Stochastic pairing for contrastive anomaly detection on time series

Guillaume Chambaret[1,3], Laure Berti-Equille[2], Frédéric Bouchara[3], Emmanuel Bruno[3], Vincent Martin[1], and Fabien Chaillan[1]

[1] Naval Group, 199 av. P. G. de Gennes, Ollioules, France
[2] ESPACE-DEV, IRD, Montpellier, France
[3] LIS UMR 7020 CNRS / AMU / UTLN, Université de Toulon, France

Abstract. Anomaly detection for predictive maintenance is a significant concern for industry. Unanticipated failures cause high costs for experts involved in maintenance policy. Traditional reconstruction-based anomaly detection methods perform well on multivariate time series but they do not consider the diversity of samples in the training dataset. An abrupt change of operating conditions, which is labeled as anomaly by experts, is often not detected due to the lack of sample diversity. Besides, obtaining large volumes of labeled training data is cumbersome and sometimes impossible in practice, whereas large amounts of unlabelled data are available and could be used by unsupervised learning techniques. In this paper, we apply the principles of contrastive learning and augmentation in a self supervised way to improve feature representation of multivariate time series. We model a large variety of operating conditions with an innovative distance based stochastic method to prepare an anomaly detection downstream task. Our approach is tested on NASA SMAP/MSL public dataset and shows good performance close to the state-of-the-art anomaly detection methods.

Keywords: Time Series Regression · Augmentation · Contrastive Learning · Anomaly Detection.

1 Introduction

Industry devices such as ships, spacecrafts, engines are typically monitored from sensor-based multivariate time series, for which anomaly detection is critical for service quality management of the organization owning the devices. However, due to complex temporal dependence and multiplicity of examples, it is often though to model diversity of operational modes. For aerospace complex systems, monitoring is frequently designed on several telemetry channels (Figure 1) in order to capture various behaviours. Due to the limited number of samples, domain experts determine labels on few portions of time series which appear to be abnormal. The drawback of this approach is generally the lack of data and the lack of labels characterizing operational modes and/or failure occurrences. In particular automated methods have to deal with too few samples relative to the diversity

of pre-existing normal operating conditions. The emergence of machine learning techniques allowed the design of data-driven systems where labels entered by human experts are mainly used to validate models trained on normal samples to detect deviations. These labels are useful to anticipate failure but they do not necessarily provide information about operational modes of the system. Frequent contributions propose to split multivariate time series into windows of fixed length in order to reconstruct them with autoencoding techniques. The gap of reconstruction is then interpreted as an anomaly score where peeks correspond to potential anomalies. Our experiments are conducted on NASA Soil Moisture Active Passive (SMAP) and Mars Science Laboratory (MSL) datasets, proposed by [6], containing respectively 55 and 27 channels (Figure 1). Full data from each channel is split into two sets (train/test). Test series contain labeled anomaly segments used to compute performance metrics. In this paper, we propose the following contributions: (1) an innovative augmentation based method to design tensor pairs for contrastive learning on time series, (2) an application of self supervised contrastive learning to multivariate time series anomaly detection, and (3) an exploration of associated settings impact on anomaly detection performance.

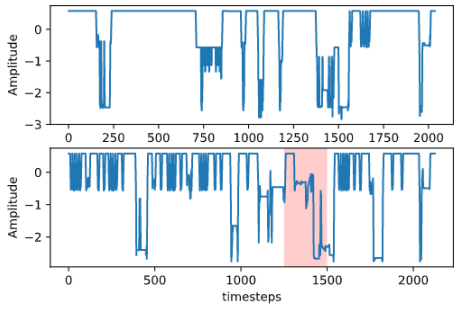


Fig. 1. Example of normalized feature measured for a single spacecraft channel with normal operating conditions (top) versus testing (bottom with an highlited anomaly).

