoder, generator, and discriminator simultaneously. The model distinguishes anomalies using both VAE reconstruction differences and discriminator results.

The recently developed **Robust Deep State Space Model (RDSSM)** [113] is an unsupervised density reconstruction-based model for detecting anomalies in MTS. Unlike many current methods, RDSSM uses raw data that might contain anomalies during training. It incorporates two transition modules to handle temporal dependency and uncertainty. The emission model includes a heavy-tail distribution error buffer, allowing it to handle contaminated and unlabelled training data robustly. Using this generative model, they created a detection method that manages fluctuating noise over time. This model provides adaptive anomaly scores for probabilistic detection, outperforming many existing methods.

In [177], a variational transformer is introduced for unsupervised anomaly detection in MTS. Instead of using a feature relationship graph, the model captures correlations through self-attention. The model's performance improves due to reduced dimensionality and sparse correlations. The transformer's positional encoding, or global temporal encoding, helps capture long-term dependencies. Multi-scale feature fusion allows the model to capture robust features from different time
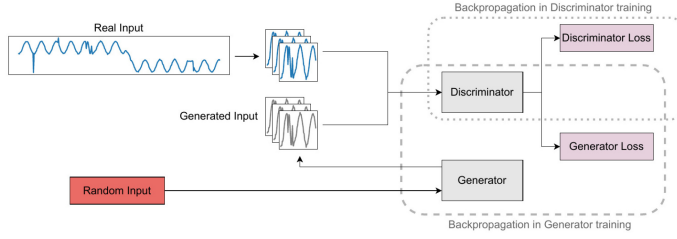
Fig. 11. Overview of a Generative Adversarial Network (GAN) with two main components: generator and discriminator. The generator creates fake time series windows for the discriminator, which learns to distinguish between real and fake data. A combined anomaly score is calculated using both the trained discriminator and generator.

scales. The residual VAE module encodes hidden space using local features, and its residual structure improves the KL divergence and enhances model generation.

*3.3.3 Generative Adversarial Networks (GAN).* A **generative adversarial network (GAN)** is an artificial intelligence algorithm designed for generative modelling based on game theory [69, 69]. In generative models, training examples are explored, and the probability distribution that generated them is learned. In this way, GAN can generate more examples based on the estimated distribution, as illustrated in Figure 11. Assume that we named the generator $G$ and the discriminator $D$. The generator and discriminator are trained using the following minimax model:

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p(X)}[log\, D(X_{t-w+1:t})] + \mathbb{E}_{z \sim p(Z)}[log(1 - D(Z_{t-w+1:t}))] \qquad (24)$$

where $p(x)$ is the probability distribution of input data and $X_{t-w+1:t}$ is a sliding window from the training set, called real input in Figure 11. Also, $p(z)$ is the prior probability distribution of the generated variable and $Z_{t-w+1:t}$ is a generated input window taken from a random space with the same window size.

In spite of the fact that GANs have been applied to a wide variety of purposes (mainly in research), they continue to involve unique challenges and research openings because they rely on game theory, which is distinct from most approaches to generative modelling. Generally, GAN-based models take into account the fact that adversarial learning makes the discriminator more sensitive to data outside the current dataset, making reconstructions of such data more challenging. BeatGAN [200] is able to regularise its reconstruction robustly because it utilises a combination of AEs and GANs [69] in cases where labels are not available. Moreover, using the time series warping method improves detection accuracy by speed augmentation in training datasets and robust BeatGAN against variability involving time warping in time series data. Research shows that BeatGAN can detect anomalies accurately in both ECG and sensor data.

However, training the GAN is usually difficult and requires a careful balance between the discriminator and generator [104]. A system based on adversarial training is not suitable for online use due to its instability and difficulty in convergence. With **Adversarial Autoencoder Anomaly Detection Interpretation (DAEMON)**, anomalies are detected using adversarially generated time series. DAEMON's training involves three steps. First, a one-dimensional CNN encodes MTS. Then, instead of decoding the hidden variable directly, a prior distribution is applied to the latent vector, and an adversarial strategy aligns the posterior distribution with the prior. This avoids inaccurate reconstructions of unseen patterns. Finally, a decoder reconstructs the time series, and another adversarial training step minimises differences between the original and reconstructed values.

**MAD-GAN (Multivariate Anomaly Detection with GAN)** [111] is a GAN-based model that uses LSTM-RNN as both the generator and discriminator to capture temporal relationships in time series. It detects anomalies using reconstruction error and discrimination loss. Furthermore, **FGANomaly (Filter GAN)** [54] tackles overfitting in AE-based and GAN-based anomaly detection models by filtering out potential abnormal samples before training using pseudo-labels. The generator uses Adaptive Weight Loss, assigning weights based on reconstruction errors during training, allowing the model to focus on normal data and reduce overfitting.

*3.3.4 Transformers.* Anomaly Transformer [185] uses an attention mechanism to spot unusual patterns by simultaneously modelling prior and series associations for each timestamp. This makes rare anomalies more distinguishable. Anomalies are harder to connect with the entire series, while normal patterns connect more easily with nearby timestamps. Prior associations estimate a focus on nearby points using a Gaussian kernel, while series associations use self-attention weights from raw data. Along with reconstruction loss, a MINIMAX approach is used to enhance the difference between normal and abnormal association discrepancies. TranAD [171] is another transformer-based model that has self-conditioning and adversarial training. As a result of its architecture, it is efficient for training and testing while preserving stability when dealing with huge input. When anomalies are subtle, transformer-based encoder-decoder networks may fail to detect them. However, TranAD's adversarial training amplifies reconstruction errors to fix this. Self-conditioning ensures robust feature retrieval, improving stability and generalisation.

Li et al. [114] present an unsupervised method called DCT-GAN, which uses a transformer to handle time series data, a GAN to reconstruct samples and spot anomalies, and dilated CNNs to capture temporal info from latent spaces. The model blends multiple transformer generators at different scales to enhance its generalisation and uses a weight-based mechanism to integrate generators, making it suitable for various anomalies. Additionally, MT-RVAE [177] significantly benefits from the transformer's sequence modelling and VAE capabilities that are categorised in both of these architectures.

The Dual-TF [136] is a framework for detecting anomalies in time series data by utilising both time and frequency information. It employs two parallel transformers to analyze data in these domains separately, then combines their losses to improve the detection of complex anomalies. This dual-domain approach helps accurately pinpoint both point-wise and subsequence-wise anomalies by overcoming the granularity discrepancies between time and frequency.

## 3.4 Representation-based Models

Representation-based models aim to learn rich representations of input time series that can then be used in downstream tasks such as anomaly detection and classification. In other words, rather than using the time series in the raw input space for anomaly detection, the learned representations in the latent space are used for anomaly detection. By learning robust representations, these models can effectively handle the complexities of time series data, which often contains noise, non-stationarity, and seasonality. These models are particularly useful in scenarios where labelled data is scarce, as they can often learn useful representations in an unsupervised or self-supervised learning schemes. While time series representation learning has become a hot topic in the time series community and a number of attempts have been made in recent years, only limited work has targeted anomaly detection tasks, and this area of research is still largely unexplored. In the following subsections we surveyed representation-based TSAD models.

*3.4.1 Transformers.* TS2Vec [190] utilises a hierarchical transformer architecture to capture contextual information at multiple scales, providing a universal representation learning approach using self-supervised contrastive learning that defines anomaly detection problem as a

downstream task across various time series datasets. In TS2Vec, positive pairs are representations at the same timestamp in two augmented contexts created by timestamp masking and random cropping, while negative samples are representations at different timestamps from the same series or from other series at the same timestamp within the batch.

*3.4.2 Convolutional Neural Networks (CNN).* **TF-C (Time-Frequency Consistency)** model [196] is a self-supervised contrastive pre-training framework designed for time series data. By leveraging both time-based and frequency-based representations, the model ensures that these embeddings are consistent within a shared latent space through a novel consistency loss. Using 3-layer 1-D ResNets as the backbone for its time and frequency encoders, the model captures the temporal and spectral characteristics of time series. This architecture allows the TF-C model to learn generalisable representations that can be used for time series anomaly detection in downstream tasks. In TF-C, a positive pair consists slightly perturbed version of an original sample, while a negative pair includes different original samples or their perturbed versions.

DCdetector [187] employs a deep CNN with a dual attention mechanism. This structure focuses on both spatial and temporal dimensions, using contrastive learning to enhance the separability of normal and anomalous patterns, making it adept at identifying subtle anomalies. In this model, a positive pair consists of representations from different views of the same time series, while it does not use negative samples and relies on the dual attention structure to distinguish anomalies by maximizing the representation discrepancy between normal and abnormal samples.

In contrast, CARLA [42] introduces a self-supervised contrastive representation learning approach using a two-phase framework. The first phase, called pretext, differentiates between anomaly-injected samples and original samples. In the second phase, self-supervised classification leverages information about the representations' neighbours to enhance anomaly detection by learning both normal behaviors and deviations indicating anomalies. In CARLA, positive pairs are selected from neighbours, while negative pairs are anomaly-injected samples. In the recent work, DACAD [43] combines a TCN with unsupervised domain adaptation techniques in its contrastive learning framework. It introduces synthetic anomalies to improve learning and generalisation across different domains, using a structure that effectively identifies anomalies through enhanced feature extraction and domain-invariant learning. DACAD selects positive pairs and negative pairs similar to CARLA.

These models exemplify the advancement in using deep learning for TSAD, highlighting the shift towards models that not only detect but also understand the intricate patterns in time series data, which makes this area of research promising. Finally, while all the models in this category are based on self-supervised contrastive learning approaches, there is no work on self-prediction-based self-supervised approaches in the TSAD literature and this research direction is unexplored.

## 3.5 Hybrid Models

These models integrate the strengths of different approaches to enhance time series anomaly detection. A forecasting-based model predicts the next timestamp, while a reconstruction-based model uses latent representations of the time series. Additionally, representation-based models learn comprehensive representations of the time series. By using a joint objective function, these combined models can be optimised simultaneously.

*3.5.1 Autoencoder (AE).* By capturing spatiotemporal correlation in multisensor time series, the **CAE-M (Deep Convolutional Autoencoding Memory network)** [197] can model generalised patterns based on normalised data by undertaking reconstruction and prediction simultaneously. It uses a deep convolutional AE with a **Maximum Mean Discrepancy (MMD)** penalty to match a target distribution in low dimensions, which helps prevent overfitting due to noise or anomalies.

