Online Time-series Anomaly Detection: A Survey of Modern Model-based Approaches

Lucas Correia (lucas.correia@mercedes-benz.com)
Mercedes-Benz (Germany)
Jan-Christoph Goos
Mercedes-Benz (Germany)
Anna V. Kononova
Leiden University
Thomas Bäck
Leiden University
Philipp Klein
Mercedes-Benz (Germany)

Research Article

Keywords: Online, Anomaly Detection, Time series, Deep Learning, Survey

Posted Date: October 5th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2120136/v1

Online Time-series Anomaly Detection: A Survey of Modern Model-based Approaches

Lucas Correia[1*], Jan-Christoph Goos[1], Philipp Klein[1], Thomas Bäck[2] and Anna V. Kononova[2]

[1]Mercedes-Benz AG, Stuttgart, Germany.
[2]Leiden University, Leiden, The Netherlands.

*Corresponding author(s). E-mail(s): lucas.correia@mercedes-benz.com;

Abstract

This survey provides an extensive overview of the state-of-the-art model-based online semi-supervised and unsupervised anomaly detection algorithms used on multivariate time series. It also outlines the most popular benchmark datasets used in literature, as well as a novel taxonomy where a distinction between online and offline, and training and inference is made. To achieve this, almost 50 peer-reviewed publications are surveyed and categorised into different model families to paint a clear picture of the anomaly detection landscape for the reader. Then, where possible, a comparison of the anomaly detection performance of the surveyed approaches is provided and the key research gaps are highlighted. It is concluded that few approaches propose any enhancements that involve the user feedback. In addition to that, transformer-based models seem to have found little application in anomaly detection so far, though this is most likely due to the novelty of the model type, not necessarily due to the lack of potential. Moreover, there is no standard benchmark procedure to assess anomaly detection performance. A variety of different datasets and evaluation metrics used for evaluation are used hence it is difficult to draw conclusions from comparisons between papers. Lastly, the rarity of approaches trained and inferred in an online manner is also pointed out, possibly explained by the lower performance compared to offline trained approaches.

Keywords: Online, Anomaly Detection, Time series, Deep Learning, Survey

1 Introduction

As a result of the fourth industrial revolution, also known as industry 4.0, immense amounts of data (referred to as "Big Data") are collected from sensors mounted at different checkpoints in many processes in research and development, manufacturing and testing (Xu et al, 2014). This data can expose subtle but important trends and correlations, as well as give the data user key insights on how to optimise systems and processes, which can potentially give a company a competitive advantage in the market.

Recording high-quality data is important, as incomplete or anomalous data can negate any potential benefits that can be extracted from it. With the rise of industry 4.0, anomaly detection has therefore gained relevance over the past decade, with the bar being set ever higher as data becomes more and more high dimensional (Zimek et al, 2012; Erfani et al, 2016). Time-series data has especially gained attention as it can exhibit temporal causality and hence represent dynamic time-variant processes, which can give additional insights but also is more complex than stationary data, due to its temporal nature. Thus, reliably finding anomalies in high-dimensional time series has been a very active research area and hence this survey focuses on that.

In addition to that, this survey lays special focus on approaches related to online applicability, i.e. the ability to perform inference while the data is being recorded and streamed. This is especially useful in industry where finding anomalies in a timely manner can help reduce cost and increase operating efficiency.

Deep learning has also been a very active research area, owing to increasing computing power but also the availability of data. It has also been applied to anomaly detection, especially in high dimensional data, which is where traditional approaches have started to reach their limits (Chalapathy and Chawla, 2019).

Also, large amounts of labelled data are often not available, hence the survey does not delve into supervised anomaly detection but focuses instead on semi-supervised and unsupervised methods.

Having laid out the key focus points, the survey is structured as follows: first, work related to this publication (Section 2) is presented. This includes surveys that specialise in time-series anomaly detection and feature, at least partially, content on online detection or in streaming data. Then, a novel taxonomy (Section 3) is defined, including anomaly types, approaches to anomaly detection and the various cases that are encompassed in the online anomaly detection domain. Following that, a small section presents publicly available datasets (Section 4) used in the literature to evaluate approaches. The four sections after that (Sections 5 - 8) discuss the approaches in literature used to detect anomalies in multivariate time series. The papers have been grouped by architectural similarity and sorted by publishing date. The following section (Section 9) is dedicated to a comparison between models for a given dataset, where this is possible. Then, Section 10 presents a discussion around the surveyed model types, as well as an assessment of what research gaps exist and what areas deserve to be further looked into. Finally, the conclusion (Section 11) summarises the current state of the time-series anomaly detection landscape, while also concisely highlighting current problems and research gaps. For a visual representation of the structure of the survey, please refer to Figure 1.

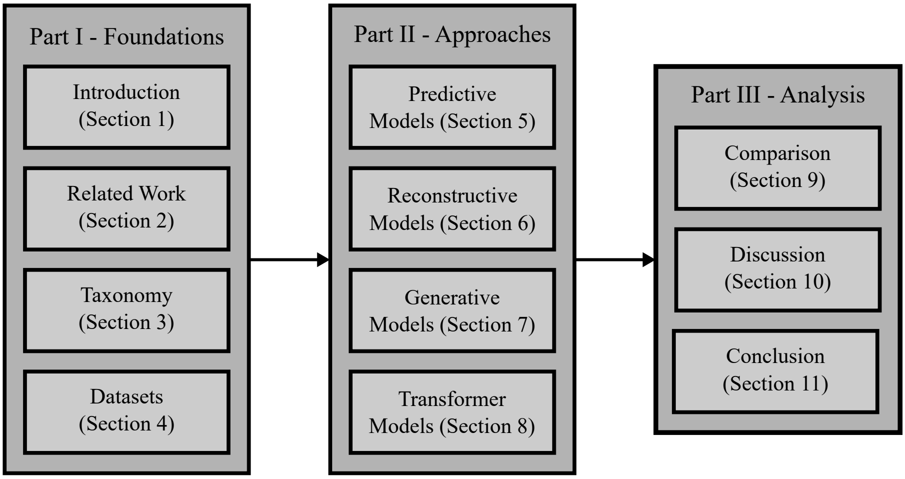

Fig. 1: A visual representation of the structure of the survey.

2 Related Work

Various anomaly detection surveys with a focus on time series and online functionality have been published over the years. A summary of the surveys is given in Table 1. The list of surveys have been found using the keywords *anomaly detection*, *survey*, *time series*, *online*, *real-time* and *streaming data* using a variety of libraries, such as IEEE Xplore, ScienceDirect, Springer RD and the ACM Digital Library, in the time between 2000 and 2022, which also applies to the publications surveyed. Up until recently, surveys published on online time-series anomaly detection only discussed conventional methods, which for the purpose of this survey will denote methods not based on deep learning.

In 2012, Chandola et al (2012) published a survey focusing on anomaly detection in discrete sequences, but also consider how the techniques can be applied to continuous-sequence anomaly detection, such as time series. These techniques have been classified into sequence-based, contiguous subsequence-based and pattern frequency-based anomaly detection approaches. Furthermore, they dedicate a chapter to online anomaly detection where the online applicability of the methods mentioned above is commented on.

Gupta et al (2014) strictly focus on continuous sequential data and delves further into the area of streaming data, where three types of approaches are presented by the authors: online learning prediction models, distance-based approaches and models for high-dimensional data streams.

In addition to that, Aggarwal (2017) also discusses time-series anomaly detection in an online setting. Two anomaly types identified in time series by the author are contextual and collective anomalies. The former can be detected using prediction models, whereas the latter can be found using transformations, as well as linear, probabilistic or distance-based models.

Mason et al (2019) published a survey on what is considered traditional methods, i.e. not based on deep learning (DL). These methods are categorised into four domains: statistics, time-series analysis, pattern mining and machine learning. They proceed to further classify each of the approaches into parametric and non-parametric. In addition, they also comment on the online applicability of the methods and provide an overview of machine learning platforms.

DL has been gaining more traction in the last few years, due its scalability and good performance when presented with larger amounts of data but also reduced need for feature engineering (Chalapathy and Chawla, 2019).

The first paper that compares traditional methods with DL-driven anomaly detection for online applications was published by Munir et al (2019a) where the authors not only survey several methods from both paradigms but also provide a quantitative evaluation and comparison between the presented models for two different publicly available data sets. They conclude that despite not being the fastest approach, DL provides the best performance for the chosen performance metrics.

Blázquez-García et al (2021) published a survey encompassing both DL and classical methods. They provide a novel taxonomy which deviates slightly from the *point, collective and contextual anomaly* convention, first suggested by Chandola et al (2009). Here the authors suggest that the point and contextual outliers in the previous convention can be considered global and local point anomalies, respectively. Subsequence outliers are synonymous with collective outliers, however, another type of anomaly is suggested: the outlier time series, where an entire input sequence is considered anomalous. Furthermore, they discuss the idea of applying univariate techniques to multivariate data, which had not been discussed in the previous publications.

Aimed at video anomaly detection, Nayak et al (2021) published a survey which reviews the current DL-based methods. Unlike multivariate sensor data, which can be seen as a series of vectors along a time axis, video is a series of matrices along a time axis and hence is also considered a time series. They discuss a wide variety of DL models that can be applied to anomaly detection and also comment on the feasibility of the respective models in an online setting.

Several gaps have been identified in the current survey landscape. Firstly, when speaking of online anomaly detection it is not explicitly distinguished between online training and online inference stages. In addition to that, none of the surveys provides an overview of the most popular datasets used for benchmarking anomaly detection algorithms. Hence, this survey aims to provide the following contributions:

- Novel taxonomy in the area of online anomaly detection, making a distinction between online training and online inference
- Overview of the most popular benchmark datasets used for evaluating time-series anomaly detection
- Updated overview on the state-of-the-art methodology used for model-based online anomaly detection

Table 1: Summary of surveys discussing online anomaly detection in time series. Key: SAD: Supervised anomaly detection, SSAD: Semi-supervised anomaly detection, USAD: Unsupervised anomaly detection, Deep: Deep Learning Approaches, Trad: Traditional Approaches

Authors	SAD	SSAD	USAD	Deep	Trad
Chandola et al (2012)		✓	✓		✓
Gupta et al (2014)	✓		✓		✓
Aggarwal (2017)	✓	✓	✓		✓
Mason et al (2019)	✓	✓	✓		✓
Munir et al (2019a)	✓	✓	✓	✓	✓
Blázquez-García et al (2021)			✓	✓	✓
Nayak et al (2021)	✓	✓	✓	✓	

3 Taxonomy of Anomaly Detection

3.1 Anomaly Types and Detection

Anomalies in time series can be assumed to have different shapes and forms. A commonly used and accepted definition (Hawkins, 1980, p. 1) reads as follows:

> *"An observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism."*

Over the years several taxonomies have been proposed to better classify different anomalies. This work mostly follows the taxonomy suggested by Blázquez-García et al (2021), where anomalies are classified into point, sub-sequence (also known as collective in literature) and time-series anomalies.