2 Related Works

2.1 Reconstruction based time series anomaly detection

Anomaly detection is the task of detecting unseen events in data. Therefore, different unsupervised methods [2] have been proposed including reconstruction-based methods. These ones aim to compare the distance between a real input (time window) and its prediction after regression on relative features. Increasingly frequent use of autoencoders made these methods more popular in the last few years. We can mention Deep Autoencoding Gaussian Mixture Model

(DAGMM) [5] which models the density distribution of series by connecting an encoder to a gaussian mixture model. Variational autoencoders, which learn a prior distribution of data, are gradually replacing traditional recurrent autoencoders. In that way, Su et al [1] proposed a stochastic recurrent neural network to learn robust multivariate time series representations with stochastic variable connections and a normalizing flow inference. The hierarchical variational autoencoder of Li et al. [4] achieves a blend of temporal dependencies and intermetric dependencies. The latter one corresponds to non linear relationships between features for a given period and modelled by embeddings. Generative models are also employed for unsupervised anomaly detection. For example, [24] proposed Tad-GAN, a generative adversarial network with cycle consistency loss. This one measures time series reconstruction and is associated to a critic which measures the quality of mapping in latent space.

2.2 Self supervised contrastive learning

Contrastive learning is a recent technique popularized by computer vision community to learn an embedding space in which similar samples are close to each other while dissimilar ones are far apart. One common way to proceed is by using siamese networks [14] consisting of two encoders sharing weights and trained with a contrastive loss. In computer vision, contrastive loss [12] repulses different images (negative pairs) while attracting views of the same image (positive pairs). Recently, Chen et al. [11] proposed a projection network trick that maps representations to the space where the loss is applied. Contrastive learning assumes the possibility of designing positive and negative pairs. For image classification, this task is often realized by matching images from different classes to build negative pairs. In the meantime, augmented images by common techniques (e.g., rotating, flipping, burring) are associated to their original images to form positive pairs. When labels are missing or unavailable, positive / negative pairs could be obtained with other techniques in order to learn representations in a self-supervised approach [11]. Self-supervision task could be realized by augmenting the samples to still obtain positive pairs while negative pairs are designed by another techniques. For example, semantic information about the datasets have been successfully used for patient biosignals. In that way, negative pairs are built by mixing samples from different individuals [15]. We can also mention the use of unrelated examples for audio signals which do not share same contexts [16].

2.3 Time series augmentation

Augmenting time series dataset is frequently used to reduce generalization error in classification tasks. Whereas in image dataset, the meaning is kept by rotating, flipping or transforming the images, augmenting time series requires some minimal assumptions. For aperiodic signals, like spacecraft multivariate time series, traditional signal processing methods can be employed. For example, jittering (adding noise), scaling, magnitude or time warping [17, 10] can be used without loss of meaning. Recently, specific augmentation techniques to mix existing

samples (pattern mixing) rather than applying "naïve" operators have been proposed to create new training examples by computing DTW [19] or shape-DTW [20] distances between real samples. Forestier et al. [9] proposed a method to create new samples based on random multivariate time series selection with DTW comparisons. More recently, Iwana et al. [3] introduced a guided DTW-warping to select samples with more diversity. Finally, we can cite generative models as recurrent conditional GAN [18]. The last one generates new samples conserving class property of different tensors.

3 Our approach

In this paper, we propose to learn multivariate time series representations for unsupervised anomaly detection. First, as it is commonly done in unsupervised anomaly detection, we split training time series data into windows of fixed length (unchanged for every considered series). Then, we design pairs of windows by random draws on the set formed by the whole windows. In order to get so-called negative pairs, we use a distance-based approach to select dissimilar samples. The so-called positive pairs are then built with augmentation techniques. Once these operations achieved, a siamese network, composed of two encoders, is trained to learn a joint representation for pairs of windows. The network is completed by the trick proposed by Chen et al [11]. In other words, a simple projection network maps the embeddings to the final L2-normalized low dimensional representation where the contrastive loss is applied. Finally, for detection process, an anomaly score is inferred from embedding vectors computed for testing windows. The best threshold is determined by the searching method proposed by [1] where an anomaly segment is correctly detected if at least one threshold crossing occurs.