To better capture temporal dependencies, it employs nonlinear bidirectional LSTMs with attention and linear autoregressive models. **Neural System Identification and Bayesian Filtering (NSIBF)** [60] is a new density-based TSAD approach for **Cyber-Physical Security (CPS)**. It uses a neural network with a state-space model to track hidden state uncertainty over time, capturing CPS dynamics. In the detection phase, Bayesian filtering is applied to the state-space model to estimate the likelihood of observed values. This combination of neural networks and Bayesian filters allows NSIBF to accurately detect anomalies in noisy CPS sensor data.

*3.5.2 Recurrent Neural Networks (RNN).* With TAnoGan [13], they have developed a method that can detect anomalies in time series if a limited number of examples are provided. TAnoGan has been evaluated using 46 NAB time series datasets covering a range of topics. Experiments have shown that LSTM-based GANs can outperform LSTM-based GANs when challenged with time series data through adversarial training.

*3.5.3 Graph Neural Networks (GNN).* In [199], two parallel **graph attention (GAT)** layers are introduced for self-supervised multivariate TSAD. These layers identify connections between different time series and learn relationships between timestamps. The model combines forecasting and reconstruction approaches: the forecasting model predicts one point, while the reconstruction model learns a latent representation of the entire time series. The model can diagnose anomalous time series (interpretability). FuSAGNet [76] fused SAE reconstruction and GNN forecasting to find complex anomalies in multivariate data. It incorporates GDN [45] but embeds sensors in each process, followed by recurrent units to capture temporal patterns. By learning recurrent sensor embeddings and sparse latent representations, the GNN predicts expected behaviours during the testing phase.

## 3.6 Model Selection Guidelines for Time Series Anomaly Detection

This section provides a concise guideline for choosing a TSAD method on specific characteristics of the data and the anomaly detection task at hand for practitioners to choose architectures that will provide the most accurate and efficient anomaly detection.

- **Multidimensional Data with Complex Dependencies:** GNNs are suitable for capturing both temporal and spatial dependencies in multivariate time series. They are particularly effective in scenarios such as IoT sensor networks and industrial systems where intricate interdependencies among dimensions exist. GNN architectures such as GCNs and GATs are suggested to be used in such settings.
- **Sequential Data with Long-Term Temporal Dependencies:** LSTM and GRU are effective for applications requiring the modelling of long-term temporal dependencies. LSTM is commonly used in financial time series analysis, predictive maintenance, and healthcare monitoring. GRU, with its simpler structure, offers faster training times and is suitable for efficient temporal dependency modelling.
- **Large Datasets Requiring Scalability and Efficiency:** Transformers utilise self-attention mechanisms to efficiently model long-range dependencies, making them suitable for handling large-scale datasets [97], such as network traffic analysis. They are designed for robust anomaly detection by capturing complex temporal patterns, with models like the Anomaly Transformer [185] and TranAD [171] being notable examples.
- **Handling Noise in Anomaly Detection:** AEs and VAEs architectures are particularly adept at handling noise in the data, making them suitable for applications like network traffic, multivariate sensor data, and cyber-physical systems.
- **High-Frequency Data and Detailed Temporal Patterns:** CNNs are useful for capturing local temporal patterns in high-frequency data. They are particularly effective in detecting

small deviations and subtle anomalies in data such as web traffic and real-time monitoring systems. TCNs extend CNNs by using dilated convolutions to capture long-term dependencies. As a result, they are suitable for applications where there exist long-range dependencies as well as local patterns [11].

— **Data with Evolving Patterns and Multimodal Distributions:** Combining the strengths of various architectures, hybrid models are designed to handle complex, high-dimensional time series data with evolving patterns like smart grid monitoring, industrial automation, and climate monitoring. These models, such as those integrating GNNs, VAEs, and LSTMs, are suitable for the mentioned applications.

— **Capturing Hierarchical and Multi-Scale Contexts:** HTM models are designed to capture hierarchical and multi-scale contexts in time series data. They are robust to noise and can learn multiple patterns simultaneously, making them suitable for applications involving complex temporal patterns and noisy data.

— **Generalisation across Diverse Datasets:** Contrastive learning excels in scenarios requiring generalisation across diverse datasets by learning robust representations through positive and negative pairs. It effectively distinguishes normal from anomalous patterns in time series data, making it ideal for applications with varying conditions, such as industrial monitoring, network security, and healthcare diagnostics.

## 4   Datasets

This section summarises datasets and benchmarks for TSAD, which provides a rich resource for researchers in TSAD. Some of these datasets are single-purpose datasets for anomaly detection, and some are general-purpose time series datasets that we can use in anomaly detection model evaluation with some assumptions or customisation. We can characterise each dataset or benchmark based on multiple aspects and their natural features. Here, we collect 48 well-known and/or highly-cited datasets examined by classic and **state-of-the-art (SOTA)** deep models for anomaly detection in time series. These datasets are characterised based on the below attributes:

— Nature of the data generation which can be real, synthetic or combined.
— Number of entities, which means the number of independent time series inside each dataset.
— Type of variety for each dataset or benchmark, which can be multivariate, univariate, or a combination of both.
— Number of dimensions, which is the number of features of an entity inside the dataset.
— Total number of samples of all entities in the dataset.
— The application domain of the dataset.

Note some datasets have been updated by their authors and contributors occasionally or regularly over time. We reported the latest update of the datasets and their attributes. Table 3 shows all 48 datasets with all mentioned attributes for each of them. It also includes hyperlinks to the primary source to download the latest version of the datasets.

Based on our exploration, the commonly used MTS datasets in SOTA TSAD models are MSL [92], SMAP [92], SMD [115], SWaT [129], PSM [1], and WADI [7]. For UTS, the commonly used datasets are Yahoo [93], KPI [25], NAB [5], and UCR [44]. These datasets are frequently used to benchmark and compare the performance of different TSAD models.

More detailed information about these datasets can be found on this Github repository: https://github.com/zamanzadeh/ts-anomaly-benchmark.

## 5   Evaluation Metrics for Time Series Anomaly Detection

Evaluating TSAD models is crucial for determining their effectiveness, especially in scenarios where anomalies are rare and tend to occur in sequences. Table 4 presents the key metrics used

Table 3. Public Dataset and Benchmarks used Mostly for Anomaly Detection in Time Series

| Dataset/Benchmark | Real/Synth | MTS/UTS[1] | # Samples[2] | # Entities[3] | # Dim[4] | Domain |
|---|---|---|---|---|---|---|
| CalIt2 [55] | Real | MTS | 10,080 | 2 | 2 | Urban events management |
| CAP [67, 168] | Real | MTS | 921,700,000 | 108 | 21 | Medical and health |
| CICIDS2017 [155] | Real | MTS | 2,830,540 | 15 | 83 | Server machines monitoring |
| Credit Card fraud detection [41] | Real | MTS | 284,807 | 1 | 31 | Fraud detectcion |
| DMDS [179] | Real | MTS | 725,402 | 1 | 32 | Industrial control systems |
| Engine Dataset [44] | Real | MTS | NA | NA | 12 | Industrial control systems |
| Exathlon [94] | Real | MTS | 47,530 | 39 | 45 | Server machines monitoring |
| GECCO IoT [132] | Real | MTS | 139,566 | 1 | 9 | Internet of things (IoT) |
| Genesis [175] | Real | MTS | 16,220 | 1 | 18 | Industrial control systems |
| GHL [63] | Synth | MTS | 200,001 | 48 | 22 | Industrial control systems |
| IOnsphere [55] | Real | MTS | 351 | | 32 | Astronomical studies |
| KDDCUP99 [51] | Real | MTS | 4,898,427 | 5 | 41 | Computer networks |
| Kitsune [55] | Real | MTS | 3,018,973 | 9 | 115 | Computer networks |
| MBD [79] | Real | MTS | 8,640 | 5 | 26 | Server machines monitoring |
| Metro [55] | Real | MTS | 48,204 | 1 | 5 | Urban events management |
| MIT-BIH Arrhythmia (ECG) [67, 131] | Real | MTS | 28,600,000 | 48 | 2 | Medical and health |
| MIT-BIH-SVDB [67, 71] | Real | MTS | 17,971,200 | 78 | 2 | Medical and health |
| MMS [79] | Real | MTS | 4,370 | 50 | 7 | Server machines monitoring |
| MSL [92] | Real | MTS | 132,046 | 27 | 55 | Aerospace |
| NAB-realAdExchange [5] | Real | MTS | 9,616 | 3 | 2 | Business |
| NAB-realAWSCloudwatch [5] | Real | MTS | 67,644 | 1 | 17 | Server machines monitoring |
| NASA Shuttle Valve Data [62] | Real | MTS | 49,097 | 1 | 9 | Aerospace |
| OPPORTUNITY [55] | Real | MTS | 869,376 | 24 | 133 | Computer networks |
| Pooled Server Metrics (PSM) [1] | Real | MTS | 132,480 | 1 | 24 | Server machines monitoring |
| PUMP [154] | Real | MTS | 220,302 | 1 | 44 | Industrial control systems |
| SMAP [92] | Real | MTS | 562,800 | 55 | 25 | Environmental management |
| SMD [115] | Real | MTS | 1,416,825 | 28 | 38 | Server machines monitoring |
| SWAN-SF [9] | Real | MTS | 355,330 | 5 | 51 | Astronomical studies |
| SWaT [129] | Real | MTS | 946,719 | 1 | 51 | Industrial control systems |
| WADI [7] | Real | MTS | 957,372 | 1 | 127 | Industrial control systems |
| NYC Bike [123] | Real | MTS/UTS | +25M | NA | NA | Urban events management |
| NYC Taxi [166] | Real | MTS/UTS | +200M | NA | NA | Urban events management |
| UCR [44] | Real/Synth | MTS/UTS | NA | NA | NA | Multiple domains |
| Dodgers Loop Sensor Dataset [55] | Real | UTS | 50,400 | 1 | 1 | Urban events management |
| KPI AIOPS [25] | Real | UTS | 5,922,913 | 58 | 1 | Business |
| MGAB [170] | Synth | UTS | 100,000 | 10 | 1 | Medical and health |
| MIT-BIH-LTDB [67] | Real | UTS | 67,944,954 | 7 | 1 | Medical and health |
| NAB-artificialNoAnomaly [5] | Synth | UTS | 20,165 | 5 | 1 | − |
| NAB-artificialWithAnomaly [5] | Synth | UTS | 24,192 | 6 | 1 | − |
| NAB-realKnownCause [5] | Real | UTS | 69,568 | 7 | 1 | Multiple domains |
| NAB-realTraffic [5] | Real | UTS | 15,662 | 7 | 1 | Urban events management |
| NAB-realTweets [5] | Real | UTS | 158,511 | 10 | 1 | Business |
| NeurIPS-TS [107] | Synth | UTS | NA | 1 | 1 | − |
| NormA [18] | Real/Synth | UTS | 1,756,524 | 21 | 1 | Multiple domains |
| Power Demand Dataset [44] | Real | UTS | 35,040 | 1 | 1 | Industrial control systems |
| SensoreScope [12] | Real | UTS | 621,874 | 23 | 1 | Internet of things (IoT) |
| Space Shuttle Dataset [44] | Real | UTS | 15,000 | 15 | 1 | Aerospace |
| Yahoo [93] | Real/Synth | UTS | 572,966 | 367 | 1 | Multiple domains |