A *point anomaly* is defined as a value at a time step that does not conform to the typical behaviour of a system. In a multivariate sequence $\mathbf{X} \in \mathbb{R}^{T \times c}$ of length T and c channels, an anomaly event $\mathbf{A} \in \mathbb{R}^{S \times d}$ of length S that is detected in d channels, where $d \leq c$, is considered a *point anomaly* when $S = 1$. An example of a *point anomaly* is illustrated in Figure 2. While more easily detectable than the other types, these anomalies are rare events in the real world as systems affected cannot usually return to a normal state before the next time step arrives unless the sampling rate is very low.

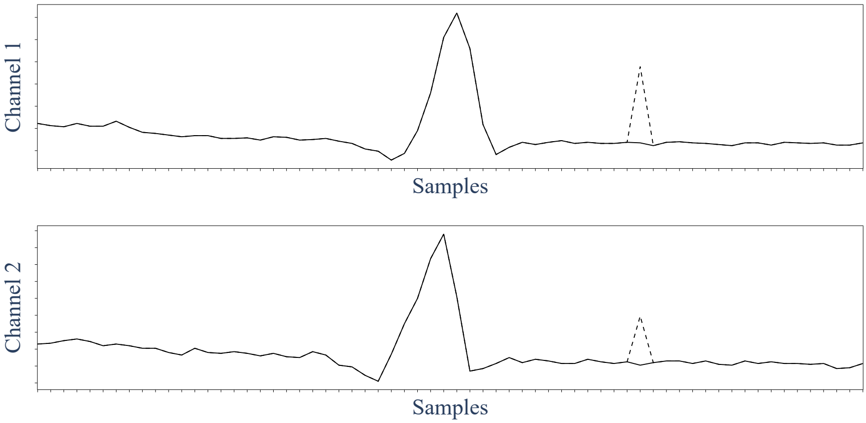

Fig. 2: Example of *point anomaly* in both channels. The continuous line represents the ground truth time series and the traced line the anomaly. The time series shown is a heart beat from the MIT-BIH Arrhythmia Database (Moody and Mark, 2001)

Sub-sequence anomalies are defined by a series of anomalous data points, i.e. a sub-sequence of points that do not reflect the normal behaviour of a system. In a multivariate sequence $\mathbf{X} \in \mathbb{R}^{T\times c}$ of length T and c channels, an anomaly event $\mathbf{A} \in \mathbb{R}^{S\times d}$ of length S that is detected in d channels, where $d \leq c$, is considered a *sub-sequence anomaly* when $1 < S < T$. In a real-world system, this type of anomaly may occur when a component in the said system runs at reduced functionality or fails but manages to recover to a normal state after a period of time. An example of a *sub-sequence anomaly* is illustrated in Figure 3.

A *time-series anomaly* can be seen as a long sub-sequence that has the same length as the sequence. Hence, in a multivariate sequence $\mathbf{X} \in \mathbb{R}^{T\times c}$ of length T and c channels, an anomaly event $\mathbf{A} \in \mathbb{R}^{S\times d}$ of length S that is detected in d channels, where $d \leq c$, is considered a *time-series anomaly* when $S = T$. This type of anomaly can occur when an initial parameter or state is anomalous, leading to all observations in the sequence being anomalous as well. An example of a *time-series anomaly* is illustrated in Figure 4.

3.2 Approaches for Anomaly Detection

Anomaly detection can be performed in three different ways: supervised, semi-supervised and unsupervised. When all available data is labelled, supervised anomaly detection is an attractive solution to find outliers. However, this is seldom the case, as labelled data is scarce because manually labelling data is laborious and often not an option due to the amount of data (Munir et al,

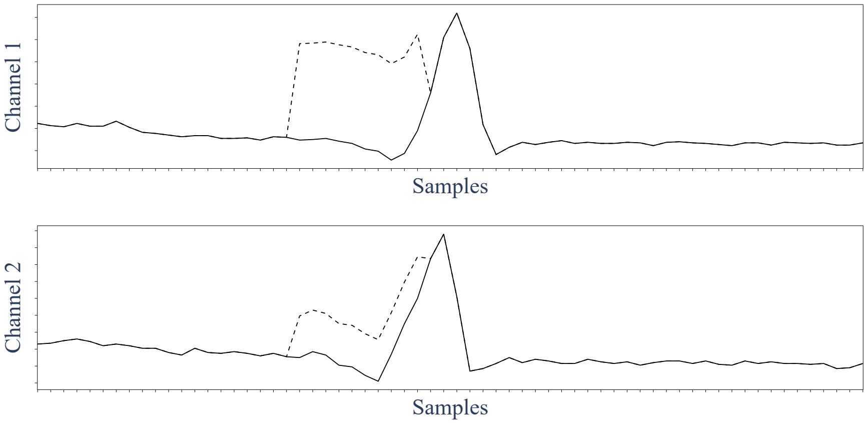

Fig. 3: Example of *sub-sequence anomaly* in both channels. The continuous line represents the ground truth time series and the traced line the anomaly. The time series shown is a heart beat from the MIT-BIH Arrhythmia Database (Moody and Mark, 2001)

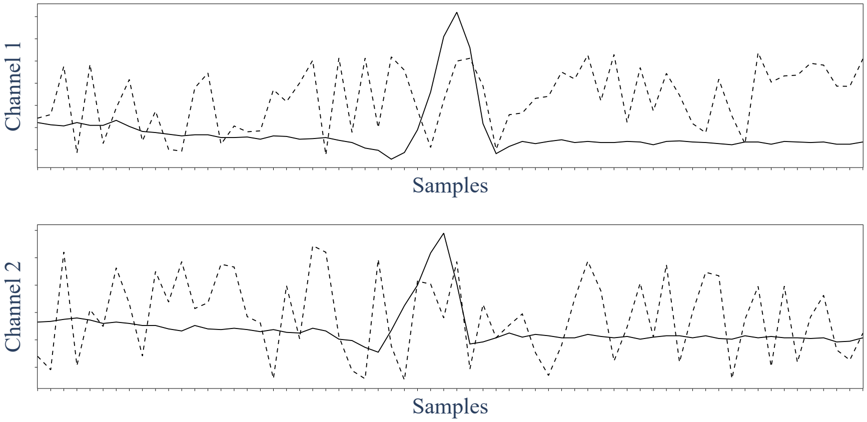

Fig. 4: Example of *time-series anomaly* in both channels. The continuous line represents the ground truth time series and the traced line the anomaly. The time series shown is a heart beat from the MIT-BIH Arrhythmia Database (Moody and Mark, 2001)

2019b). This limits the applicability of supervised methods for anomaly detection, hence this survey does not delve into that branch in the literature. On the other hand, semi-supervised and unsupervised anomaly detection can be applied when little or no labelled data is available, and therefore are more attractive solutions. A graphic classification of the model-based anomaly detection landscape is shown in Figure 5. Supervised anomaly detection is shown as a traced box for completeness, although it is not discussed in this survey.

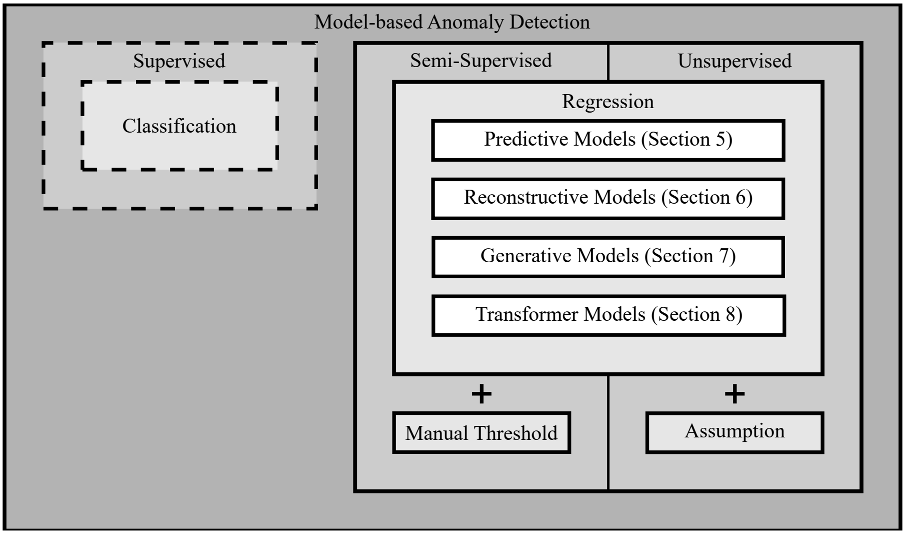
Model-based Anomaly Detection
Supervised
Classification
Semi-Supervised
Unsupervised
Regression
Predictive Models (Section 5)
Reconstructive Models (Section 6)
Generative Models (Section 7)
Transformer Models (Section 8)
+
+
Manual Threshold
Assumption

Fig. 5: Model-based anomaly detection landscape

In both semi-supervised and unsupervised anomaly detection the goal is to create a data-driven representation of the system, hence we speak of model-based anomaly detection. Often, this is done using regression, with one exception, namely when the discriminator of a generative adversarial network (GAN) is used (see Section 7). From literature, four model families have been identified: predictive models (Section 5), reconstructive models (Section 6), generative models (Section 7) and transformer models (Section 8).

The main difference between semi-supervised and unsupervised anomaly detection approaches is how the output from the model is interpreted. In semi-supervised anomaly detection, a small amount of labelled data is available, hence this can be used to improve detection. For example, labelled data can be used to set an error threshold or to create a receiver operating characteristic (ROC) curve to find the ideal balance between precision and recall.

Unsupervised anomaly detection on the other hand makes use of assumptions to detect anomalies. For instance, for a given error sequence, i.e. error between observation and computed output, it can be assumed that in normal samples the error does not deviate more than a certain number of standard deviations from the mean if it is normally distributed. Another example could be calculating the reconstruction probability that a sequence is output given the latent distribution in a variational autoencoder (VAE).