3.1 Preprocessing and stochastic pairing

For each multivariate time series (training data), we first apply a z-score scaling. This is then applied to corresponding test series. At first sight, we have no expert knowledge to determine which portion belongs to which operating mode. Note that different conditions can appear multiple times across multiple time segments. Therefore, considering multiple time windows can naively capture operating patterns. The samples are simply obtained by splitting the complete series into fixed-size windows. This task produces a dataset D where we can assume variable operating conditions depending on the position of the window along the global time series. We proceed then to pairing (Figure 2). We split equally and randomly D into 2 datasets D_1, D_2 such that $D_1 \sqcup D_2 = D$. The first one is used for reality modelling. The second one is used to make random batch of windows. Given a window W extracted from D_1, we first apply an augmentation operator to obtain another window W', (W, W') forms a positive pair. To get a negative pair, we extract a batch $B = (W_1, ..W_p)$ from D_2 and we choose then the most dissimilar windows of B related to W. Similarity is determined by computing Shape-DTW distance d between W and the observed windows of B.

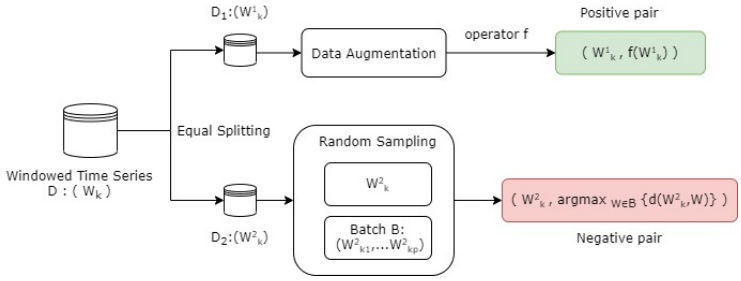


Fig. 2. Pairing process on multivariate time series.

3.2 Shape-dynamic time warping

Shape-DTW is a variation of classic Dynamic Time Warping (DTW) [19] proposed by Zhao et al. [20]. It improves distance measure for time series with shape descriptors rather than traditional alignments. Consider two multivariate time series $r = (r_1,\ldots,r_i,\ldots,r_I)$ and $s = (s_1,\ldots,s_i,\ldots,s_J)$ with sequence lengths I and J, respectively. DTW seeks for that minimizes Euclidean distance between aligned series. To proceed, DTW tests different warping in order to get the best alignment which minimizes global cost under constraints with dynamic programming. Solving DTW implies to compute a given cumulative sum matrix D using Equation 1:

$$D(i,j) = C\left(r_i, s_j\right) + \min_{(i',j')\in\{(i,j-1),(i-1,j),(i-1,j-1)\}} D\left(i',j'\right) \tag{1}$$

where $D(i,j)$ refers to cumulative sum of i-th and j-th elements, and C(r_i, s_j) is defined as the local distance between r_i and s_j. For next steps, as is often the case, Euclidean distance will be used as cost function for DTW computations with constant warping window equal to sample length. Finally, global distance for $r = (r_1,\ldots,r_i,\ldots,r_I)$, and $s = (s_1,\ldots,s_i,\ldots,s_J)$ is defined as $D(I,J)$. To compute shape-DTW, element-wise matching is replaced by shape descriptors matching. For a series r, a descriptor d at position i is defined as an extracted sub-series of r with length $l : d_{r_i} = \left(r_{i-\lceil\frac{1}{2}l\rceil},\ldots,r_i,\ldots,r_{i+\lfloor\frac{1}{2}l\rfloor}\right)^\top$. To complete descriptors located at the beginning of r (respectively at the end), zero-padding is applied to the left of the sub-series (respectively to the right). Then, series (r,s) are replaced by series of descriptors $((d_{r_0},\ldots,d_{r_I}),(d_{s_0},\ldots,d_{s_J}))$ in the previous DTW computation (eq 1). This method which tends to align descriptors instead of points is commonly used to cope with misalignment of time series. This is the case in particular when specific peeks might induce a high DTW distance measurement (Figure 3). Due to processing local time series (windows) and improper alignments, we choose this distance rather than the classic DTW.