[1]MTS/UTS: *Multivariate/Univariate*, [2]#samples: *total number of samples*, [3]#Entities: *number of distinct time series*,
[4]#Dim: *number of dimensions in MTS*.
There are direct hyperlinks to their names in the first column.

to assess TSAD model performance, including Precision, Recall, F1 Score, $F1_{PA}$ Score, **AU-PR (Area Under the Precision-Recall Curve)**, **AU-ROC (Area Under the Receiver Operating Characteristic Curve)**, **MTTD (Mean Time to Detect)**, Affliation [91], and VUS [142]. Detailed guidelines on when to utilise each metric and how to interpret their values are provided in Table 4.

Table 4. Evaluation Metrics for Time Series Anomaly Detection - Definitions and Guidelines

| Definitions and Formulas | | |
|---|---|---|
| **Metric** | **Definition** | **Formula**[*] |
| **Precision** | The proportion of true positive results among all positive results predicted by the model. In time series anomaly detection, it indicates the accuracy of the detected anomalies. | $\text{Precision} = \frac{TP}{TP+FP}$ |
| **Recall** | The proportion of true positive results among all actual positive cases. It measures the model's ability to detect all actual anomalies. | $\text{Recall} = \frac{TP}{TP+FN}$ |
| **F1 Score** | The harmonic mean of precision and recall, providing a balance between the two metrics. It is useful when both precision and recall are important. | $F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}}$ |
| **F1$_{PA}$ Score** | **F1$_{PA}$ score** is an F1 score that utilises a segment-based evaluation technique named point adjustment (PA), which at least one point within that segment is detected as abnormal [184]. This method can overestimate the performance of TSAD models (as mentioned in [42]). | $F1_{PA} = 2 \cdot \frac{\text{Precision}_{PA} \cdot \text{Recall}_{PA}}{\text{Precision}_{PA}+\text{Recall}_{PA}}$ |
| **PA%K** | **F1$_{PA}$** is mitigated by employing a **PA%K** protocol [152] by focusing on segments of data $w$. A segment is correctly detected as anomalous if at least $K\%$ of its points are TP ($TP_w$). | $Accuracy_w = \begin{cases} 1 & \text{if } \frac{TP_w}{|w|} \geq \frac{K}{100} \\ 0 & \text{otherwise} \end{cases}$ |
| **AU-PR** | Area Under Precision-Recall Curve is a performance measurement for classification problems at various thresholds ($t$). It is particularly useful for imbalanced datasets. | $\text{AU-PR} = \int_0^1 \text{Precision}(t) \frac{d(\text{Recall}(t))}{dt} \, dt$ |
| **AU-ROC** | Area Under Receiver Operating Characteristic Curve represents the ability of the model to distinguish between classes based on different thresholds ($t$). A higher AU-ROC indicates better model performance. | $\text{AU-ROC} = \int_0^1 \text{Recall}(t) \frac{d(\text{FPR}(t))}{dt} \, dt$ |
| **MTTD** | Mean Time to Detect is the average time taken to detect an anomaly at time $T_{\text{detect}}$ after it occurs in time $T_{\text{true}}$. This metric evaluates the model's responsiveness. | $\text{MTTD} = \frac{1}{n} \sum_{i=1}^n (T_{\text{detect}} - T_{\text{true}})$ |
| **Affiliation** | The affiliation metric assesses the degree of overlap between the detected anomalies (D) and the actual anomalies (A). It is designed to provide a more nuanced evaluation by considering both the precision and recall of the detected anomalies. | $\text{Affiliation} = \frac{|D \cap A|}{|D \cup A|}$ |
| **VUS** | The Volume Under the Surface quantifies the volume between the true anomaly signal $y$ and the predicted one, $\hat{y}$, over time. It captures both the temporal and amplitude differences between the two signals, providing a holistic measure of the detection performance. | $\text{VUS} = \int_0^T |y_t - \hat{y}_t| \, dt$ |
| Guideline to Use and Assess Evaluation Metrics | | |
| **Metric** | **Value Explanation** | **When to Use** |
| **Precision** | Low precision indicates many false alarms (normal instances classified as anomalies). High precision indicates most detected anomalies are actual anomalies, implying few false alarms. | Use when it is crucial to minimise false alarms and ensure that detected anomalies are truly significant. |
| **Recall** | Low recall indicates many true anomalies are missed, leading to undetected critical events. High recall indicates most anomalies are detected, ensuring prompt action on critical events. | Use when it is critical to detect all anomalies, even if it means tolerating some false alarms. |
| **F1** | Low F1 score indicates poor balance between precision and recall, leading to either many missed anomalies and/or many false alarms. High F1 score indicates a good balance, ensuring reliable anomaly detection with minimal misses and false alarms. | Use when a balance between precision and recall is needed to ensure reliable overall performance. |
| **F1$_{PA}$ Score** | Low FF1$_{PA}$ indicates difficulty in accurately identifying the exact points of anomalies. High F1$_{PA}$ indicates effective handling of slight deviations, ensuring precise anomaly detection. | Use when anomalies may not be precisely aligned, and slight deviations in detection points are acceptable. |
| **PA%K** | Low PA%K indicates that the model struggles to detect a sufficient portion of the anomalous segment. High PA%K indicates effective detection of segments, ensuring that a significant portion of the segment is identified as anomalous. | Use when evaluating the model's performance in detecting segments of anomalies rather than individual points. |
| **AU-PR** | Low AU-PR indicates poor model performance, especially with imbalanced datasets. High AU-PR indicates strong performance, maintaining high precision and recall across thresholds. | Use when dealing with imbalanced datasets, where anomalies are rare compared to normal instances. |
| **AU-ROC** | Low AU-ROC indicates the model struggles to distinguish between normal and anomalous patterns. High AU-ROC indicates effective differentiation, providing reliable anomaly detection. | Use for a general assessment of the model's ability to distinguish between normal and anomalous instances. |
| **MTTD** | High MTTD indicates significant delays in detecting anomalies. Low MTTD indicates quick detection, allowing prompt responses to critical events. | Use when the speed of anomaly detection is critical, and prompt action is required. |
| **Affiliation** | High value of the affiliation metric indicates a strong overlap or alignment between the detected anomalies and the true anomalies in a time series. | Use when a comprehensive evaluation is required, or the focus is early detection. |
| **VUS** | A lower VUS value indicates better performance, as it means the predicted anomaly signal is closer to the true signal. | Use when a holistic and threshold-free evaluation of TSAD methods is required. |

[*]TP: True Positive, FP: False Positive, TN: True Negative, FN: False Negative, FPR: FP/(FP + TN).

## 6 Interpretability Metrics

These metrics collectively offer a way to assess the interpretability of anomaly detection systems, specifically their ability to identify and prioritise the most relevant factors or dimensions contributing to each detected anomaly.

**HitRate@P%** is defined in [163] from HitRate@K used in recommender systems, modified to evaluate the accuracy of interpreting anomalies at the segment level. HitRate@P% assesses whether the true causes (relevant dimensions) of an anomaly are included within the top P% of the identified causes by the algorithm.

$$HitRate@P\% = \frac{\text{Number of true causes in top P\%}}{\text{Total number of true causes}} \tag{25}$$

**Interpretation Score (IPS)** [117] is adapted from the concept of HitRate@K, provides a precise measure of interpretative performance by quantifying the model's ability to pinpoint the most relevant factors contributing to each anomaly. It is typically defined in a manner that reflects the proportion of correctly identified causes within the top-k ranked items or factors, adjusted for their ranking order:

$$\text{IPS} = \frac{1}{N} \sum_{i=1}^{N} \frac{\text{Number of true causes in top k for segment } i}{\text{Total number of true causes in segment } i} \tag{26}$$

Where $N$ is the number of segments analysed, and the counts are taken from the top k causes identified by the model for each segment.

**RC-top-k (Relevant Causes top-k)** metric [64] measures the fraction of events for which at least one of the identified causes is among the top k causes identified by the model. This metric focuses on the model's ability to capture at least one relevant cause out of the potentially several contributing factors.

$$\text{RC-top-k} = \frac{\text{Number of events with at least one true cause in top k}}{\text{Total number of events}} \tag{27}$$

HitRate@P% rewards identifying all of the true causes while RC-top-k rewards identifying at least one of the causes.

**Reconstructed Discounted Cumulative Gain (RDCG@P%)** is an adaptation (defined by [33]) of the **Normalised Discounted Cumulative Gain (NDCG)**, a well-known metric in information retrieval used to evaluate ranking quality. For anomaly detection, RDCG@P% measures the effectiveness of the model in identifying the most relevant dimensions (causes) of an anomaly, based on their ranking according to the reconstruction error. The higher the error, the more likely it is that the dimension contributes significantly to the anomaly.

$$\text{RDCG@P\%} = \sum_{i=1}^{P} \frac{2^{r_i} - 1}{\log_2(i + 1)} \tag{28}$$

Where $r_i$ is the relevance score of the dimension at position $i$ in the ranking, up to the top P% of ranked dimensions.

## 7 Application Areas of Deep Anomaly Detection in Time Series

Applications generate data through processes that reflect system operations or provide observational information. Anomalies arise when these processes exhibit abnormal behaviour, revealing unusual characteristics of the systems or entities involved. Identifying these anomalies can offer valuable insights across various applications. Below, we categorise deep models based on their application areas.