3.3 Online Anomaly Detection

Online anomaly detection, also known as anomaly detection in streaming data, is defined as the process of detecting anomalies within data as it arrives, meaning that only information up to the present is observed. This eliminates methods that rely on an entire sequence to flag anomalies, because a decision cannot be made as data arrives and has to be back-dated (Thill, 2022).

Real-time anomaly detection can be considered a more strict subclass within online anomaly detection as it requires computation time to be equally long or shorter than the intervals at which data arrives.

Within model-based online anomaly detection, three distinctive fields have been identified. The way they relate to each other is visualised in Table 2.

Table 2: The three anomaly detection fields discussed in this survey shown in the style of a confusion matrix.

		Training	
		Offline	Online
Inference	Offline	✓	✓
	Online		✓

One approach to online anomaly detection involves models that not only detect anomalies in real-time but also learn during application, meaning model parameters are adjusted during inference. This eliminates the need for prior training, however, it also means that the models may need a few runs until they can provide adequate anomaly detection performance.

However, online anomaly detection can still be performed when the model is trained prior to inference, in other words, it is trained offline but used for inference online.

Extensive work has also been done in the field of offline anomaly detection. While the publications cited in the following sections have not explicitly been applied to or focused on online inference capability of their models, it is believed that the approaches taken could still be applied in such setting if the method fulfils the above-mentioned requirements for online anomaly detection.

4 Datasets

Anomaly detection literature has used a variety of time-series datasets over the years. Publicly available datasets play an important role in enabling researchers to benchmark their methods against the state of the art. To provide the reader with an idea of what the most used datasets are, the key information has been summarised in Table 3. To qualify, the datasets must have been released between 2000 and 2022 as well as found use in at least two publications. The oldest multivariate dataset is the MIT-BIH Arrhythmia Database (Moody and Mark, 2001), which features two-channel ECG time-series data. Its data represents 48 30-minute records with a wide range of different anomaly types labelled by domain experts. The Singapore University of Technology and Design published two datasets: the Secure Water Treatment (SWaT) (Mathur and Tippenhauer, 2016) and the Water Distribution (WADI) (Ahmed et al, 2017) datasets. SWaT consists of network traffic data under various attack scenarios and features 51 channels. WADI is similar, also representing various attack scenarios but much higher dimensional data with 118 channels. NASA also published two commonly used datasets: the Soil Moisture Active Passive (SMAP) satellite dataset and the Mars Science Laboratory (MSL) rover dataset (Hundman et al, 2018). They are both multivariate time series data and consist of 25 and 55 channels, respectively, some of them being temperature, radiation and power signals. Another multivariate time-series dataset is the Server Machine Dataset (SMD) (Su et al, 2019) which features 38 channels.

For a well rounded evaluation, it is recommended that authors of future publications consider using all datasets, as each of them features different dimensionality and signal characteristics.

Table 3: Key information about the most popular datasets summarised. *Features* refers to the number of channels of the time-series signals in the dataset, *Time Steps* to the total number of data points, *Time Series* to the number of sequences, *Anomalies* to the number of anomalous events over all time series and *Occurrences* to the number of times the dataset is used in the publications considered in this survey.

Name	Features	Time Steps	Subsets	Anomalies	Occurrences
MIT-BIH	2	31,207,680	48	112,624	7
WADI	118	2,178,744	2	27	4
SWaT	51	944,919	1	35	8
SMAP	25	562,800	55	67	9
MSL	55	132,046	27	36	9
SMD	38	1,416,825	28	654	5

5 Predictive Models

Predictive models are models that have been trained to predict a time step given a window of past values. They can be used for anomaly detection by comparing their predictions with the observed/measured value using a variety of techniques to obtain an error metric. The key assumption that this type of model relies on is that normal, i.e. anomaly-free, sequences will be predicted with a small error, whereas anomalies will lead to a large error between prediction and observed values. Predictive models also tend to be the simplest and on average the oldest model family discussed in this survey. An overview of all the prediction-based modelling publications is presented in Table 4. The keys in the table have been chosen such that the reader can quickly recognise what type of approach has been taken by the respective author.

Table 4: Summary of all publications surveyed in the area of prediction-based time-series anomaly detection. The key is as follows, T: Training, I: Inference, L: LSTM, G: GRU, CL: ConvLSTM, C: CNN, D: Dense, M: Miscellaneous, Pred: Prediction, Rec: Reconstruction, AD: Anomaly Detection, SS: Semi-supervised, US: Unsupervised

	Model	Online		Architecture						AD
Authors	Class	T	I	L	G	CL	C	D	M	Type
Yen et al (2019)	Pred	✓	✓	✓			✓			SS
Que et al (2019)	Pred		✓	✓						US
Munir et al (2019b)	Pred		✓				✓			SS
Malhotra et al (2015)	Pred			✓						SS
Chauhan and Vig (2015)	Pred			✓						SS
Filonov et al (2016)	Pred			✓						SS
Filonov et al (2017)	Pred			✓						US
Goh et al (2017)	Pred			✓						SS
Thill et al (2019)	Pred			✓						SS
Hundman et al (2018)	Pred			✓						US
Tambuwal and Neagu (2021)	Pred			✓						SS
Sakuma and Matsutani (2021)	Pred								✓	US

5.1 Online Training and Online Inference

Yen et al (2019) present an anomaly detection algorithm for computer logging records, that convolutes the input log key sequence using four parallel convolutional neural network (CNN) (Fukushima, 1980; Lecun et al, 1998) layers with varying kernel sizes. The resulting feature maps, i.e. the vector representation of the CNN output, are then concatenated and fed into an long short term memory (LSTM) (Hochreiter and Schmidhuber, 1997; Gers et al, 2000) layer which then predicts a probability distribution across all log keys. The most likely log keys are considered normal, where the minimum likelihood at which a key is considered normal is the threshold. Yen et al (2019) experimented with different online training techniques, including regular retraining independently of anomaly detection result and retraining if the false-positive rate exceeds a given value.

5.2 Offline Training and Online Inference

Que et al (2019) brings forward a method to detect anomalies during flight testing of commercial aircraft. It utilises the encoder component of a trained autoencoder to reduce the dimensionality of a multi-variate signal. Following that, a pair of LSTM layers predict the next time step based on the latent vector output by the transferred encoder. The error resulting from these predictions is fit to a Gaussian distribution, then allowing for a likelihood to be estimated, hence anomalies can be detected if the likelihood exceeds a given confidence interval.

Munir et al (2019b) proposes a stacked CNN network, which predicts the following time step based on a given history window. The anomaly score is based on the Euclidean distance between the observed value and the prediction and needs to be evaluated against a manually defined threshold.

The approach used by Que et al (2019) clearly distinguishes itself from the one taken by Munir et al (2019b) by not only being based on LSTM rather than CNN layers but also by employing transfer learning.

5.3 Offline Training and Offline Inference but Online Capable

The majority of existing work makes use of, sometimes stacked, recurrent neural networks (RNN) to predict future time steps. These predictions are then held up against the observed value from which an anomaly score is calculated. The error metrics used as the anomaly score is where the approaches differ from each other. Most algorithms use a manually defined threshold in order to detect anomalies and rely on the maximum likelihood estimation (Malhotra et al, 2015; Chauhan and Vig, 2015), mean squared error (Filonov et al, 2016, 2017), cumulative sum (Goh et al, 2017) or on the Mahalanobis distance (Thill et al, 2019) for the anomaly score.

Hundman et al (2018) propose using a dynamic error threshold on the smoothed prediction error to detect anomalies in spacecraft sensor data. This is done by computing the threshold for every time step using a custom function that takes in the mean and standard deviations of the absolute reconstruction error, as well as an adjustable parameter that adjusts the sensitivity.

Rather than just predicting the next time step, Tambuwal and Neagu (2021) present an implementation of a stacked LSTM network that also outputs quantiles for that prediction. Using the predicted quantiles a confidence interval can be calculated using the difference between the upper and lower quantiles. Hence this confidence interval can be used to find anomalies, which would be indicated by a large confidence interval.

Another rather unique approach is taken by Sakuma and Matsutani (2021), where a sparse recurrent neural network, known as an echo state network (ESN), is used to make next time-step predictions from which an anomaly score similar to the z-score is calculated. Hotelling's T^2 test is used to automatically obtain a threshold.

As mentioned, a large number of publications are based on LSTM networks and only differ on how the predictions are interpreted to detect anomalies. Tambuwal and Neagu (2021) is, as far as the authors are aware, the only approach that uses confidence intervals for anomaly detection, whereas Sakuma and Matsutani (2021) are the only authors, other than Suh et al (2016), that use ESNs.

6 Reconstructive Models

Reconstructive modelling is usually based on some variation of an autoencoder. Autoencoders work by compressing a given input space into a latent vector, i.e. a reduced representation of the input space. The latent vector is then expanded again to reconstruct the input. These two processes are done by an encoder and decoder, respectively, which generally mirror each other in architecture. Reconstructive models rely on the assumption that the reconstruction error will be large when faced with anomalous data. An overview of the reconstruction-based modelling approaches discussed in this survey is presented in Table 5.

Table 5: Summary of all publications surveyed in the area of reconstruction-based time-series anomaly detection. The key is as follows, T: Training, I: Inference, L: LSTM, G: GRU, CL: ConvLSTM, C: CNN, D: Dense, M: Miscellaneous, Pred: Prediction, Rec: Reconstruction, AD: Anomaly Detection, SS: Semi-supervised, US: Unsupervised

	Model	Online		Architecture						AD
Authors	Class	T	I	L	G	CL	C	D	M	Type
Gugulothu et al (2018)	Rec		✓	✓						SS
Hsieh et al (2019)	Rec		✓	✓						SS
Bayram et al (2021)	Rec		✓			✓				SS
Malhotra et al (2016)	Rec			✓						SS
Homayouni et al (2020)	Rec			✓						SS
Lindemann et al (2020)	Rec			✓						SS
Nguyen et al (2021)	Rec			✓						US
Naito et al (2021)	Rec									SS
Kieu et al (2018)	Rec			✓			✓			SS
Kieu et al (2019)	Rec								✓	SS
Tayeh et al (2022)	Rec					✓				US
Zhang et al (2019)	Rec					✓	✓			SS
Chadha et al (2021)	Rec						✓			US
Thill et al (2021)	Rec						✓			SS
Gong et al (2021)	Rec						✓			SS
Zhang et al (2021a)	Rec						✓			US

6.1 Online Training and Online Inference

Approaches based on autoencoders that are trained and infer in an online fashion do not exist, to the best of the author's knowledge. Technically such methods are possible, presumably they would work very similarly to the approach in Subsection 5.1. It is difficult to estimate how well this hypothetical algorithm would work, though it would be research direction to pursue.