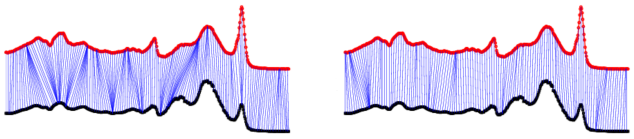


Fig. 3. DTW and Shape-DTW (right) alignments where cost increase due to curve pinchs is mitigated by the use of descriptors [20].

3.3 Augmentation methods

To augment windows of the first partitioned dataset D_1 and then form positive pairs (Figure 2), we use the following operators:

- **Identity**: the positive pair is obtained by duplication of the given window;
- **Noising**: random Gaussian noise is added to each feature with mean $\mu = 0$ and standard deviation $\sigma = 0.05$;
- **Magnitude Scaling**: each feature is summed to a scalar derived from a Gaussian distribution with mean $\mu = 1$ and standard deviation $\sigma = 0.1$;
- **Time Warping**: Time Warping based on a random smooth warping curve generated with cubic spline with 4 knots at random magnitudes with $\mu = 1$ and $\sigma = 0.2$;
- **Dual Averaging**: averaging the window with another one based on shape DTW computation on a batch from the first partition (as it is done for negative pairs).

Note that, above parameters of different methods have been obtained with tuning in a range magnitude proposed proposed by Um et al.[17]. In order to enrich previous datasets by various methods, a weighted combination of previous methods is tested with random weights such that the sum of weights equals 1. We call it mixed augmentation.

3.4 Learning architecture

In this section, we detail in Figure 4 the architecture we propose to encode features. The siamese network consists of two encoding pipelines (one by window extracted from a given pair). Encoding networks are the same for each ones and share their weights. As it is suggested by [11], the outputs of pipelines are linked to a simple projection network where the contrastive loss is applied. Projection network consists in a single layer perceptron of size 32 which will be unchanged for follow-up applications.

As processing time series implies naturally recurrent aspects, we propose to use double stacked long-short term memory layers [22] (LSTM) with downstream fully connected (FC) layers as a first way to encode pairs. The second proposed encoder is a VGG-like network [23] which consists of 2 stacked monodimensional convolutional layers. Parameters for each encoder are given below :

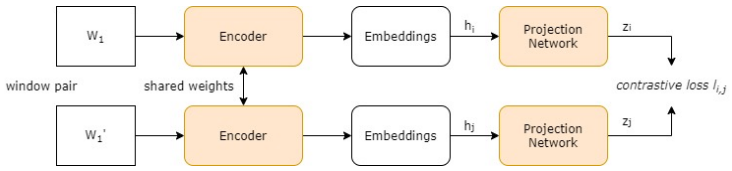


Fig. 4. Siamese architecture with projection networks.

- **Recurrent Encoder**: LTSM [128]- LSTM [64]-FC [64]. *Tanh* activation. Dropout [0.1] after recurrent layers. Embedding (output) dimension: 64.
- **Convolutional Encoder**: 2 * [Conv1D [filters: 64, kernel size: 3 strides: 1]– MaxPooling1D]– 2 * [Conv1D [filters: 32, kernel size: 3 strides: 1]– MaxPooling1D]– FC[1024]– FC[128]. ReLu activation. Dropout [0.3] after FC layers. Embedding (output) dimension: 128.

Selected loss for training is the Normalized Temperature-scaled Cross Entropy (NT-Xent) [11] defined in Equation 2:

$$\mathbf{l}_{i,j} = -\log \frac{\exp\left(sim\left(\mathbf{z}_i, \mathbf{z}_j\right)/\tau\right)}{\sum_{k=1}^{2N} 1_{[k \neq i]} \exp\left(sim\left(\mathbf{z}_i, \mathbf{z}_k\right)/\tau\right)} \tag{2}$$

Where sim represents the cosine similarity between input vectors. τ is the temperature parameter (equals to 1 for this contribution) and N is the batch size.