Intrusion detection is a crucial task for network administrators. Traditional misuse detection methods struggle to identify new and unknown threats, whereas anomaly detection focuses on distinguishing between malicious events and normal network behaviour.

An essential part of defending a company's computer networks is the use of **network intrusion detection systems (NIDS)** to detect different security breaches. The feasibility and sustainability of contemporary networks are challenged by the need for increased human interaction and decreasing accuracy of detection. In [96], using deep learning techniques, they obtain a high-quality feature representation from unlabelled network traffic data and apply a supervised model on the KDD Cup 99 dataset [164]. Also, [8], a **Restricted Boltzmann Machine (RBM)** and a deep belief network are used for attack (anomaly) detection in KDD Cup 99. S-NDAE [157] is trained in an unsupervised manner to extract significant features from the dataset using unsupervised feature learning with **nonsymmetric deep autoencoders (NDAEs)**.

With the rapid expansion of mobile data traffic and the number of connected devices and applications, it is necessary to establish a network management system capable of predicting and detecting anomalies effectively. A measure of latency in these networks is the **round trip delay (RTT)** between a probe and a central server that monitors radio availability. RCAD [6] presents a distributed architecture for unsupervised detection of RTT anomalies, specifically increases in RTT. It employs the **hierarchical temporal memory (HTM)** algorithm to build a predictive model.

### 7.1 Medicine and Health

The widespread use of electronic health records has heightened the focus on predictive models for clinical time series data. New methods aim to analyse physiological time series, identify illness risks, and suggest mitigation measures. Wang et al. [176] employ convolutional layers to extract features, which are then used in a multivariate Gaussian distribution to detect anomalies.

**Electrocardiography (ECG)** signals are frequently used to assess the health of the heart. A complex organ like the heart can cause many different arrhythmias. Thus, it would be very beneficial to adopt an anomaly detection approach for analysing ECG signals which are developed in [98, 200] and [28].

**Cardiovascular diseases (CVDs)** are the leading cause of death in the world. Detecting abnormal heart rates can help doctors find patients' CVDs. Using a CNN, Rubin et al. [148] develop an automated recognition system for unusual heartbeats based on deep learning. In comparison to other popular deep models like CNN, RNNs are more effective at capturing the temporal characteristics of heartbeat sequences. A study on abnormal heartbeat detection using phonocardiography signals is presented in [148]. It has been shown that RNNs are capable of producing promising results even in the presence of noise. Also, Latif et al. [108] use RNNs because of their ability to model sequential and temporal data, even in noisy environments, to detect abnormal heartbeats automatically. [29] proposes a model using the classical **echo state network (ESN)** [95] trained on an imbalanced univariate heart rate dataset.

An epilepsy detection framework based on TCN, Gaussian mixture models and Bayesian inference called TCN-GMM [119] uses TCN to extract features from EEG time series. Also, it is possible to treat Alzheimer's disease more effectively if the disease is detected early. A 2D-CNN randomised ensemble model is presented in [120] that uses **magnetoencephalography (MEG)** synchronisation measures to detect early Alzheimer's disease symptoms.

### 7.2 Internet of Things (IoT)

As part of the smart world, the **Internet of Things (IoT)** is playing an increasingly significant role in monitoring various industrial equipment used in power plants and handling emergency situations [145]. Analysing data anomalies can identify environmental circumstances that require human attention, uncover outliers when cleaning sensor data, or save computing resources by

prefiltering undesirable portions of the data. Greenhouse [110] applies a multi-step ahead predictive LSTM over high volumes of IoT time series. A semi-supervised hierarchical stacking TCN is presented in [35], which targets the detection of anomalies in smart homes' communication. Due to their use of offline learning, these approaches are not resistant to changes in input distribution. In the **Industrial Internet of Things (IIoT)**, massive amounts of data are generated, which are valuable for monitoring the status of the underlying equipment and boosting operational performance. An LSTM-based model is presented in [195] for analysis and forecasting of sensor data from IIoT devices to capture the time span surrounding the failures. Kim et al. [99] perform unsupervised anomaly detection using real industrial IIoT time series, such as manufacturing CNC and UCI time series, using a **Squeezed Convolutional Variational Autoencoder (SCVAE)** deployed in an edge computing environment.

### 7.3 Server Machines Monitoring and Maintenance

Cloud systems have fueled the development of microservice architecture in the IT industry. A service failure in such an architecture can cause a series of failures, negatively impacting the customer experience and the company's revenue [53]. Troubleshooting needs to be performed as soon as possible after an incident. For this reason, continuously monitoring online systems for any anomalies is essential. SLA-VAE [89] uses a semi-supervised VAE to identify anomalies in MTS in order to enhance robustness. Using active learning, a framework is designed that can learn and update a detection model online based on a small sample size of highly uncertain data. Cloud server data from two different types of game businesses are used for the experiments. For each cloud server, 11 monitored metrics, such as CPU usage, CPU load, disk usage, and memory usage, are adopted.

Detecting anomalies is essential in **wireless sensor networks (WSNs)** as it can reveal information about equipment faults and previously unknown events. Luo and Nagarajan [122] introduce an AE-based model to solve anomaly detection problems in WSNs. The algorithm is designed to detect anomalies in sensors locally without requiring communication with other sensors or the cloud.

### 7.4 Urban Events Management

Traffic anomalies, like accidents and unexpected crowd gatherings, can threaten public safety if not promptly addressed. Detecting these anomalies is challenging due to the complex spatiotemporal dynamics of traffic data and the varying criteria across different locations and times. Zhang et al. [193] outline a spatiotemporal decomposition framework, which is proposed for detecting urban anomalies. Spatial and temporal features are derived using a graph embedding algorithm to adapt to different locations and times. [46] presents a traffic anomaly detection model based on a **spatiotemporal graph convolutional adversarial network (STGAN)**. Spatiotemporal generators can be used to capture the spatiotemporal dependencies of traffic data.

In order to model dynamic multivariate data effectively, CHAT [86] is devised. In CHAT, the authors model the urban anomaly prediction problem based on hierarchical attention networks. Uber uses an end-to-end neural network architecture for uncertainty estimation [201]. To improve anomaly detection accuracy, the proposed uncertainty estimate is used to measure the uncertainty of special events (such as holidays).

One of the most challenging tasks in transportation is forecasting the speed of traffic. The use of a traffic prediction system prior to travel in urban areas can help drivers avoid potential congestion and reduce travel time. The aim of GTransformer [121] is to study how GNNs can be combined with attention mechanisms to improve traffic prediction accuracy. Also, **TH-GAT** [87] is a **temporal hierarchical graph attention network** designed specifically for this purpose.

### 7.5 Astronomical Studies

As astronomical observations and data processing technology advance, an enormous amount of data is generated exponentially. "Light curve" is generated using a series of processing steps on a star image. Studying light curves contributes to astronomy as a new method for detecting abnormal astronomical events [116]. In [194], an LSTM neural network is proposed for predicting light curves.

### 7.6 Aerospace

Due to the complexity and cost of spacecraft, failure to detect hazards during flight could lead to serious or even catastrophic destruction. In [130], a transformer-based model with two novel components is presented, namely, an attention mechanism that updates timestamps concurrently and a masking strategy that detects anomalies in advance. Testing was conducted on NASA telemetry datasets.

Monitoring and diagnosing the health of **liquid rocket engines (LREs)** is the most significant concern for spacecraft and vehicle safety, particularly for human launch. Failure of the engine will result directly in the failure of the space launch, resulting in irreparable losses. To achieve reliable and automatic anomaly detection for large equipment such as LREs and multisource data, Feng et al. [61] suggest using a multimodal unsupervised method with missing sources.

### 7.7 Natural Disaster Detection

Earthquake prediction relies on detecting anomalies in precursor data, which can be categorised into two main types: tendency changes and high-frequency disturbances. A changing tendency occurs when the data deviates from its normal periodic pattern, while high-frequency disturbances are sudden, irregular changes with large amplitude. Cai et al. [21] develop a predictive model using LSTM units, which effectively detect earthquake precursors with minimal preprocessing.

The detection of earthquakes in real time requires a high-density network to fully leverage inexpensive sensors. Over the past few years, low-cost acceleration sensors have become widely used for accurate earthquake detection. Accordingly, Perol et al. [144] proposed CNNs for detecting earthquakes and locating them from two local stations in Oklahoma. Using deep CNNs, Phasenet [202] is able to determine the arrival time of earthquake waves in archives. In CrowdQuake [90], a convolutional RNN model is proposed as the core detection algorithm. Moreover, past acceleration data can be stored in databases and analysed post-hoc to identify earthquakes that may have been missed by real-time detection. In this model, abnormal sensors that might compromise earthquake events can be identified regularly.

### 7.8 Energy

It is inevitable that purification and refinement will affect various petroleum products. Regarding this, [63] as an LSTM-based approach is employed to monitor and detect faults in a multivariate industrial time series that includes signals from sensors and control systems of **gasoil plant heating loop (GHL)**. Likewise, according to Wen and Keyes [180], a CNN is used to detect time series anomalies using a transfer learning framework to solve data sparsity problems. The results were demonstrated on the GHL dataset [63], which contains data on cyber-attacks against utility systems.

The use of **phasor measurement units (PMU)** by utilities for power system monitoring increases the potential for cyberattacks. In [14], anomalies are detected in MTS data generated by PMU data packets corresponding to different events, such as line faults, trips, generation and load before each state estimation cycle. Consequently, it can help operators identify targeted cyber-attacks and make better decisions to ensure grid reliability.

The management of energy in buildings can improve energy efficiency, increase equipment life, and reduce energy consumption and operational costs. Fan et al. [59] propose an autoencoder-based ensemble method for the analysis of energy time series in buildings and the detection of unexpected consumption patterns and excessive waste.

### 7.9 Industrial Control Systems

System calls can be generated through regularly scheduled tasks, which are a consequence of events from a given process, and sometimes, they are caused by interrupts that are triggered by events. It is difficult to construct profiles using system call information since some processes are time-driven, event-driven, or both.

THREAT [58] provides a deeper insight into anomaly detection in system processes using their properties and system calls. Detecting anomalies at the kernel level provides new insights into the more complex machine-to-machine interactions. This is achieved by extracting useful features from system calls to detect a broad scope of anomalies.