6.2 Offline Training and Online Inference

The first approach that involved offline training of an autoencoder that is capable of online inference is proposed by Gugulothu et al (2018). It is based on the work by Malhotra et al (2016) and adds a feature reduction layer between the input layer and the encoder, which acts as a regulariser in capturing dependencies between channels. The encoder and decoder are both based on LSTM layers, whose output, i.e. reconstruction, is then evaluated using the Mahalanobis distance as the anomaly score, which is held up against a threshold obtained through F_1 score maximisation to detect anomalies. The F_1 score corresponds to the harmonic mean between precision and recall.

Hsieh et al (2019) also used an LSTM-based autoencoder to detect anomalies in smart manufacturing via the mean squared error. They applied transfer learning to improve the algorithm performance significantly. This is done by pre-training the model on data from different points in the manufacturing process.

Bayram et al (2021) compared a convolutional autoencoder with a novel autoencoder architecture to detect acoustic anomalies in industrial processes. It combines convolutional layers with convolutional LSTM layers, introduced by Shi et al (2015), which relies on the convolution operation rather than matrix multiplication. The Euclidean distance is then computed from the reconstruction. The threshold is taken as the value that yields the maximum difference between the true-positive rate and the true-negative rate.

While Gugulothu et al (2018), Hsieh et al (2019) all use some version of the LSTM autoencoder they each attempt to enhance the learning process in a different way, be it through reducing the dimensionality of the input data, augmenting the training data set or removing out-of-distribution data points from the data set.

6.3 Offline Training and Offline Inference but Online Capable

Malhotra et al (2016) propose using an LSTM autoencoder which uses the Mahalanobis distance between the absolute error vector and the distribution of errors resulting from normal data. A threshold is obtained by maximising a custom version of the F score, with weighted precision over recall, using a small amount of labelled data.

Homayouni et al (2020) later proposed an LSTM autoencoder approach which is capable of ingesting both unlabelled and labelled data to improve the detection of anomalies. This is done by adding an extra label column as one of the features to be reconstructed. The label value varied depending on whether the time step is faulty, suspicious, unknown or valid. In addition to that, they also utilised a decision tree model in an attempt to provide explainability to the anomalies, by isolating the channel that most likely caused the anomaly.

Lindemann et al (2020) employed an LSTM autoencoder with discrete wavelet transforms at each end to better intake data of different frequencies. After training, the encoder is removed and the decoder is used as an inverse model for the meta model, a NARX network. The difference between the meta model input and the decoder output is then used to detect anomalies.

Nguyen et al (2021) used an LSTM autoencoder paired with a one-class support vector machine to detect anomalies. It works by separating normal and anomalous data points in the error vector resulting from the reconstruction using a hyperplane.

Naito et al (2021) used two autoencoders, one to reconstruct the time series, another one to reconstruct the reconstruction error from the previous autoencoder, which is then added to the reconstructed time series, in theory yielding a superior reconstruction.

Kieu et al (2018) compared a 2D CNN autoencoder with an LSTM autoencoder for anomaly detection in enriched time series. Enriched means the time series which has been augmented with derived and statistical features, hence the enriched time series has a larger feature space than the raw time-series. In addition to that, the autoencoder reconstruction is enhanced by embedding one-hot encoded contextual information in it.

Kieu et al (2019) also proposes using an ensemble of sparse recurrent autoencoders. The autoencoders differ from each other as they use different sparseness weight vectors. Furthermore, two types of ensembles are proposed: an independent framework, where the autoencoders are run independently and a shared framework, where the latent vector is shared between autoencoders. Anomalies are then detected using the median value from the Euclidean distance vector of all autoencoders and a manually defined threshold.

Zhang et al (2019) attempted to detect anomalies by creating a convolutional autoencoder that works with correlation matrices, here called signature matrices, rather than raw time series, where signature matrices are representations of the correlations between windows in the time series. The autoencoder is further enhanced by skip connections that connect layers in the encoder with their decoder counterparts in order to allow the model to better deal with long sequences. The skip connections also process the information flowing through them, by running it through a ConvLSTM layer paired with a unnamed custom attention layer. The algorithm can also provide a root cause analysis by labelling the channels associated with the three worst reconstructed correlations in a given matrix.

Tayeh et al (2022) proposed a similar an autoencoder structure similar to Zhang et al (2019), though without the skip connections. Here a Bahdanau-style attention mechanism (Bahdanau et al, 2015) has been added to the model to maintain performance with increasing length of input sequences. The approach can dynamically adjust its anomaly detection threshold, similar to Hundman et al (2018), which is used to label time steps, in this case correlation matrices, as anomalies. Like what was proposed by Zhang et al (2019), this algorithm can also provide a degree of explainability to its outputs, though by using a threshold to detect suspicious channels within the correlation matrix.

Chadha et al (2021) expanded on the idea to use a convolutional autoencoder for anomaly detection by adding a clustering element to the latent space. The latent space is split into a discriminative and reconstructive component which is used for clustering and reconstruction, respectively. The clustering element regularises the latent space by adding a clustering loss term to the total loss function. In addition to that, the authors propose a semi-supervised variation of the model where the trained decoder is replaced by fully connected layers that work as a classifier.

Thill et al (2021) suggested a TCN autoencoder with a series of improvements. For example, the authors suggest reversing the dilation rate ordering at the decoder, to better adapt it to the lower resolution representation of the time series in the latent space. Inspired by the ResNet (He et al, 2016) and DenseNet (Huang et al, 2017) architectures, they also implement skip connections to prevent an overreliance of the model on the high dilation rate representation. Furthermore, Thill et al. also proposed using the reduced representations between hidden layers to detect anomalies, as they represent different frequency scales due to the variable dilation rate. In addition to that, they also used map reduction layers after each hidden layer to reduce the dimensionality of the convolution and therefore the number of tunable parameters. Lastly, they also proposed running the anomaly score through a high-pass filter set to cut off frequencies lower than 1Hz to remove drifts from the signal. The anomaly score used is the Mahalanobis distance.

Gong et al (2021) proposed a different way to implement skip connections in a convolutional autoencoder. Rather than moving the information through a ConvLSTM and an attention layer at the skip connection, like Zhang et al (2019), the information is processed by a multi-layer perceptron (MLP) layer followed by a Bahdanau-style attention layer. Another innovation is the use of a prediction module at the bottleneck of the autoencoder to enhance the latent space mapping during training. The prediction module consists of an MLP which predicts the following input window from the latent space. The loss function is also customised to include both reconstruction and prediction loss, requiring a weight parameter that varies the ratio between either loss.

Zhang et al (2021a) developed an adversarial approach to train an anomaly detection network. Here the model consists of a convolutional autoencoder and a deconvolutional classifier, which, in a way, compete against each other during training. Normal samples are corrupted using a custom algorithm to represent anomalies, parameterised through a corruption level value, and are input into the autoencoder along with their uncorrupted counterpart. Once the latent vector has been computed for either sample the latent vector error is found and minimised. The goal of this is to create as plausible a reconstruction from the latent vector as possible, regardless of the input sample. This reconstruction is then used as a class to train the classifier, which is used to tell the normal, uncorrupted sample apart from the reconstruction. The loss function used for the autoencoder is composed of the reconstruction loss and the latent vector error multiplied by a pre-defined weight.

7 Generative Models

Generative models can be further segmented into two most common model types: VAEs (Kingma and Welling, 2014) and GANs (Goodfellow et al, 2014). The former bears some similarity to traditional autoencoders other than the fact that the low-dimensionality representation is not mapped to a vector but rather a distribution from which a latent variable is sampled. In the generative modelling literature, the encoder of an autoencoder is referred to as the recognition model and the decoder as the generative model. GANs, on the other hand, work by training a generator network to generate a plausible sample from a random noise input. A second network called the discriminator then tries to distinguish between the generated input and the training example. As training progresses the generator becomes better at generating samples, hence discriminating between the generated input and the real data becomes harder and harder. However, the discriminator also becomes better at telling real samples apart from generated samples. Hence, training can be thought of as a two-player min-max game (Goodfellow et al, 2014). An overview of all the generation-based modelling publications is presented in Table 6.

Table 6: Summary of all publications surveyed in the area of generation-based time-series anomaly detection. The key is as follows, T: Training, I: Inference, L: LSTM, G: GRU, CL: ConvLSTM, C: CNN, D: Dense, M: Miscellaneous, Gen: Generative, Tran: Transformer, AD: Anomaly Detection, SS: Semi-supervised, US: Unsupervised

	Model	Online		Architecture						AD
Authors	Class	T	I	L	G	CL	C	D	M	Type
Suh et al (2016)	Gen	✓	✓						✓	SS
Park et al (2018)	Gen		✓	✓						SS
Su et al (2019)	Gen		✓		✓					US
Chen et al (2020)	Gen		✓				✓			SS
Zhang et al (2021c)	Gen		✓		✓					SS
Li et al (2019)	Gen		✓	✓						SS
Guo et al (2018)	Gen				✓					US
von Schleinitz et al (2021)	Gen			✓						SS
Li et al (2021)	Gen				✓					US
Choi et al (2022)	Gen			✓						SS
Zhou et al (2019)	Gen						✓			SS
Choi et al (2020)	Gen						✓			SS
Geiger et al (2020)	Gen			✓						SS
Zhu et al (2019)	Gen			✓						SS
Sun et al (2019)	Gen						✓			US

7.1 Online Training and Online Inference

Suh et al (2016) proposed a VAE based on ESNs in an effort to adapt it for anomaly detection in time-series data. This model is trained in an online manner, i.e. the parameters are updated as new data is input. The anomaly score is given by the negative log-likelihood of an observation variable given the previous echo state, which is compared against a set threshold.

7.2 Offline Training and Online Inference

Park et al (2018) suggested using LSTM layers rather than dense layers in VAEs in order to detect anomalies in robot-assisted feeding. To regularise the VAE the authors added noise to the input. The anomaly score is calculated using the negative log-likelihood of the input with respect to the reconstructed distribution. This anomaly score is held up against a threshold which is calculated dynamically depending on the current state to achieve a lower false-positive rate. The function that obtains the threshold is a mapping of the latent space to the anomaly score, which is based on support vector regression.