3.5 Anomaly score

After training the siamese network, we aim to determine an anomaly score (AS) related to testing windows and based on embedding representation. The key idea is to consider the separating property of contrastive learning which tends to take away abnormal windows from an averaging pattern (operating mode for industrial time series). In order to achieve this, we compute the Frobenius distance between embeddings of a given testing window W_{test} and the mean representations of training windows in the latent space. So, considering the encoded vector $E(W_{test})$, the anomaly score is defined as follows (Equation 3):

$$AS\left(W_{test}\right) = \left\| E\left(W_{test}\right) - \overline{\{E\left(W_{training}\right)\}} \right\|_2 \tag{3}$$

For convenience, mean representation of training windows is computed once before anomaly score processing. To limit noise and extreme peeks, exponential weighted moving average smoothing is applied to the computed score (eq 3).

3.6 Detection method

Anomalies are often stretched on time-segments which are longer than the length of sliding windows. We propose to use the point-adjust detection already employed by [1, 4]. This approach allocates a label to every observed window. If it

contains at least one point of anomaly segment, label will be 1, 0 otherwise. The goal is now to find the best threshold for the given metrics (Equation 4) :

$$Precision = \frac{TP}{TP + FP},\ Recall = \frac{TP}{TP + FN} \quad F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall} \quad (4)$$

A high precision means that the model will limit the number of false positive windows. For sensitive systems with low risk-tolerance, precision will be the first criterion of the detector. High recall is associated to a low number of undetected true anomalies. Depending on the situation, a prioritization decision has to be formulated to privilege a metric. F1 score is a popular trade-off between previous metrics to evaluate the quality of detection. It can be used without industrial assumptions and it is often proposed as single metric to compare detection methods. To evaluate our approach, we use a fixed threshold. If anomaly score is higher than this one on a single point, the window containing the point will be detected as anomaly window. In other words, its predicted label will be 1. On the contrary, windows below the threshold will have a predicted label equals to 0. To find the optimal threshold, we use the F_1 score as a criterion. We adjust the value of the threshold by a grid search procedure to select the threshold corresponding to the best F_1 value.

4 Experiments on public datasets

Experiments are conducted on MSL and SMAP public datasets [6]. Testing series are labelled with anomaly segments. For every experiment run, we suppose a fixed length for each window of 32. The mixed augmentation (combination of each augmentation method) is applied for positive pairing with best weights determined by random search: (0.36, 0.28, 0.11, 0.04, 0.21). Batch size for negative pairing will be initially fixed to 15 and explored in next section. For training process, Adam optimizer [7] is used to train the model. Each model is trained for 250 epochs with a learning rate 5 x 10e-4. To control the loss variation, train/validation partition of ratio 80/20 is applied. To reduce the overall training time, an early stopping is applied when the validation loss did not decrease for more than 15 epochs. Experiments are conducted with tensorflow [8] (v 2.4.1) and CUDA-GPU acceleration on Nvidia Quadro RTX 5000 device. We give results compared to other unsupervised methods proposed in literature in Table 1. LSTM encoding achieves better performance than VGG-encoder for both datasets. As we can observe, autoencoding techniques [1] remain the most adapted models but they have to process the complete dataset rather than trying to model data with a given fraction as we made with augmented pairs. These methods perform the best possible reconstruction regardless of the existence of anomalies in the input window. Our method achieves comparable performance and can be adapted to limited samples augmented to obtain as many pairs as needed for training. Although our approach is limited to anomalies based on existing contrast between windows, it could allow exhaustive normality modelling

Method	SMAP			MSL		
	P	R	F1	P	R	F1
TadGAN [24]	0.523	0.835	0.643	0.490	0.694	0.574
MRONet[21]	0.487	0.833	0.615	0.521	0.806	0.632
Hundman et al.[6]	0.855	0.835	0.844	0.926	0.694	0.793
OmniAnomaly [1]	0.758	0.974	0.852	0.914	0.888	0.900
VGG-contrastive	0.708	0.852	0.763	0.845	0.903	0.869
LSTM-contrastive	0.751	0.923	**0.827**	0.860	0.891	**0.874**

Table 1. SMAP / MSL results for the proposed encoders compared to state-of-the-art.

by augmentation techniques. However, our architecture works with different encoding methods, so it will be suited for contextual anomalies that are often produced in altered operating conditions. In next section we observe the influence of parameters of pairing process on performance. For conciseness, results will be given for SMAP dataset with LSTM encoder in network.