An AE based on LSTM was implemented by Hsieh et al. [84] to detect anomalies in multivariate streams occurring in production equipment components. In this technique, LSTM networks are used to encode and decode actual values and evaluate deviations between reconstructed and actual values. Using CNN to handle MTS generated from semiconductor manufacturing processes is the basis for the model in [100]. Further, an MTS-CNN is proposed in [85] to detect anomalous wafers and provide useful information for root cause analysis in semiconductor production.

### 7.10 Robotics

In the modern manufacturing industry, as production lines become increasingly dependent on robots, failures of any robot can cause a plunge into a disastrous situation, while some faults are difficult to identify. In order to detect incipient failures in robots before they stop working completely, a real-time method is required to continuously track robots by collecting time series from robots. A **sliding-window convolutional variational autoencoder (SWCVAE)** is proposed in [31] to detect anomalies in MTS both spatially and temporally in an unsupervised manner.

Also, many people with disabilities require physical assistance from caregivers, although robots can substitute some human caregiving. Robots can help with daily living activities, such as feeding and shaving. By detecting and stopping abnormal task execution in assistance, potential hazards can be prevented or reduced [143].

### 7.11 Environmental Management

In ocean engineering, structures and systems are designed in or near the ocean, such as offshore platforms, piers and harbours, ocean wave energy conversion, and underwater life-support systems. The **ocean observing system (OOS)** provides marine data by using sensors and equipment that work under severe conditions. In order to prevent big losses from total machine failure or even natural disasters, it is necessary to detect OOS anomalies early enough. The real-time OceanWNN model [178] leverages a novel **WNN-based (Wavelet Neural Network)** method for detecting anomalies in ocean fixed-point observing time series without any labelled training data. **Wastewater treatment plants (WWTPs)** play a crucial role in protecting the environment. A method based on LSTMs was used by [127] to monitor the process and detect collective faults in WWTPS, superior to earlier methods. Moreover, energy management systems must manage gas storage and transportation continuously in order to reduce expenses and safeguard the environment. In [162], an end-to-end CNN-based model is used to implement an internal-flow-noise leak detector in pipes.

## 8 Discussion and Conclusion

In spite of the numerous advances in time series anomaly detection, there are still major challenges in detecting several types of anomalies (as described in Section 2.4). In contrast to the tasks relating to the majority (regular patterns), anomaly detection focuses on minority, unpredictable and unusual events, which bring about some challenges. The following are some challenges that have to be overcome in order to detect anomalies in time series data using deep learning models:

— System behaviour in the real world is highly dynamic and influenced by the prevailing environmental conditions, rendering time series data inherently non-stationary with frequently changing data distributions. This non-stationary nature necessitates the adaptation of deep learning models through online or incremental training approaches, enabling them to update continuously and detect anomalies in real-time. Such methodologies are crucial as they allow models to remain effective in the face of evolving patterns and sudden shifts, thereby ensuring timely and accurate anomaly detection.

— The detection of anomalies in multivariate high-dimensional time series data presents a particular challenge as data can become sparse in high dimension and the model requires simultaneous consideration of both temporal dependencies and relationships between dimensions.

— In the absence of labelled anomalies, unsupervised, semi-supervised or self-supervised approaches are required. Because of this, a large number of normal instances are incorrectly identified as anomalies. Hence, one of the key challenges is to find a mechanism for minimising false positives and improve recall rates of detection.

— Time series datasets can exhibit significant differences in noise existence, and noisy instances may be irregularly distributed. Thus, models are vulnerable, and their performance is compromised by noise in the input data.

— The use of anomaly detection for diagnostic purposes requires interpretability. Even so, anomaly detection research focuses primarily on detection precision, failing to address the issue of interpretability.

— In addition to being rarely addressed in the literature, anomalies that occur on a periodic basis make detection more challenging. A periodic subsequence anomaly is a subsequence that repeats over time [146]. The periodic subsequence anomaly detection technique, in contrast to point anomaly detection, can be adapted in areas like fraud detection to identify periodic anomalous transactions over time.

The main objective of this study was to explore and identify state-of-the-art deep learning models for TSAD, industrial applications, and datasets. In this regard, a variety of perspectives have been explored regarding the characteristics of time series, types of anomalies in time series, and the structure of deep learning models for TSAD. On the basis of these perspectives, 64 recent deep models were comprehensively discussed and categorised. Moreover, time series deep anomaly detection applications across multiple domains were discussed along with datasets commonly used in this area of research. In the future, active research efforts on time series deep anomaly detection are necessary to overcome the challenges we discussed in this survey.

## References

[1] Ahmed Abdulaal, Zhuanghua Liu, and Tomer Lancewicki. 2021. Practical approach to asynchronous multivariate time series anomaly detection and localization. In *KDD*. 2485–2494.

[2] Oludare Isaac Abiodun, Aman Jantan, Abiodun Esther Omolara, Kemi Victoria Dada, Nachaat AbdElatif Mohamed, and Humaira Arshad. 2018. State-of-the-art in artificial neural network applications: A survey. *Heliyon* 4, 11 (2018), e00938.

[3] Charu C. Aggarwal. 2007. *Data Streams: Models and Algorithms*. Vol. 31. Springer.

[4] Charu C. Aggarwal. 2017. An introduction to outlier analysis. In *Outlier Analysis*. Springer, 1–34.

[5] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. 2017. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing* 262 (2017), 134–147.

[6] Azza H. Ahmed, Michael A. Riegler, Steven A. Hicks, and Ahmed Elmokashfi. 2022. RCAD: Real-time collaborative anomaly detection system for mobile broadband networks. In *KDD*. 2682–2691.

[7] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P. Mathur. 2017. WADI: A water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. 25–28.

[8] Khaled Alrawashdeh and Carla Purdy. 2016. Toward an online anomaly intrusion detection system based on deep learning. In *ICMLA*. IEEE, 195–200.

[9] Rafal Angryk, Petrus Martens, Berkay Aydin, Dustin Kempton, Sushant Mahajan, Sunitha Basodi, Azim Ahmadzadeh, Xumin Cai, Soukaina Filali Boubrahimi, Shah Muhammad Hamdi, Micheal Schuh, and Manolis Georgoulis. 2020. SWAN-SF. (2020). https://doi.org/10.7910/DVN/EBCFKM

[10] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. 2020. USAD: Unsupervised anomaly detection on multivariate time series. In *KDD*. 3395–3404.

[11] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).

[12] Guillermo Barrenetxea. 2019. SensorScope Data. (April 2019). https://doi.org/10.5281/zenodo.2654726

[13] Md. Abul Bashar and Richi Nayak. 2020. TAnoGAN: Time series anomaly detection with generative adversarial networks. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 1778–1785.

[14] Sagnik Basumallik, Rui Ma, and Sara Eftekharnejad. 2019. Packet-data anomaly detection in PMU-based state estimator using convolutional neural network. *International Journal of Electrical Power & Energy Systems* 107 (2019), 690–702.

[15] Seif-Eddine Benkabou, Khalid Benabdeslem, and Bruno Canitia. 2018. Unsupervised outlier detection for time series by entropy and dynamic time warping. *Knowledge and Information Systems* 54, 2 (2018), 463–486.

[16] Siddharth Bhatia, Arjit Jain, Pan Li, Ritesh Kumar, and Bryan Hooi. 2021. MSTREAM: Fast anomaly detection in multi-aspect streams. In *WWW*. 3371–3382.

[17] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A. Lozano. 2021. A review on outlier/anomaly detection in time series data. *CSUR* 54, 3 (2021), 1–33.

[18] Paul Boniol, Michele Linardi, Federico Roncallo, Themis Palpanas, Mohammed Meftah, and Emmanuel Remy. 2021. Unsupervised and scalable subsequence anomaly detection in large data series. *The VLDB Journal* 30, 6 (2021), 909–931.

[19] Loïc Bontemps, Van Loi Cao, James McDermott, and Nhien-An Le-Khac. 2016. Collective anomaly detection based on long short-term memory recurrent neural networks. In *FDSE*. Springer, 141–152.

[20] Mohammad Braei and Sebastian Wagner. 2020. Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433* (2020).

[21] Yin Cai, Mei-Ling Shyu, Yue-Xuan Tu, Yun-Tian Teng, and Xing-Xing Hu. 2019. Anomaly detection of earthquake precursor data using long short-term memory networks. *Applied Geophysics* 16, 3 (2019), 257–266.

[22] David Campos, Tung Kieu, Chenjuan Guo, Feiteng Huang, Kai Zheng, Bin Yang, and Christian S. Jensen. 2021. Unsupervised time series outlier detection with diversity-driven convolutional ensembles. *VLDB* 15, 3 (2021), 611–623.

[23] Ander Carreño, Iñaki Inza, and Jose A. Lozano. 2020. Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework. *Artificial Intelligence Review* 53, 5 (2020), 3575–3594.

[24] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

[25] International AIOPS Challenges. 2018. KPI Anomaly Detection. (2018). https://competition.aiops-challenge.com/home/competition/1484452272200032281

[26] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *CSUR* 41, 3 (2009), 1–58.

[27] Shiyu Chang, Yang Zhang, Wei Han, Mo Yu, Xiaoxiao Guo, Wei Tan, Xiaodong Cui, Michael Witbrock, Mark A. Hasegawa-Johnson, and Thomas S. Huang. 2017. Dilated recurrent neural networks. *NeurIPS* 30 (2017).

[28] Sucheta Chauhan and Lovekesh Vig. 2015. Anomaly detection in ECG time signals via deep long short-term memory networks. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE, 1–7.

[29] Qing Chen, Anguo Zhang, Tingwen Huang, Qianping He, and Yongduan Song. 2020. Imbalanced dataset-based echo state networks for anomaly detection. *Neural Computing and Applications* 32, 8 (2020), 3685–3694.

[30] Run-Qing Chen, Guang-Hui Shi, Wan-Lei Zhao, and Chang-Hui Liang. 2021. A joint model for IT operation series prediction and anomaly detection. *Neurocomputing* 448 (2021), 130–139.

[31] Tingting Chen, Xueping Liu, Bizhong Xia, Wei Wang, and Yongzhi Lai. 2020. Unsupervised anomaly detection of industrial robots using sliding-window convolutional variational autoencoder. *IEEE Access* 8 (2020), 47072–47081.