Su et al (2019) proposed a similar approach to Park et al (2018), though with a few of modifications. Firstly, the encoder and decoder are composed of gated recurrent unit (GRU) layers (Cho et al, 2014) rather than LSTM layers to minimise trainable parameters. In addition to that, the previous hidden state in both the encoder and decoder is concatenated with the current input to explicitly represent temporal dependencies. Lastly, the authors make use of planar normalising flow to counteract a case where the data does not follow a Gaussian distribution. The anomaly score used in this work is the reconstruction probability, i.e. the log probability of the reconstruction given the latent representation. This is then compared with a threshold automatically obtained using the peaks over threshold (POT) method, which is based on extreme value theory. This threshold search method is also compared with grid-searching for the ideal F_1 score and yields similar results. Lastly the authors also propose a way to find the channel contributing to the anomaly.

Chen et al (2020) used a VAE based on 2D convolutional layers to detect anomalies in an industrial robot arm. The time-series data is windowed before being input to the model, hence only the last value from the window is used for the computation of the reconstruction probability, i.e. the probability that a reconstructed sample belongs to the latent distribution, which is then compared with a manually defined threshold.

Zhang et al (2021c) proposed an approach to anomaly detection using federated learning. Here, several convolutional GRU-based VAEs are trained on edge nodes which are then combined at a central aggregator node. Anomalies are detected when the reconstruction error exceeds a threshold. This threshold is searched for using the validation passes during training which maximised the F_1 score.

Li et al (2019) proposed a way to detect anomalies using recurrent GANs. Consisting of LSTM layers, both the discriminator and generator are used to compute an anomaly score. During training, the generator uses random input vectors to generate credible time series, however, during inference the input samples are mapped to a latent vector to serve as the input of the generator. To obtain the anomaly score, a loss consisting of the weighted generator residuals and the weighted discriminator cross-entropy is used to compute an anomaly detection loss for each of the time-series windows. This loss is then mapped to the original time series to obtain an anomaly score sequence, which is then presumably compared against a threshold.

7.3 Offline Training and Offline Inference but Online Capable

Guo et al (2018) suggested an improvement to recurrent VAEs which uses Gaussian mixture priors for improved distribution mapping with multi-modal data, rather than a singular Gaussian distribution. The reconstruction probability is then used as the anomaly score, which is compared to a threshold based on a chosen percentile.

von Schleinitz et al (2021) applied anomaly detection in an attempt to make vehicle dynamics time-series prediction more robust to anomalies. Anomalies in input data decrease the performance of prediction networks, therefore the authors implemented an LSTM-based VAE to correct the input sample before it is used for prediction. The anomaly score is given by a custom distance metric, similar to z-score, which is compared with a threshold obtained from a grid search.

Li et al (2021) proposed a GRU-based VAE with a custom prior acting as smoothing regularisation. This forces the VAE to feature smooth transitions in distribution parameters along the time-axis, which increases robustness. Like in most publications the authors opted for the reconstruction probability as the anomaly score.

Choi et al (2022) created an LSTM-based VAE which also added a 1D convolutional layer to the encoder input to act as a feature extractor. The anomaly score is taken as the Euclidean distance, which is compared to a manually defined threshold.

Autoencoders have also been successfully combined with GANs. Here, the generator input is a sample from a latent distribution obtained from the encoder, whereas in classical GANs the input to the generator is a random noise vector sampled from a distribution.

The first publication that proposed adding an encoder before the generator was Zhou et al (2019), where the discriminator is presented with real data and reconstructed data from a VAE rather than generated data from noise. Only the autoencoder part is used for anomaly detection, as the anomaly score is calculated using the Euclidean distance between input and reconstruction. The discriminator only plays a role in regularisation during training.

Choi et al (2020) employed a 2D CNN GAN which also featured an autoencoder-based generator in an effort to detect anomalies in power plant data. Here, the input sequences are processed into euclidean distance matrices (Lele and Richtsmeier, 1991) in order to increase robustness when faced with noise. The anomaly score is then computed using the sum of the reconstruction error, given by the L1 norm, and the weighted feature loss, obtained using the output of an intermediate layer. Here the L2 norm between the output given the real sample and the output given the generated sample is taken as the feature loss. The anomaly score is then compared to a threshold defined by a domain expert.

Geiger et al (2020) not only introduced an encoder to the GAN architecture, but also a second discriminator that measures how well the current latent distribution is mapped. To prevent mode collapse, the authors opted to use the Wasserstein loss, as well as the L2 norm between original and reconstructed samples. The anomaly score consists of a reconstruction component and a discriminator component. To ensure a comparable scale on both components, they are standardised. To combine the components sensibly, the authors have proposed to either use them as a weighted sum or as a weighted product. They also suggested calculating the reconstruction component in three different ways: using a simple difference, an area difference and the dynamic time warping distance, hence six different variations of the anomaly score are tested. The anomaly score is then compared with a threshold which is calculated as 4 standard deviations from the mean of the current window.

Zhu et al (2019) used a GAN composed of a CNN-based generator and a CNN and LSTM-based discriminator to detect anomalies. The anomaly score is computed using a weighted sum of generator and discriminator loss.

Sun et al (2019) proposed a GAN composed of a dense generator and a 1D CNN discriminator to detect anomalies in sensor data from commercial vehicles, as well as predictive maintenance. A model is trained for each channel, which is why a dense generator is applicable. No anomaly score is used since the approach made use of the discriminator to detect the anomalies.

8 Transformer Models

Transformers, first introduced by Vaswani et al (2017), have been on the rise in the machine learning research landscape, thanks to significant advances enabled in natural language processing by *BERT* (Devlin et al, 2019) and *GPT-3* (Brown et al, 2020). Transformer models have also started to find their place in time-series anomaly detection, though by far to a lesser extent than the previously discussed model families. The original transformer shows some resemblance to encoder-decoder architectures, though with a variety of improvements. First, it does not use recurrent but rather feed-forward layers to process input information in an effort to accelerate training through parallelisation, which is otherwise a sequential operation with LSTM and GRU layers. The temporal order of input data is maintained through the positional encoding of the inputs before entering the model. Furthermore, transformers use an evolution of the attention mechanism, called multi-headed attention, which allows the model to attend to information more effectively. A typical transformer block contains a feed-forward layer as well as a multi-headed attention layer, transformers can consist of several encoder and decoder blocks. An overview of all the transformer-based modelling publications discussed in this survey is presented in Table 7.

8.1 Online Training and Online Inference

Like with Section 6.1, no transformer that trains and infers in an online manner exists, though in theory such approaches could work in a similar way to the solutions discussed in Subsection 5.1.

Table 7: Summary of all publications surveyed in the area of transformer-based time-series anomaly detection. The key is as follows, T: Training, I: Inference, L: LSTM, G: GRU, CL: ConvLSTM, C: CNN, D: Dense, M: Miscellaneous, Gen: Generative, Tran: Transformer, AD: Anomaly Detection, SS: Semi-supervised, US: Unsupervised

	Model	Online		Architecture						AD
Authors	Class	T	I	L	G	CL	C	D	M	Type
Zhang et al (2021b)	Tran		✓					✓		US
Tuli et al (2022)	Tran		✓					✓		US
Xu et al (2022)	Tran							✓		US
Chen et al (2021)	Tran							✓		SS
Wang et al (2022)	Tran							✓		SS

8.2 Offline Training and Online Inference

Zhang et al (2021b) first combined VAEs with transformers (TransAnomaly) to detect anomalies. Here, the latent representation output by the transformer encoder is a distribution that is sampled from to reconstruct the input signal. The anomaly score is taken as the reconstruction probability which is then compared with a threshold obtained with the POT method.

Tuli et al (2022) proposed using a pair of transformers which interface with each other to obtain two reconstructions. The first receives input windows as the input, whereas the second receives a concatenation between the entire time series and a focus score. The focus score is an error metric that describes the deviation between the input and reconstruction from the first transformer. Training is done in two stages, where, in the first stage, the model attempts to minimise the reconstruction error (L2 norm) between the input window and the respective decoder reconstruction. In the second stage, training becomes adversarial, the second decoder attempts to maximise the reconstruction error between the input and second decoder reconstruction, whereas the first decoder attempts to minimise it. The anomaly score is then taken as the equal sum of the reconstruction errors obtained from both decoders. The threshold is obtained using the POT method, like Zhang et al (2021b).

8.3 Offline Training and Offline Inference but Online Capable

Xu et al (2022) laid out an architectural change to regular transformers in the form of anomaly attention blocks which are used instead of self-attention blocks. These anomaly attention blocks feature another parallel pipeline which takes in a fourth parameter to model the prior association, along with the query, key and value computations found in a regular attention mechanism. This is done in an effort to model the prior association and the series association at the same time, which regular transformers cannot. The anomaly score is calculated using a product of the reconstruction error and the association discrepancy.

Chen et al (2021) added a context encoding block before the encoder block in a transformer to better model multivariate input dependencies through graph networks. Anomalies are then detected using the L2 norm held up against a threshold obtained through F_1 score-maximising grid search.

Wang et al (2022) took a similar approach to Zhang et al (2021b) by combining transformer models with VAEs. Here, the authors add a feature extraction component which upsamples input data to reduce the sensitivity of the model when faced with corrupted data. In addition to that, the authors do not use the reconstruction probability to detect anomalies, but rather by calculating the upper and lower error bounds for any given window. These are calculated using the sum of the mean value and the product between the variance and a manually set parameter.

9 Model Comparison

To help the reader make a decision on which group to focus on, a large-scale comparison outside the group structure is required. For the most-used public datasets, the anomaly detection scores have been summarised in Tables 8 and 9. As is evident, no paper used all of the datasets for evaluation, except for Tuli et al (2022).

For the MIT-BIH dataset (Table 8), it is clear that the newer, transformer-based approach by Tuli et al (2022) outperforms both prediction- and reconstructive-based models, as far as the F_1 score is concerned, although not by a large margin compared to the method by Thill et al (2021). In terms of AUC, the algorithms by Zhang et al (2021a), Zhou et al (2019) and Tuli et al (2022) all performed very similarly, with Zhang et al (2021a) having a slight edge. However, it should be noted that, unlike the other models, it only used a subset of MIT-BIH dataset, which could have skewed results. The same holds true for Thill et al (2019) and Thill et al (2021).