5 Effects of parameter settings

5.1 Augmentation techniques

In this section, we aim to study the influence of augmentation techniques on performance metrics. Every augmentation method is tested as a single augmentation applied to every window from the first partition (positive pair design). As we can see in Table 2, augmentations methods sharply differ in terms of performance. First, it appears, that a combination of methods performs well than separated ones.

Method	P	R	F1
Identity	**0.801**	0.765	0.782
Noising	0.788	0.826	0.807
Magnitude Scaling	0.641	0.722	0.679
Time Warping	0.414	0.603	0.491
Dual Averaging	0.700	**0.942**	0.803
Mixed	0.751	0.923	**0.827**

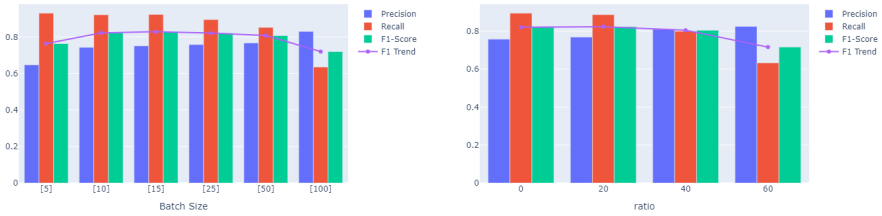
Table 2. Separate evaluation of augmentation techniques used for positive pairing. “Mixed“ refers to a weighted combination of augmentations (0.36, 0.28, 0.11, 0.04, 0.21).

Obviously, the positive pairs need diversity in order to model similarity between windows sharing contextual information. As we previously mentioned, mixed augmentation has been obtained by random search on weight combinations (with 30 iterations). The best weights do not strictly correspond to the

linear combination of metrics. Besides, we can notice that our architecture can consider pairing without augmentation in order to get the best precision. This result can be explained by the increasing number of false positives induced by an excessive augmentation which tends to add noise. For the next sections, augmentation method will be the mixed one.

5.2 Batch parameters

In this section, we study the influence of batch parameters. First we observe its size in terms of negative pairing with mixed augmentation. At a first glance, selecting a high number of windows in a batch will improve the contrast between windows for negative pairs. The bias induced by stochastic sampling will consequently be reduced. However, large batch sizes will eliminate pairs with a moderate similarity which might be useful to model soft contrast. Thus, a decrease of recall is expected due to a high sensitive contrast required. As can be seen on Figure 5a, optimal batch size is around 15 windows . For high batch sizes, contrast modelling relies on redundant pairs. It causes a significant decrease of performance in terms of recall and F_1-score. But moderate increase of size tends to slightly improve the precision. This can be useful for monitoring system with scarce anomalies. Another way to limit the bias due to the batching process is to



(a) (b)

Fig. 5. Metrics variations according to batch size and ratio mitigation

post-process the negative pairs by observing the shape-DTW distances between paired windows. For a fixed batch size of 15, we observe how varying elimination ratio of the lowest shape-DTW pairs impact metrics. Note that in order to work with a constant number of negative pairs, deleted pairs are replaced by duplicating those with the highest shape-DTW. In other words, ratio corresponds to percentage of deleted windows from the batch. As it is shown on Figure 5b, post-process batch by mitigating similar pairs may slightly improve detection. The method is clearly limited to small ratios but implies less computations than raising the batch size as it was done just before.