[32] Wenxiao Chen, Haowen Xu, Zeyan Li, Dan Pei, Jie Chen, Honglin Qiao, Yang Feng, and Zhaogang Wang. 2019. Unsupervised anomaly detection for intricate KPIs via adversarial training of VAE. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 1891–1899.

[33] Xuanhao Chen, Liwei Deng, Feiteng Huang, Chengwei Zhang, Zongquan Zhang, Yan Zhao, and Kai Zheng. 2021. DAEMON: Unsupervised anomaly detection and interpretation for multivariate time series. In *ICDE*. IEEE, 2225–2230.

[34] Zekai Chen, Dingshuo Chen, Xiao Zhang, Zixuan Yuan, and Xiuzhen Cheng. 2021. Learning graph structures with transformer for multivariate time series anomaly detection in IoTt. *IEEE Internet of Things Journal* (2021).

[35] Yongliang Cheng, Yan Xu, Hong Zhong, and Yi Liu. 2019. HS-TCN: A semi-supervised hierarchical stacking temporal convolutional network for anomaly detection in IoT. In *IPCCC*. IEEE, 1–7.

[36] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. 103–111.

[37] Kukjin Choi, Jihun Yi, Changhwa Park, and Sungroh Yoon. 2021. Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines. *IEEE Access* (2021).

[38] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[39] Yuwei Cui, Subutai Ahmad, and Jeff Hawkins. 2017. The HTM spatial pooler–a neocortical algorithm for online sparse distributed coding. *Frontiers in Computational Neuroscience* (2017), 111.

[40] Enyan Dai and Jie Chen. 2022. Graph-augmented normalizing flows for anomaly detection of multiple time series. In *ICLR*.

[41] Andrea Dal Pozzolo, Olivier Caelen, Reid A. Johnson, and Gianluca Bontempi. 2015. Calibrating probability with undersampling for unbalanced classification. In *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, 159–166.

[42] Zahra Zamanzadeh Darban, Geoffrey I. Webb, Shirui Pan, Charu C. Aggarwal, and Mahsa Salehi. 2024. CARLA: Self-supervised contrastive representation learning for time series anomaly detection. *Pattern Recognition* (2024), 110874.

[43] Zahra Zamanzadeh Darban, Yiyuan Yang, Geoffrey I. Webb, Charu C. Aggarwal, Qingsong Wen, and Mahsa Salehi. 2024. DACAD: Domain adaptation contrastive learning for anomaly detection in multivariate time series. *arXiv preprint arXiv:2404.11269* (2024).

[44] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. 2018. The UCR Time Series Classification Archive. (October 2018). https://www.cs.ucr.edu/~eamonn/time_series_data_2018/

[45] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *AAAI*, Vol. 35. 4027–4035.

[46] Leyan Deng, Defu Lian, Zhenya Huang, and Enhong Chen. 2022. Graph convolutional adversarial networks for spatiotemporal anomaly detection. *TNNLS* 33, 6 (2022), 2416–2428.

[47] Rahul Dey and Fathi M. Salem. 2017. Gate-variants of gated recurrent unit (GRU) neural networks. In *IEEE 60th International Midwest Symposium on Circuits and Systems*. IEEE, 1597–1600.

[48] Nan Ding, Huanbo Gao, Hongyu Bu, Haoxuan Ma, and Huaiwei Si. 2018. Multivariate-time-series-driven real-time anomaly detection based on Bayesian network. *Sensors* 18, 10 (2018), 3367.

[49] Nan Ding, HaoXuan Ma, Huanbo Gao, YanHua Ma, and GuoZhen Tan. 2019. Real-time anomaly detection based on long short-term memory and Gaussian mixture model. *Computers & Electrical Engineering* 79 (2019), 106458.

[50] Zhiguo Ding and Minrui Fei. 2013. An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window. *IFAC Proceedings Volumes* 46, 20 (2013), 12–17.

[51] Third International Knowledge Discovery and Data Mining Tools Competition. 1999. KDD Cup 1999 Data. (1999). https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[52] Yadolah Dodge. 2008. *Time Series*. Springer New York, New York, NY, 536–539. https://doi.org/10.1007/978-0-387-32833-1_401

[53] Nicola Dragoni, Saverio Giallorenzo, Alberto Lluch Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, and Larisa Safina. 2017. Microservices: Yesterday, today, and tomorrow. *Present and Ulterior Software Engineering* (2017), 195–216.

[54] Bowen Du, Xuanxuan Sun, Junchen Ye, Ke Cheng, Jingyuan Wang, and Leilei Sun. 2021. GAN-based anomaly detection for multivariate time series using polluted training set. *TKDE* (2021).

[55] Dheeru Dua and Casey Graff. 2017. UCI Machine Learning Repository. (2017).

[56] Tolga Ergen and Suleyman Serdar Kozat. 2019. Unsupervised anomaly detection with LSTM neural networks. *TNNLS* 31, 8 (2019), 3127–3141.

[57] Philippe Esling and Carlos Agon. 2012. Time-series data mining. *CSUR* 45, 1 (2012), 1–34.

[58] Okwudili M. Ezeme, Qusay Mahmoud, and Akramul Azim. 2020. A framework for anomaly detection in time-driven and event-driven processes using kernel traces. *TKDE* (2020).

[59] Cheng Fan, Fu Xiao, Yang Zhao, and Jiayuan Wang. 2018. Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data. *Applied Energy* 211 (2018), 1123–1135.

[60] Cheng Feng and Pengwei Tian. 2021. Time series anomaly detection for cyber-physical systems via neural system identification and Bayesian filtering. In *KDD*. 2858–2867.

[61] Yong Feng, Zijun Liu, Jinglong Chen, Haixin Lv, Jun Wang, and Xinwei Zhang. 2022. Unsupervised multimodal anomaly detection with missing sources for liquid rocket engine. *TNNLS* (2022).

[62] Bob Ferrell and Steven Santuro. 2005. NASA Shuttle Valve Data. (Feb. 2005). http://www.cs.fit.edu/~pkc/nasa/data/

[63] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. 2016. Multivariate industrial time series with cyber-attack simulation: Fault detection using an LSTM-based predictive data model. *arXiv preprint arXiv:1612.06676* (2016).

[64] A. Garg, W. Zhang, J. Samaran, R. Savitha, and C. S. Foo. 2022. An evaluation of anomaly detection and diagnosis in multivariate time series. *TNNLS* 33, 6 (2022), 2508–2517.

[65] Dileep George. 2008. *How the Brain Might Work: A Hierarchical and Temporal Model for Learning and Recognition*. Stanford University.

[66] Jonathan Goh, Sridhar Adepu, Marcus Tan, and Zi Shan Lee. 2017. Anomaly detection in cyber physical systems using recurrent neural networks. In *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE'17)*. IEEE, 140–145.

[67] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. 2000. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. (June 2000), E215–20 pages.

[68] Abbas Golestani and Robin Gras. 2014. Can we predict the unpredictable? *Scientific Reports* 4, 1 (2014), 1–6.

[69] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *NeurIPS* 27 (2014).

[70] Adam Goodge, Bryan Hooi, See-Kiong Ng, and Wee Siong Ng. 2020. Robustness of autoencoders for anomaly detection under adversarial impact. In *IJCAI*. 1244–1250.

[71] Scott David Greenwald, Ramesh S. Patil, and Roger G. Mark. 1990. *Improved Detection and Classification of Arrhythmias in Noise-corrupted Electrocardiograms using Contextual Information*. IEEE.

[72] Frank E. Grubbs. 1969. Procedures for detecting outlying observations in samples. *Technometrics* 11, 1 (1969), 1–21.

[73] Antonio Gulli and Sujit Pal. 2017. *Deep Learning with Keras*. Packt Publishing Ltd.

[74] Yifan Guo, Weixian Liao, Qianlong Wang, Lixing Yu, Tianxi Ji, and Pan Li. 2018. Multidimensional time series anomaly detection: A GRU-based Gaussian mixture variational autoencoder approach. In *Asian Conference on Machine Learning*. PMLR, 97–112.

[75] James Douglas Hamilton. 2020. *Time Series Analysis*. Princeton University Press.

[76] Siho Han and Simon S. Woo. 2022. Learning sparse latent graph representations for anomaly detection in multivariate time series. In *KDD*. 2977–2986.

[77] Douglas M. Hawkins. 1980. *Identification of Outliers*. Vol. 11. Springer.

[78] Yangdong He and Jiabao Zhao. 2019. Temporal convolutional networks for anomaly detection in time series. In *Journal of Physics: Conference Series*, Vol. 1213. IOP Publishing, 042050.

[79] Zilong He, Pengfei Chen, Xiaoyun Li, Yongfeng Wang, Guangba Yu, Cailin Chen, Xinrui Li, and Zibin Zheng. 2020. A spatiotemporal deep learning approach for unsupervised anomaly detection in cloud systems. *TNNLS* (2020).

[80] Michiel Hermans and Benjamin Schrauwen. 2013. Training and analysing deep recurrent neural networks. *NeurIPS* 26 (2013).

[81] Geoffrey E. Hinton and Ruslan R. Salakhutdinov. 2006. Reducing the dimensionality of data with neural networks. *Science* 313, 5786 (2006), 504–507.

[82] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[83] Xiaodi Hou and Liqing Zhang. 2007. Saliency detection: A spectral residual approach. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

[84] Ruei-Jie Hsieh, Jerry Chou, and Chih-Hsiang Ho. 2019. Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing. In *IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA'19)*. IEEE, 90–97.

[85] Chia-Yu Hsu and Wei-Chen Liu. 2021. Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. *Journal of Intelligent Manufacturing* 32, 3 (2021), 823–836.

[86] Chao Huang, Chuxu Zhang, Peng Dai, and Liefeng Bo. 2021. Cross-interaction hierarchical attention networks for urban anomaly prediction. In *IJCAI*. 4359–4365.

[87] Ling Huang, Xing-Xing Liu, Shu-Qiang Huang, Chang-Dong Wang, Wei Tu, Jia-Meng Xie, Shuai Tang, and Wendi Xie. 2021. Temporal hierarchical graph attention network for traffic prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)* 12, 6 (2021), 1–21.

[88] Ling Huang, XuanLong Nguyen, Minos Garofalakis, Michael Jordan, Anthony Joseph, and Nina Taft. 2006. In-network PCA and anomaly detection. *NeurIPS* 19 (2006).