Moving on to SMAP (Table 8), the trend continues. The transformer-based approaches outperform the rest in terms of F_1 score, with Xu et al (2022) beating even its transformer counterparts by a large margin. When considering the AUC metric the only two approaches tested performed very similarly.

Using the MSL dataset (Table 8) tells a similar story. All transformer-based approaches perform very well, but the gap with the other approaches is smaller than with SMAP. Like with MIT-BIH, the method by Tuli et al (2022) scored the highest. Standing out from the others is TadGAN, performing relatively poorly and failing to beat the baseline provided by Hundman et al (2018) in both datasets.

For the SWaT dataset (Table 9), transformer-based approaches again score the highest, with the model by Xu et al (2022) performing best, followed closely by Chen et al (2021). For the first time, the method by Tuli et al (2022) is outperformed by non-transformer models, like the approaches by Naito et al (2021), Gong et al (2021) and Zhang et al (2021c). It should be noted that in the latter case, the score from the individual entity outside the federated

Table 8: F_1 and AUC scores for the MIT-BIH, SMAP and MSL datasets. [1]: Equivalent F_1 score calculated from publication results. [2]: Only subset of dataset is used. [3]: Score from single entity without federated learning, for more information refer to source. The best score is shown in bold.

	MIT-BIH		SMAP		MSL	
Authors	F_1	AUC	F_1	AUC	F_1	AUC
Chauhan and Vig (2015)	0.620[1]	-	-	-	-	-
Thill et al (2019)	0.810[2]	-	-	-	-	-
Hundman et al (2018)	-	-	0.855[1]	-	0.793[1]	-
Naito et al (2021)	-	-	-	-	-	-
Thill et al (2021)	0.926[2]	-	-	-	-	-
Gong et al (2021)	-	-	0.883	-	0.926	-
Zhang et al (2021a)	-	**0.996**[2]	-	-	-	-
Su et al (2019)	-	-	0.843	-	0.899	-
Zhang et al (2021c)	-	-	0.865[3]	-	0.913[3]	-
Li et al (2019)	-	-	-	-	-	-
Zhou et al (2019)	-	0.945	-	-	-	-
Geiger et al (2020)	-	-	0.704	-	0.623	-
Zhu et al (2019)	-	-	-	-	-	-
Zhang et al (2021b)	-	-	0.879	-	0.912	-
Tuli et al (2022)	**0.978**	0.989	0.892	0.992	**0.959**	**0.992**
Xu et al (2022)	-	-	**0.969**	**0.994**	0.936	0.981
Chen et al (2021)	-	-	0.904	-	0.911	-

learning framework was chosen, as it performed best. The method by Tuli et al (2022) also falls behind the one by Xu et al (2022) by a sizeable margin when the AUC metric is considered.

The WADI dataset (Table 9) is used the least of all datasets mentioned. Here the average evaluation scores also lie far below what is achieved with other data. Especially the approach by Tuli et al (2022) performed poorly with an F_1 score of only 0.495. One possible explanation for this is the fact that WADI has more than double the amount of features than the next most feature-rich dataset, which makes anomaly detection more difficult. Nevertheless, the method by Chen et al (2021) still performed respectably, outscoring all other approaches.

The SMD dataset (Table 9) is the first case where the highest F_1 score is achieved by a non-transformer model, namely the approach by Gong et al (2021), however, the algorithms by Tuli et al (2022) and Xu et al (2022) are only behind by a small margin.

It is evident that in this comparison transformer-based models are the superior approach in most datasets when it comes to anomaly detection performance. Whether they are suitable for every use-case is another question as they tend to be difficult to train (Popel and Bojar, 2018) and scale well with large amounts of data (Lan et al, 2020), while sometimes only outperforming simpler approaches by a small margin.

For all datasets, except WADI, it could be argued that approaches are reaching saturation of the evaluation metrics, indicating that it may no longer be suitable as a benchmark dataset. On the other hand, there is no approach that scores very highly across all benchmarks, which means that the general applicability of detection algorithms could still be improved.

Table 9: F_1 and AUC scores for the SWaT, WADI and SMD datasets. [1]: Equivalent F_1 score calculated from publication results. [2]: Only subset of dataset is used. [3]: Score from single entity without federated learning, for more information refer to source. The best score is shown in bold.

	SWaT		WADI		SMD	
Authors	F_1	AUC	F_1	AUC	F_1	AUC
Chauhan and Vig (2015)	-	-	-	-	-	-
Thill et al (2019)	-	-	-	-	-	-
Hundman et al (2018)	-	-	-	-	-	-
Naito et al (2021)	0.899	-	0.777	-	-	-
Thill et al (2021)	-	-	-	-	-	-
Gong et al (2021)	0.845	-	-	-	**0.970**	-
Zhang et al (2021a)	-	-	-	-	-	-
Su et al (2019)	-	-	-	-	0.886	-
Zhang et al (2021c)	0.888[3]	-	-	-	-	-
Li et al (2019)	0.810	-	0.620	-	-	-
Zhou et al (2019)	-	-	-	-	-	-
Geiger et al (2020)	-	-	-	-	-	-
Zhu et al (2019)	-	-	-	-	-	-
Zhang et al (2021b)	-	-	-	-	0.893	-
Tuli et al (2022)	0.815	0.849	0.495	**0.897**	0.961[2]	**0.997**[2]
Xu et al (2022)	**0.941**	**0.988**	-	-	0.923	0.987
Chen et al (2021)	0.910	-	**0.840**	-	-	-

10 Discussion

Of the four model groups identified, transformer-based approaches have received the least attention so far, due to their relatively recent introduction in 2017. The authors believe that the number of publications in this area is set to rise in the coming years. A new type of model may also emerge that proves particularly useful to detect anomalies in time-series data due it its ease of implementation and reliability.

After considering almost 50 papers it is clear that it is often difficult to directly compare competing solutions. This is especially clear when inspecting Tables 8 and 9, which are composed of mostly missing scores. Some publications investigate proprietary datasets exclusively (Que et al, 2019; Hsieh et al, 2019; Bayram et al, 2021; Lindemann et al, 2020; Park et al, 2018; Chen et al, 2020; von Schleinitz et al, 2021; Choi et al, 2020; Sun et al, 2019) and hence provide only a limited contribution to the state of the art, because the reader cannot properly assess relative anomaly detection performance. Another problem is that anomaly detection performance can be measured in different ways. The most common metric is the F_1 score which is used in over half of the papers, however, to obtain this score a threshold often needs to be found. To avoid this, some publications use the AUC evaluation metric which is independent of the ideal threshold. The AUC value has found application in a large number of publications, though not as many as the F_1 score. The two scores are incompatible with each other, hence comparing work that only uses one of the metrics is not possible. Then, some publications also use different types of F score. The most used version is the F_1 score, where precision and recall are equally weighted, however, some publications use the $F_{0.5}$ (Hundman et al, 2018), the $F_{0.1}$ (Malhotra et al, 2015, 2016; Chauhan and Vig, 2015) or even the $F_{0.05}$ score (Malhotra et al, 2016) which favours precision over recall. For obvious reasons, this makes a comparison of the approaches difficult, however, this is a rare pattern throughout the literature and exclusively appeared in older publications. In general, however, the research area of time-series anomaly detection needs a commonly accepted, standardised evaluation procedure that facilitates direct comparisons.

Also, self-supervised learning could be paired with unsupervised or semi-supervised anomaly detection approaches. Self-supervised learning works by using part of the unlabelled input data as the target. For example, this could be a model that, given a masked input of an image, predicts the input subset that is hidden, allowing it to learn the structure of the data by performing supervision on its own. Rather than just predicting the next time step, prediction models could be trained in a self-supervised manner by predicting randomly masked time steps of an input sequence. Anomaly detection could then be performed in a similar manner to the approaches discussed in Section 5. As far as the authors are aware, only Fu and Xue (2022) have investigated this, which indicates that potential future work could further develop such approaches. In theory, self-supervised learning could also be applied to reconstructive modelling, by creating autoencoders that not only reconstruct the input from a latent vector but also learn to reconstruct masked inputs.

Another observation made is that models that both detect and learn in an online manner rarely find application. They only appeared a few times in the predictive modelling landscape, once in the generative modelling area and nowhere in the remaining sections. One possible explanation is that with increasing model complexity and parameter count, more and more data is required to properly model the data distribution, which is not available at inference time.

In addition to that, a few enhancing aspects that have otherwise found applications in other research areas only rarely have been applied to anomaly detection. The most prominent of these is active learning, where a domain expert can actively label training samples which can be used to further improve the detection performance of a model initially trained in an unsupervised manner. Active learning allows the detection performance to be maximised in a setting where most data is unlabelled but a few samples are labelled. It should outperform classifiers trained on the small amount of labelled data and unsupervised approaches trained on the unlabelled data. The main challenge is that models trained in an unsupervised manner are inherently incompatible with labelled data. One easy way to circumvent this problem would be to use the labelled data for a more complete threshold search if the approach relies on it to detect anomalies. This would not require any further training of the model, however, it also may not offer any improvements if the new data cannot significantly change the distribution of the data used for this process. Going further, Homayouni et al (2020) have successfully enhanced autoencoder structures using labelled data by modifying the loss function to allow for the ingestion of labelled data. Another possible approach to applying active learning to anomaly detection could also be training a GAN to obtain a reliable classifier. This classifier could then be used on its own to detect anomalies, but could also be further trained online by receiving feedback from a domain expert, that correctly labels a false negative or false positive sample output by the model. Depending on the generator architecture, this method could also be combined with the approach taken by Homayouni et al (2020) instead of discarding it. This could pose an elegant solution native to labelled data, which will be investigated in future. Another possible approach would be to either train a classifier from scratch using the small amount of labelled data or perhaps make re-use some component of the trained model and turn it into a classifier. For example the encoder of an autoencoder, which already has been trained to efficiently compress useful properties from the input data, could be paired with a classification layer at its output and be retrained using the small amount of labelled data.

Another aspect that could enhance a model-based anomaly detection algorithm is automated machine learning (AutoML). Here, not only model hyperparameters are tuned automatically, but also more macro aspects of models, such as layer count or even architectures (He et al, 2021; Hutter et al, 2019). This would significantly lower the initial hurdle to implementing such an algorithm, making it usable for users not familiar with the intricacies of machine learning. One possible challenge is the availability of labelled data. Even if approaches work in a fully unsupervised manner, labelled test data is still required to validate anomaly detection approaches. The process of AutoML would require a separate test set from the one used for evaluation, which in turn would increase the requirement for labelled data.