6 Conclusion

We proposed an application of contrastive learning to anomaly detection with a method dealing with missing assumptions, as is the case for unsupervised anomaly detection. Our approach has been successfully tested on two public datasets and tend to demonstrate that our pairing design is intrinsically linked to the nature of data. This method and the siamese architecture are generic and can be adapted to model several phenomena with an enhancing tolerance to noise. In addition, the unsupervised representations allow us to explore other downstream tasks. For example, with provided labels, it is possible to infer the class (anomaly/normal) of windows from latent representations. It has to be optimized in terms of precision and recall according to industry requirements. Future developments will focus on local anomaly scoring by comparing sequences of consecutive windows instead of computing distance to a mean embedding vector. Another possible extension could also consist in an incremental learning to aggregate previous detected anomalies to pairing process.

Acknowledgements This material is based on research fund by Naval Group. The views and results contained herein are those of the authors and should not be interpreted as necessarily representing the official policies of Naval Group.

References

1. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D. (2019, July). Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining (pp. 2828-2837).
2. Blázquez-García, A., Conde, A., Mori, U., Lozano, J. A. (2021). A Review on outlier/Anomaly Detection in Time Series Data. ACM Computing Surveys (CSUR), 54(3), 1-33.
3. Iwana, B. K., Uchida, S. (2021, January). Time series data augmentation for neural networks by time warping with a discriminative teacher. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 3558-3565). IEEE.
4. Li, Z., Zhao, Y., Han, J., Su, Y., Jiao, R., Wen, X., Pei, D. (2021, August). Multivariate Time Series Anomaly Detection and Interpretation using Hierarchical Inter-Metric and Temporal Embedding. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery Data Mining (pp. 3220-3230).
5. Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., Chen, H. (2018, February). Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In International conference on learning representations.
6. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T. (2018, July). Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery data mining (pp. 387-395).
7. Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

8. Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).
9. Forestier, G., Petitjean, F., Dau, H. A., Webb, G. I., Keogh, E. (2017, November). Generating synthetic time series to augment sparse datasets. In 2017 IEEE international conference on data mining (ICDM) (pp. 865-870). IEEE.
10. Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H. (2020). Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478.
11. Chen, T., Kornblith, S., Norouzi, M., Hinton, G. (2020, November). A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.
12. Hadsell, R., Chopra, S., LeCun, Y. (2006, June). Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06) (Vol. 2, pp. 1735-1742). IEEE.
13. Oord, A. V. D., Li, Y., Vinyals, O. (2018). Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748.
14. Melekhov, I., Kannala, J., Rahtu, E. (2016, December). Siamese network features for image matching. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 378-383). IEEE.
15. Kiyasseh, D., Zhu, T., Clifton, D. A. (2021, July). Clocs: Contrastive learning of cardiac signals across space, time, and patients. In International Conference on Machine Learning (pp. 5606-5615). PMLR.
16. Fonseca, E., Ortego, D., McGuinness, K., O'Connor, N. E., Serra, X. (2021, June). Unsupervised contrastive learning of sound event representations. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 371-375). IEEE.
17. Um, T. T., Pfister, F. M., Pichler, D., Endo, S., Lang, M., Hirche, S. Kulić, D. (2017, November). Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In Proceedings of the 19th ACM International Conference on Multimodal Interaction (pp. 216-220).
18. Esteban, C., Hyland, S. L., Rätsch, G. (2017). Real-valued (medical) time series generation with recurrent conditional gans. arXiv preprint arXiv:1706.02633.
19. Sakoe, H., Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing, 26(1), 43-49.
20. Zhao, J., Itti, L. (2018). shapedtw: Shape dynamic time warping. Pattern Recognition, 74, 171-184.
21. Baireddy, S., Desai, S. R., Mathieson, J. L., Foster, R. H., Chan, M. W., Comer, M. L., Delp, E. J. (2021). Spacecraft Time-Series Anomaly Detection Using Transfer Learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1951-1960).
22. Hochreiter, S., Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
23. Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
24. Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., Veeramachaneni, K. (2020, December). TadGAN: Time series anomaly detection using generative adversarial networks. In 2020 IEEE International Conference on Big Data (Big Data) (pp. 33-43). IEEE.