[89] Tao Huang, Pengfei Chen, and Ruipeng Li. 2022. A semi-supervised VAE based active anomaly detection framework in multivariate time series for online systems. In *WWW*. 1797–1806.

[90] Xin Huang, Jangsoo Lee, Young-Woo Kwon, and Chul-Ho Lee. 2020. CrowdQuake: A networked system of low-cost sensors for earthquake detection via deep learning. In *KDD*. 3261–3271.

[91] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. 2022. Local evaluation of time series anomaly detection algorithms. In *KDD*. 635–645.

[92] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. 2018. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *KDD*. 387–395.

[93] Yahoo Inc. 2021. S5-A Labeled Anomaly Detection Dataset, Version 1.0. (Aug. 2021). https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70

[94] Vincent Jacob, Fei Song, Arnaud Stiegler, Bijan Rad, Yanlei Diao, and Nesime Tatbul. 2021. Exathlon: A benchmark for explainable anomaly detection over time series. *VLDB* (2021).

[95] Herbert Jaeger. 2007. Echo state network. *Scholarpedia* 2, 9 (2007), 2330.

[96] Ahmad Javaid, Quamar Niyaz, Weiqing Sun, and Mansoor Alam. 2016. A deep learning approach for network intrusion detection system. In *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS'16)*. 21–26.

[97] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. 2022. Transformers in vision: A survey. *CSUR* 54, 10s (2022), 1–41.

[98] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S. Jensen. 2019. Outlier detection for time series with recurrent autoencoder ensembles. In *IJCAI*. 2725–2732.

[99] Dohyung Kim, Hyochang Yang, Minki Chung, Sungzoon Cho, Huijung Kim, Minhee Kim, Kyungwon Kim, and Eunseok Kim. 2018. Squeezed convolutional variational autoencoder for unsupervised anomaly detection in edge device Industrial Internet of Things. In *2018 International Conference on Information and Computer Technologies (ICICT'18)*. IEEE, 67–71.

[100] Eunji Kim, Sungzoon Cho, Byeongeon Lee, and Myoungsu Cho. 2019. Fault detection and diagnosis using self-attentive convolutional neural networks for variable-length sensor data in semiconductor manufacturing. *IEEE Transactions on Semiconductor Manufacturing* 32, 3 (2019), 302–309.

[101] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. 2022. Towards a rigorous evaluation of time-series anomaly detection. In *AAAI*, Vol. 36. 7194–7201.

[102] Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational Bayes. *Stat.* 1050 (2014), 1.

[103] Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

[104] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. 2017. On convergence and stability of GANs. *arXiv preprint arXiv:1705.07215* (2017).

[105] Mark A. Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal* 37, 2 (1991), 233–243.

[106] Kwei-Herng Lai, Lan Wang, Huiyuan Chen, Kaixiong Zhou, Fei Wang, Hao Yang, and Xia Hu. 2023. Context-aware domain adaptation for time series anomaly detection. In *SDM*. SIAM, 676–684.

[107] Kwei-Herng Lai, Daochen Zha, Junjie Xu, Yue Zhao, Guanchu Wang, and Xia Hu. 2021. Revisiting time series outlier detection: Definitions and benchmarks. In *NeurIPS*.

[108] Siddique Latif, Muhammad Usman, Rajib Rana, and Junaid Qadir. 2018. Phonocardiographic sensing using deep learning for abnormal heartbeat detection. *IEEE Sensors Journal* 18, 22 (2018), 9393–9400.

[109] Alexander Lavin and Subutai Ahmad. 2015. Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark. In *ICMLA*. IEEE, 38–44.

[110] Tae Jun Lee, Justin Gottschlich, Nesime Tatbul, Eric Metcalf, and Stan Zdonik. 2018. Greenhouse: A zero-positive machine learning system for time-series anomaly detection. *arXiv preprint arXiv:1801.03168* (2018).

[111] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. 2019. MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks. In *ICANN*. Springer, 703–716.

[112] Longyuan Li, Junchi Yan, Haiyang Wang, and Yaohui Jin. 2020. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *TNNLS* 32, 3 (2020), 1177–1191.

[113] Longyuan Li, Junchi Yan, Qingsong Wen, Yaohui Jin, and Xiaokang Yang. 2022. Learning robust deep state space for unsupervised anomaly detection in contaminated time-series. *TKDE* (2022).

[114] Yifan Li, Xiaoyan Peng, Jia Zhang, Zhiyong Li, and Ming Wen. 2021. DCT-GAN: Dilated convolutional transformer-based GAN for time series anomaly detection. *TKDE* (2021).

[115] Zeyan Li, Wenxiao Chen, and Dan Pei. 2018. Robust and unsupervised KPI anomaly detection based on conditional variational autoencoder. In *IPCCC*. IEEE, 1–9.

[116] Zhang Li, Bian Xia, and Mei Dong-Cheng. 2001. Gamma-ray light curve and phase-resolved spectra from Geminga pulsar. *Chinese Physics* 10, 7 (2001), 662.

[117] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. 2021. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *KDD*. 3220–3230.

[118] Fan Liu, Xingshe Zhou, Jinli Cao, Zhu Wang, Tianben Wang, Hua Wang, and Yanchun Zhang. 2020. Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional LSTM-CNN. *TKDE* (2020).

[119] Jianwei Liu, Hongwei Zhu, Yongxia Liu, Haobo Wu, Yunsheng Lan, and Xinyu Zhang. 2019. Anomaly detection for time series using temporal convolutional networks and Gaussian mixture model. In *Journal of Physics: Conference Series*, Vol. 1187. IOP Publishing, 042111.

[120] Manuel Lopez-Martin, Angel Nevado, and Belen Carro. 2020. Detection of early stages of Alzheimer's disease based on MEG activity with a randomized convolutional neural network. *Artificial Intelligence in Medicine* 107 (2020), 101924.

[121] Zhilong Lu, Weifeng Lv, Zhipu Xie, Bowen Du, Guixi Xiong, Leilei Sun, and Haiquan Wang. 2022. Graph sequence neural network with an attention mechanism for traffic speed prediction. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 2 (2022), 1–24.

[122] Tie Luo and Sai G. Nagarajan. 2018. Distributed anomaly detection using autoencoder neural networks in WSN for IoT. In *ICC*. IEEE, 1–6.

[123] Lyft. 2022. Citi Bike Trip Histories. (2022). https://ride.citibikenyc.com/system-data

[124] Junshui Ma and Simon Perkins. 2003. Online novelty detection on temporal sequences. In *KDD*. 613–618.

[125] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. 2016. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arXiv:1607.00148* (2016).

[126] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. 2015. Long short term memory networks for anomaly detection in time series. In *ESANN*. 89–94.

[127] Behrooz Mamandipoor, Mahshid Majd, Seyedmostafa Sheikhalishahi, Claudio Modena, and Venet Osmani. 2020. Monitoring and detecting faults in wastewater treatment plants using deep learning. *Environmental Monitoring and Assessment* 192, 2 (2020), 1–12.

[128] Mohammad M. Masud, Qing Chen, Latifur Khan, Charu Aggarwal, Jing Gao, Jiawei Han, and Bhavani Thuraisingham. 2010. Addressing concept-evolution in concept-drifting data streams. In *ICDM*. IEEE, 929–934.

[129] Aditya P. Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 International Workshop on Cyber-Physical Systems for Smart Water Networks (CySWater'16)*. IEEE, 31–36.

[130] Hengyu Meng, Yuxuan Zhang, Yuanxiang Li, and Honghua Zhao. 2019. Spacecraft anomaly detection via transformer reconstruction error. In *International Conference on Aerospace System Science and Engineering*. Springer, 351–362.

[131] George B. Moody and Roger G. Mark. 2001. The impact of the MIT-BIH arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine* 20, 3 (2001), 45–50.

[132] Steffen Moritz, Frederik Rehbach, Sowmya Chandrasekaran, Margarita Rebolledo, and Thomas Bartz-Beielstein. 2018. GECCO Industrial Challenge 2018 Dataset: A Water Quality Dataset for the 'Internet of Things: Online Anomaly Detection for Drinking Water Quality' Competition at the Genetic and Evolutionary Computation Conference 2018, Kyoto, Japan. (Feb. 2018). https://doi.org/10.5281/zenodo.3884398

[133] Masud Moshtaghi, James C. Bezdek, Christopher Leckie, Shanika Karunasekera, and Marimuthu Palaniswami. 2014. Evolving fuzzy rules for anomaly detection in data streams. *IEEE Transactions on Fuzzy Systems* 23, 3 (2014), 688–700.

[134] Meinard Müller. 2007. Dynamic time warping. *Information Retrieval for Music and Motion* (2007), 69–84.

[135] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. 2018. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *IEEE Access* 7 (2018), 1991–2005.

[136] Youngeun Nam, Susik Yoon, Yooju Shin, Minyoung Bae, Hwanjun Song, Jae-Gil Lee, and Byung Suk Lee. 2024. Breaking the time-frequency granularity discrepancy in time-series anomaly detection. In *WWW*. 4204–4215.

[137] Andrew Ng. 2011. Sparse autoencoder. *CS294A Lecture Notes* 72, 2011 (2011), 1–19.

[138] Zijian Niu, Ke Yu, and Xiaofei Wu. 2020. LSTM-based VAE-GAN for time-series anomaly detection. *Sensors* 20, 13 (2020), 3738.

[139] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. 2015. Learning deconvolution network for semantic segmentation. In *ICCV*. 1520–1528.

[140] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton van den Hengel. 2021. Deep learning for anomaly detection: A review. *CSUR* 54, 2 (2021), 1–38.

[141] Guansong Pang, Chunhua Shen, and Anton van den Hengel. 2019. Deep anomaly detection with deviation networks. In *KDD*. 353–362.

[142] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J. Franklin. 2022. Volume under the surface: A new accuracy evaluation measure for time-series anomaly detection. *VLDB* 15, 11 (2022), 2774–2787.

[143] Daehyung Park, Yuuna Hoshi, and Charles C. Kemp. 2018. A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1544–1551.

[144] Thibaut Perol, Michaël Gharbi, and Marine Denolle. 2018. Convolutional neural network for earthquake detection and location. *Science Advances* 4, 2 (2018), e1700578.