Lastly, explainable artificial intelligence (XAI) could also be used to enhance current approaches but has not been applied to anomaly detection, to the best knowledge of the author. Like AutoML, XAI could lower the usability barrier for users as it could provide a degree of explainability to model outputs. For example, if an anomaly is detected, the channel that most likely caused the obtained model response could be highlighted to direct attention to a potential problem. So far, only Tayeh et al (2022), Zhang et al (2019), Homayouni et al (2020) and Su et al (2019) have proposed approaches that provide information about the problematic input channel. It is challenging to establish a high level of trust in data-driven models which holds back their wide-spread adoption. Closing the gap in explainability is the key to setting up data-driven models as the standard for real-world anomaly detection applications.

11 Conclusion

This survey provides an overview of the model-based approaches taken to detect anomalies in multivariate time series. They are clustered around four model groups, then further segmented into online training and online inference, offline training and online inference, and offline training and offline inference but online capable. The publications analysed were then sorted by similarities in architecture and publishing date and also compared, where possible. Several research gaps were identified, some centred around unexplored model types and configurations, but also around the lack of research focused on how humans interface with the models, i.e. how a human can interpret, train and enhance anomaly detection approaches. Therefore, this work provides the reader with a clear picture of the online time-series anomaly detection landscape, as well as potential research areas that future work could focus on and issues that hinder progress in said area.

References

Aggarwal CC (2017) Time Series and Multidimensional Streaming Outlier Detection. In: Outlier Analysis. Springer International Publishing, Cham, p 273–310, doi:10.1007/978-3-319-47578-3_9

Ahmed CM, Palleti VR, Mathur AP (2017) WADI: A water distribution testbed for research in the design of secure cyber physical systems. 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks, CySWATER 2017 pp 25–28. doi:10.1145/3055366.3055375

Bahdanau D, Cho K, Bengio Y (2015) Neural Machine Translation by Jointly Learning to Align and Translate. In: International Conference on Learning Representations (ICLR), URL https://arxiv.org/abs/1409.0473

Bayram B, Duman TB, Ince G (2021) Real time detection of acoustic anomalies in industrial processes using sequential autoencoders. Expert Systems 38(1):e12,564. doi:10.1111/exsy.12564

Blázquez-García A, Conde A, Mori U, et al (2021) A Review on Outlier/Anomaly Detection in Time Series Data. ACM Computing Surveys 54(3):1–33. doi:10.1145/3444690

Brown TB, Mann B, Ryder N, et al (2020) Language models are few-shot learners. In: Advances in Neural Information Processing Systems. Curran Associates, Inc., URL https://papers.nips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf

Chadha GS, Islam I, Schwung A, et al (2021) Deep Convolutional Clustering-Based Time Series Anomaly Detection. Sensors 21(16):5488. doi:10.3390/s21165488

Chalapathy R, Chawla S (2019) Deep Learning for Anomaly Detection: A Survey pp 1–50. URL http://arxiv.org/abs/1901.03407

Chandola V, Banerjee A, Kumar V (2009) Anomaly detection. ACM Computing Surveys 41(3):1–58. doi:10.1145/1541880.1541882

Chandola V, Banerjee A, Kumar V (2012) Anomaly Detection for Discrete Sequences: A Survey. IEEE Transactions on Knowledge and Data Engineering 24(5):823–839. doi:10.1109/TKDE.2010.235

Chauhan S, Vig L (2015) Anomaly detection in ECG time signals via deep long short-term memory networks. In: IEEE Data Science and Advanced Analytics (DSAA). IEEE, Paris, France, pp 1–7, doi:10.1109/DSAA.2015.7344872

Chen T, Liu X, Xia B, et al (2020) Unsupervised Anomaly Detection of Industrial Robots Using Sliding-Window Convolutional Variational Autoencoder. IEEE Access 8:47,072–47,081. doi:10.1109/ACCESS.2020.2977892

Chen Z, Chen D, Zhang X, et al (2021) Learning Graph Structures with Transformer for Multivariate Time Series Anomaly Detection in IoT. IEEE Internet of Things Journal doi:10.1109/JIOT.2021.3100509

Cho K, van Merriënboer B, Bahdanau D, et al (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. Association for Computational Linguistics, Doha, Qatar, pp 103–111, doi:10.3115/v1/W14-4012

Choi T, Lee D, Jung Y, et al (2022) Multivariate Time-series Anomaly Detection using SeqVAE-CNN Hybrid Model. In: International Conference on Information Networking (ICOIN). IEEE, pp 250–253, doi:10.1109/ICOIN53446.2022.9687205

Choi Y, Lim H, Choi H, et al (2020) GAN-Based Anomaly Detection and Localization of Multivariate Time Series Data for Power Plant. In: IEEE International Conference on Big Data and Smart Computing (BigComp). IEEE, pp 71–74, doi:10.1109/BigComp48618.2020.00-97

Devlin J, Chang MW, Lee K, et al (2019) BERT: Pre-training of deep bidirectional transformers for language understanding. Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference 1:4171–4186. URL https://aclanthology.org/N19-1423.pdf

Erfani SM, Rajasegarar S, Karunasekera S, et al (2016) High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. Pattern Recognition 58:121–134. doi:10.1016/j.patcog.2016.03.028

Filonov P, Lavrentyev A, Vorontsov A (2016) Multivariate Industrial Time Series with Cyber-Attack Simulation: Fault Detection Using an LSTM-based Predictive Data Model. In: Time Series Workshop at NIPS, Barcelona, URL https://arxiv.org/abs/1612.06676

Filonov P, Kitashov F, Lavrentyev A (2017) RNN-based Early Cyber-Attack Detection for the Tennessee Eastman Process. In: Time Series Workshop at ICML, Sydney, URL http://roseyu.com/time-series-workshop/submissions/TSW2017_paper_5.pdf

Fu Y, Xue F (2022) MAD: Self-Supervised Masked Anomaly Detection Task for Multivariate Time Series. In: International Joint Conference on Neural Networks (IJCNN). IEEE

Fukushima K (1980) Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics 36:193–202. doi:10.1007/BF00344251

Geiger A, Liu D, Alnegheimish S, et al (2020) TadGAN: Time Series Anomaly Detection Using Generative Adversarial Networks. In: IEEE International Conference on Big Data (Big Data). IEEE, pp 33–43, doi:10.1109/BigData50022.2020.9378139

Gers FA, Schmidhuber J, Cummins F (2000) Learning to Forget: Continual Prediction with LSTM. Neural Computation 12(10):2451–2471. doi:10.1162/089976600300015015

Goh J, Adepu S, Tan M, et al (2017) Anomaly Detection in Cyber Physical Systems Using Recurrent Neural Networks. In: IEEE High Assurance Systems Engineering (HASE). IEEE, Singapore, pp 140–145, doi:10.1109/HASE.2017.36

Gong S, Wu Z, Liu Y, et al (2021) A Prediction-Augmented AutoEncoder for Multivariate Time Series Anomaly Detection. In: Neural Information Processing (ICONIP), Lecture Notes in Computer Science, vol 13108. Springer, Cham, p 681–692, doi:10.1007/978-3-030-92185-9_56

Goodfellow IJ, Pouget-Abadie J, Mirza M, et al (2014) Generative Adversarial Nets. In: Advances in Neural Information Processing Systems. Curran Associates, Inc., URL https://papers.nips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

Gugulothu N, Malhotra P, Vig L, et al (2018) Sparse Neural Networks for Anomaly Detection in High-Dimensional Time Series. In: AI4IOT Workshop at IJCAI, Stockolm, URL https://www.zurich.ibm.com/AI4IoT/2018/AI4IoT-18_Gugulothu.pdf

Guo Y, Liao W, Wang Q, et al (2018) Multidimensional Time Series Anomaly Detection: A GRU-based Gaussian Mixture Variational Autoencoder Approach. In: Asian Conference on Machine Learning (ACML). PMLR, pp 97–112, URL http://proceedings.mlr.press/v95/guo18a/guo18a.pdf

Gupta M, Gao J, Aggarwal CC, et al (2014) Outlier Detection for Temporal Data: A Survey. IEEE Transactions on Knowledge and Data Engineering 26(9):2250–2267. doi:10.1109/TKDE.2013.184

Hawkins DM (1980) Identification of Outliers. Monographs on Statistics and Applied Probability, Springer International Publishing, Dordrecht, doi:10.1007/978-94-015-3994-4

He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Las Vegas, NV, USA, p 770–778, doi:10.1109/CVPR.2016.90, URL http://ieeexplore.ieee.org/document/7780459/

He X, Zhao K, Chu X (2021) AutoML: A survey of the state-of-the-art. Knowledge-Based Systems 212:106,622. doi:10.1016/j.knosys.2020.106622

Hochreiter S, Schmidhuber J (1997) Long Short-Term Memory. Neural Computation 9(8):1735–1780. doi:10.1162/neco.1997.9.8.1735

Homayouni H, Ghosh S, Ray I, et al (2020) An Autocorrelation-based LSTM-Autoencoder for Anomaly Detection on Time-Series Data. In: IEEE International Conference on Big Data (Big Data). IEEE, pp 5068–5077, doi:10.1109/BigData50022.2020.9378192

Hsieh RJ, Chou J, Ho CH (2019) Unsupervised Online Anomaly Detection on Multivariate Sensing Time Series Data for Smart Manufacturing. In: IEEE Service-Oriented Computing and Applications (SOCA). IEEE, pp 90–97, doi:10.1109/SOCA.2019.00021

Huang G, Liu Z, Van Der Maaten L, et al (2017) Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Honolulu, HI, p 2261–2269, doi:10.1109/CVPR.2017.243

Hundman K, Constantinou V, Laporte C, et al (2018) Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). ACM, New York, NY, USA, pp 387–395, doi:10.1145/3219819.3219845

Hutter F, Kotthoff L, Vanschoren J (2019) Automated Machine Learning. The Springer Series on Challenges in Machine Learning, Springer International Publishing, Cham, doi:10.1007/978-3-030-05318-5, URL http://link.springer.com/10.1007/978-3-030-05318-5

Kieu T, Yang B, Jensen CS (2018) Outlier Detection for Multidimensional Time Series Using Deep Neural Networks. In: Mobile Data Management (MDM). IEEE, pp 125–134, doi:10.1109/MDM.2018.00029