[145] Tie Qiu, Ruixuan Qiao, and Dapeng Oliver Wu. 2017. EABS: An event-aware backpressure scheduling scheme for emergency Internet of Things. *IEEE Transactions on Mobile Computing* 17, 1 (2017), 72–84.

[146] Faraz Rasheed and Reda Alhajj. 2013. A framework for periodic outlier pattern detection in time-series sequences. *IEEE Transactions on Cybernetics* 44, 5 (2013), 569–582.

[147] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, and Qi Zhang. 2019. Time-series anomaly detection service at Microsoft. In *KDD*. 3009–3017.

[148] Jonathan Rubin, Rui Abreu, Anurag Ganguli, Saigopal Nelaturi, Ion Matei, and Kumar Sricharan. 2017. Recognizing abnormal heart sounds using deep learning. In *IJCAI*.

[149] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *ICML*. PMLR, 4393–4402.

[150] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly detection using autoencoders with nonlinear dimensionality reduction. In *Workshop on Machine Learning for Sensory Data Analysis*. 4–11.

[151] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE Transactions on Neural Networks* 20, 1 (2008), 61–80.

[152] Udo Schlegel, Hiba Arnout, Mennatallah El-Assady, Daniela Oelke, and Daniel A. Keim. 2019. Towards a rigorous evaluation of XAI methods on time series. In *ICCVW*. IEEE, 4197–4201.

[153] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. 2022. Anomaly detection in time series: A comprehensive evaluation. *VLDB* 15, 9 (2022), 1779–1797.

[154] Pump sensor data. 2018. Pump Sensor Data for Predictive Maintenance. (2018). https://www.kaggle.com/datasets/nphantawee/pump-sensor-data

[155] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. 2018. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp* 1 (2018), 108–116.

[156] Lifeng Shen, Zhuocong Li, and James Kwok. 2020. Timeseries anomaly detection using temporal hierarchical one-class network. *NeurIPS* 33 (2020), 13016–13026.

[157] Nathan Shone, Tran Nguyen Ngoc, Vu Dinh Phai, and Qi Shi. 2018. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerging Topics in Computational Intelligence* 2, 1 (2018), 41–50.

[158] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. 2017. Anomaly detection in streams with extreme value theory. In *KDD*. 1067–1075.

[159] Maximilian Sölch, Justin Bayer, Marvin Ludersdorfer, and Patrick van der Smagt. 2016. Variational inference for on-line anomaly detection in high-dimensional time series. *arXiv preprint arXiv:1602.07109* (2016).

[160] Huan Song, Deepta Rajan, Jayaraman Thiagarajan, and Andreas Spanias. 2018. Attend and diagnose: Clinical time series analysis using attention models. In *AAAI*, Vol. 32.

[161] Xiaomin Song, Qingsong Wen, Yan Li, and Liang Sun. 2022. Robust time series dissimilarity measure for outlier detection and periodicity detection. In *CIKM*. 4510–4514.

[162] Yanjue Song and Suzhen Li. 2021. Gas leak detection in galvanised steel pipe with internal flow noise using convolutional neural network. *Process Safety and Environmental Protection* 146 (2021), 736–744.

[163] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. 2019. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *KDD*. 2828–2837.

[164] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. 2009. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. IEEE, 1–6.

[165] David M. J. Tax and Robert P. W. Duin. 2004. Support vector data description. *Machine Learning* 54 (2004), 45–66.

[166] NYC Taxi and Limousine Commission. 2022. TLC Trip Record Data. (2022). https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[167] Ahmed Tealab. 2018. Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal* 3, 2 (2018), 334–340.

[168] M. G. Terzano, L. Parrino, A. Sherieri, R. Chervin, S. Chokroverty, C. Guilleminault, M. Hirshkowitz, M. Mahowald, H. Moldofsky, A. Rosa, R. Thomas, and A. Walters. 2001. Atlas, rules, and recording techniques for the scoring of cyclic alternating pattern (CAP) in human sleep. *Sleep Med.* 2, 6 (Nov. 2001), 537–553.

[169] Markus Thill, Wolfgang Konen, and Thomas Bäck. 2020. Time series encodings with temporal convolutional networks. In *International Conference on Bioinspired Methods and Their Applications*. Springer, 161–173.

[170] Markus Thill, Wolfgang Konen, and Thomas Bäck. 2020. MarkusThill/MGAB: The Mackey-Glass Anomaly Benchmark. (April 2020). https://doi.org/10.5281/zenodo.3760086

[171] Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. 2022. TranAD: Deep transformer networks for anomaly detection in multivariate time series data. *VLDB* 15 (2022), 1201–1214.

[172] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *NeurIPS* 30 (2017).

[173] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *Stat.* 1050 (2018), 4.

[174] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *ICML*. 1096–1103.

[175] Alexander von Birgelen and Oliver Niggemann. 2018. Anomaly detection and localization for cyber-physical production systems with self-organizing maps. In *Improve-Innovative Modelling Approaches for Production Systems to Raise Validatable Efficiency*. Springer Vieweg, Berlin,, 55–71.

[176] Kai Wang, Youjin Zhao, Qingyu Xiong, Min Fan, Guotan Sun, Longkun Ma, and Tong Liu. 2016. Research on healthy anomaly detection model based on deep learning from multiple time-series physiological signals. *Scientific Programming* 2016 (2016).

[177] Xixuan Wang, Dechang Pi, Xiangyan Zhang, Hao Liu, and Chang Guo. 2022. Variational transformer-based anomaly detection approach for multivariate time series. *Measurement* 191 (2022), 110791.

[178] Yi Wang, Linsheng Han, Wei Liu, Shujia Yang, and Yanbo Gao. 2019. Study on wavelet neural network based anomaly detection in ocean observing data series. *Ocean Engineering* 186 (2019), 106129.

[179] Politechnika Warszawska. 2020. Damadics Benchmark Website. (2020). https://iair.mchtr.pw.edu.pl/Damadics

[180] Tailai Wen and Roy Keyes. 2019. Time series anomaly detection using convolutional neural networks and transfer learning. *arXiv preprint arXiv:1905.13628* (2019).

[181] Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. 2023. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *ICLR*.

[182] Jia Wu, Weiru Zeng, and Fei Yan. 2018. Hierarchical temporal memory method for time-series-based anomaly detection. *Neurocomputing* 273 (2018), 535–546.

[183] Wentai Wu, Ligang He, Weiwei Lin, Yi Su, Yuhua Cui, Carsten Maple, and Stephen A. Jarvis. 2020. Developing an unsupervised real-time anomaly detection scheme for time series with multi-seasonality. *TKDE* (2020).

[184] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, Jie Chen, Wang Zhaogang, and Honglin Qiao. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. In *WWW*. 187–196.

[185] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. 2021. Anomaly transformer: Time series anomaly detection with association discrepancy. In *ICLR*.

[186] Kenji Yamanishi and Jun-ichi Takeuchi. 2002. A unifying framework for detecting outliers and change points from non-stationary time series data. In *KDD*. 676–681.

[187] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. 2023. DCdetector: Dual attention contrastive representation learning for time series anomaly detection. In *KDD*.

[188] Chin-Chia Michael Yeh, Yan Zhu, Liudmila Ulanova, Nurjahan Begum, Yifei Ding, Hoang Anh Dau, Diego Furtado Silva, Abdullah Mueen, and Eamonn Keogh. 2016. Matrix profile I: All pairs similarity joins for time series: A unifying view that includes motifs, discords and shapelets. In *ICDM*. 1317–1322.

[189] Yue Yu, Jie Chen, Tian Gao, and Mo Yu. 2019. DAG-GNN: DAG structure learning with graph neural networks. In *ICML*. PMLR, 7154–7163.

[190] Zhihan Yue, Yujing Wang, Juanyong Duan, Tianmeng Yang, Congrui Huang, Yunhai Tong, and Bixiong Xu. 2022. TS2Vec: Towards universal representation of time series. In *AAAI*, Vol. 36. 8980–8987.

[191] Chunkai Zhang, Shaocong Li, Hongye Zhang, and Yingyang Chen. 2019. VELC: A new variational autoencoder based model for time series anomaly detection. *arXiv preprint arXiv:1907.01702* (2019).

[192] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V. Chawla. 2019. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *AAAI*, Vol. 33. 1409–1416.

[193] Mingyang Zhang, Tong Li, Hongzhi Shi, Yong Li, and Pan Hui. 2019. A decomposition approach for urban anomaly detection across spatiotemporal data. In *International Joint Conferences on Artificial Intelligence (IJCAI'19)*.

[194] Runtian Zhang and Qian Zou. 2018. Time series prediction and anomaly detection of light curve using LSTM neural network. In *Journal of Physics: Conference Series*, Vol. 1061. IOP Publishing, 012012.

[195] Weishan Zhang, Wuwu Guo, Xin Liu, Yan Liu, Jiehan Zhou, Bo Li, Qinghua Lu, and Su Yang. 2018. LSTM-based analysis of industrial IoT equipment. *IEEE Access* 6 (2018), 23551–23560.

[196] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. 2022. Self-supervised contrastive pretraining for time series via time-frequency consistency. *NeurIPS* 35 (2022), 3988–4003.

[197] Yuxin Zhang, Yiqiang Chen, Jindong Wang, and Zhiwen Pan. 2021. Unsupervised deep anomaly detection for multi-sensor time-series signals. *TKDE* (2021).

[198] Yuxin Zhang, Jindong Wang, Yiqiang Chen, Han Yu, and Tao Qin. 2022. Adaptive memory networks with self-supervised learning for unsupervised anomaly detection. *TKDE* (2022).

[199] Hang Zhao, Yujing Wang, Juanyong Duan, Congrui Huang, Defu Cao, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, and Qi Zhang. 2020. Multivariate time-series anomaly detection via graph attention network. In *ICDM*. IEEE, 841–850.

[200] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. 2019. BeatGAN: Anomalous rhythm detection using adversarially generated time series. In *IJCAI*. 4433–4439.

[201] Lingxue Zhu and Nikolay Laptev. 2017. Deep and confident prediction for time series at Uber. In *ICDMW*. IEEE, 103–110.

[202] Weiqiang Zhu and Gregory C. Beroza. 2019. PhaseNet: A deep-neural-network-based seismic arrival-time picking method. *Geophysical Journal International* 216, 1 (2019), 261–273.

[203] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding Gaussian mixture model for unsupervised anomaly detection. In *ICLR*.