Kieu T, Yang B, Guo C, et al (2019) Outlier Detection for Time Series with Recurrent Autoencoder Ensembles. In: International Joint Conference on Artificial Intelligence (IJCAI). International Joint Conferences on Artificial Intelligence Organization, California, pp 2725–2732, doi:10.24963/ijcai.2019/378

Kingma DP, Welling M (2014) Auto-encoding variational bayes. URL http://arxiv.org/abs/1312.6114

Lan Z, Chen M, Goodman S, et al (2020) ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In: International Conference on Learning Representations (ICLR), URL https://openreview.net/pdf?id=H1eA7AEtvS

Lecun Y, Bottou L, Bengio Y, et al (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324. doi:10.1109/5.726791

Lele S, Richtsmeier JT (1991) Euclidean distance matrix analysis: A coordinate-free approach for comparing biological shapes using landmark data. American Journal of Physical Anthropology 86(3):415–427. doi:10.1002/ajpa.1330860307

Li D, Chen D, Jin B, et al (2019) MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In: Artificial Neural Networks and Machine Learning (ICANN), Lecture Notes in Computer Science, vol 11730. Springer, Cham, p 703–716, doi:10.1007/978-3-030-30490-4_56

Li L, Yan J, Wang H, et al (2021) Anomaly Detection of Time Series With Smoothness-Inducing Sequential Variational Auto-Encoder. IEEE Transactions on Neural Networks and Learning Systems 32(3):1177–1191.

doi:10.1109/TNNLS.2020.2980749

Lindemann B, Jazdi N, Weyrich M (2020) Anomaly detection and prediction in discrete manufacturing based on cooperative LSTM networks. In: IEEE Conference on Automation Science and Engineering (CASE). IEEE, pp 1003–1010, doi:10.1109/CASE48305.2020.9216855

Malhotra P, Vig L, Shroff G, et al (2015) Long Short Term Memory Networks for Anomaly Detection in Time Series. In: European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN), Bruges, Belgium, pp 1–650

Malhotra P, Ramakrishnan A, Anand G, et al (2016) LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection. In: Anomaly Detection Workshop at ICML, URL https://drive.google.com/file/d/0B8Dg3PBX90KNQWRwMElkVkQ0aFgzZGpzOGQtUU5DeWZYUlVV/view?resourcekey=0-7XUrmVudwhGHk1h05eKeaA

Mason A, Zhao Y, He H, et al (2019) Online Anomaly Detection of Time Series at Scale. In: International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA). IEEE, pp 1–8, doi:10.1109/CyberSA.2019.8899398

Mathur AP, Tippenhauer NO (2016) SWaT: a water treatment testbed for research and training on ICS security. In: International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater). IEEE, Figure 1, pp 31–36, doi:10.1109/CySWater.2016.7469060

Moody G, Mark R (2001) The Impact of the MIT-BIH Arrhythmia Database. IEEE Engineering in Med and Biology 20(3):45–50. doi:10.13026/C2F305

Munir M, Chattha MA, Dengel A, et al (2019a) A Comparative Analysis of Traditional and Deep Learning-Based Anomaly Detection Methods for Streaming Data. In: 18th IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, pp 561–566, doi:10.1109/ICMLA.2019.00105

Munir M, Siddiqui SA, Dengel A, et al (2019b) DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series. IEEE Access 7:1991–2005. doi:10.1109/ACCESS.2018.2886457

Naito S, Taguchi Y, Nakata K, et al (2021) Anomaly Detection for Multivariate Time Series on Large-scale Fluid Handling Plant Using Two-stage Autoencoder. In: International Conference on Data Mining Workshops (ICDMW). IEEE, pp 542–551, doi:10.1109/ICDMW53433.2021.00072

Nayak R, Pati UC, Das SK (2021) A comprehensive review on deep learning-based methods for video anomaly detection. Image and Vision Computing 106:104,078. doi:10.1016/j.imavis.2020.104078

Nguyen H, Tran K, Thomassey S, et al (2021) Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. International Journal of Information Management (IJIM) 57:102,282. doi:10.1016/j.ijinfomgt.2020.102282

Park D, Hoshi Y, Kemp CC (2018) A Multimodal Anomaly Detector for Robot-Assisted Feeding Using an LSTM-Based Variational Autoencoder. IEEE Robotics and Automation Letters 3(3):1544–1551. doi:10.1109/LRA.2018.2801475

Popel M, Bojar O (2018) Training tips for the transformer model. The Prague Bulletin of Mathematical Linguistics 110:43–70. doi:10.2478/pralin-2018-0002

Que Z, Liu Y, Guo C, et al (2019) Real-Time Anomaly Detection for Flight Testing Using AutoEncoder and LSTM. In: International Conference on Field-Programmable Technology (ICFPT). IEEE, pp 379–382, doi:10.1109/ICFPT47387.2019.00072

Sakuma T, Matsutani H (2021) An Area-Efficient Recurrent Neural Network Core for Unsupervised Time-Series Anomaly Detection. IEICE Transactions on Electronics E104.C(6):247–256. doi:10.1587/transele.2020LHP0003

von Schleinitz J, Graf M, Trutschnig W, et al (2021) VASP: An autoencoder-based approach for multivariate anomaly detection and robust time series prediction with application in motorsport. Engineering Applications of Artificial Intelligence 104:104,354. doi:10.1016/j.engappai.2021.104354

Shi X, Chen Z, Wang H, et al (2015) Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. Advances in Neural Information Processing Systems 1:802–810. doi:10.5555/2969239.2969329

Su Y, Zhao Y, Niu C, et al (2019) Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network. In: ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD). ACM, New York, NY, USA, pp 2828–2837, doi:10.1145/3292500.3330672

Suh S, Chae DH, Kang HG, et al (2016) Echo-state conditional variational autoencoder for anomaly detection. In: International Joint Conference on Neural Networks (IJCNN). IEEE, pp 1015–1022, doi:10.1109/IJCNN.2016.7727309

Sun Y, Yu W, Chen Y, et al (2019) Time Series Anomaly Detection Based on GAN. In: Social Networks Analysis, Management and Security (SNAMS). IEEE, pp 375–382, doi:10.1109/SNAMS.2019.8931714

Tambuwal AI, Neagu D (2021) Deep Quantile Regression for Unsupervised Anomaly Detection in Time-Series. SN Computer Science 2(6):475. doi:10.1007/s42979-021-00866-4

Tayeh T, Aburakhia S, Myers R, et al (2022) An Attention-Based ConvLSTM Autoencoder with Dynamic Thresholding for Unsupervised Anomaly Detection in Multivariate Time Series. Machine Learning and Knowledge Extraction 4(2):350–370. doi:10.3390/make4020015

Thill M (2022) Machine Learning and Deep Learning Approaches for Multivariate Time Series Prediction and Anomaly Detection. PhD thesis, University of Leiden, URL https://hdl.handle.net/1887/3279161

Thill M, Däubener S, Konen W, et al (2019) Anomaly Detection in Electrocardiogram Readings with Stacked LSTM Networks. In: Information Technologies - Applications and Theory (ITAT), Donovaly, pp 17–25, URL http://ceur-ws.org/Vol-2473/paper10.pdf

Thill M, Konen W, Wang H, et al (2021) Temporal convolutional autoencoder for unsupervised anomaly detection in time series. Applied Soft Computing 112:107,751. doi:10.1016/j.asoc.2021.107751

Tuli S, Casale G, Jennings NR (2022) TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. In: Very Large Databases (LVDB), Sydney

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention Is All You Need. In: Advances in Neural Information Processing Systems. Curran Associates, Inc., URL https://papers.nips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Wang X, Pi D, Zhang X, et al (2022) Variational transformer-based anomaly detection approach for multivariate time series. Measurement: Journal of the International Measurement Confederation 191:110,791. doi:10.1016/j.measurement.2022.110791

Xu J, Wu H, Wang J, et al (2022) Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy. In: International Conference on Learning Representations (ICLR), URL https://openreview.net/forum?id=LzQQ89U1qm_

Xu LD, He W, Li S (2014) Internet of things in industries: A survey. IEEE Transactions on Industrial Informatics 10(4):2233–2243.

doi:10.1109/TII.2014.2300753

Yen S, Moh M, Moh TS (2019) CausalConvLSTM: Semi-Supervised Log Anomaly Detection Through Sequence Modeling. In: IEEE International Conference On Machine Learning And Applications (ICMLA). IEEE, pp 1334–1341, doi:10.1109/ICMLA.2019.00217

Zhang C, Song D, Chen Y, et al (2019) A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data. AAAI Conference on Artificial Intelligence 33:1409–1416. doi:10.1609/aaai.v33i01.33011409

Zhang C, Zuo W, Li S, et al (2021a) Reconstruct Anomaly to Normal: Adversarially Learned and Latent Vector-Constrained Autoencoder for Time-Series Anomaly Detection. In: Pacific Rim International Conference on Artifical Inteligence (PRICAI), Lecture Notes in Computer Science, vol 13032. Springer, Cham, p 515–529, doi:10.1007/978-3-030-89363-7_39

Zhang H, Xia Y, Yan T, et al (2021b) Unsupervised Anomaly Detection in Multivariate Time Series through Transformer-based Variational Autoencoder. In: Chinese Control and Decision Conference (CCDC). IEEE, pp 281–286, doi:10.1109/CCDC52312.2021.9601669

Zhang K, Jiang Y, Seversky L, et al (2021c) Federated Variational Learning for Anomaly Detection in Multivariate Time Series. In: IEEE International Performance, Computing, and Communications Conference (IPCCC). IEEE, pp 1–9, doi:10.1109/IPCCC51483.2021.9679367

Zhou B, Liu S, Hooi B, et al (2019) BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series. In: International Joint Conference on Artificial Intelligence (IJCAI). International Joint Conferences on Artificial Intelligence Organization, California, pp 4433–4439, doi:10.24963/ijcai.2019/616

Zhu G, Zhao H, Liu H, et al (2019) A Novel LSTM-GAN Algorithm for Time Series Anomaly Detection. In: Prognostics and System Health Management Conference (PHM-Qingdao). IEEE, pp 1–6, doi:10.1109/PHM-Qingdao46334.2019.8942842

Zimek A, Schubert E, Kriegel HP (2012) A survey on unsupervised outlier detection in high-dimensional numerical data. Statistical Analysis and Data Mining 5(5):363–387. doi:10.1002/sam.11161