Assessing the Performance of Transformer for Time Series Anomaly Detection

Takuto Nakashima[1], Takehisa Yairi[2]

[1] *School of Engineering, Department of Aeronautics and Astronautics, The University of Tokyo*
tnutac1145@g.ecc.u-tokyo.ac.jp

[2] *Research Center for Advanced Science and Technology, The University of Tokyo*
yairi@g.ecc.u-tokyo.ac.jp

ABSTRACT

This study aims to assess the effectiveness of the Transformer-based reconstruction approach for detecting anomalies in time series data. The reconstruction error-based anomaly detection method was applied to both multivariate time series from NASA SMAP/MSL and univariate time series from UCR. Four deep learning models, including Transformer, Dilated CNN, LSTM, and MLP, were compared in terms of their ability to reconstruct input data. Dilated CNN outperformed the other models in almost all experimental results, achieving a 25% higher score than Transformer on the UCR dataset when trained with random masking, and a 60% higher score when trained with middle masking. These results suggest that the Transformer did not perform as well as expected for anomaly detection based on time series reconstruction errors, and its inferiority to Dilated CNN may be attributed to the characteristics of the time series and the limited training data. Future research should focus on developing Transformer models that can better capture the properties of time series data and investigating the relationship between the model's performance, data volume, and model complexity.

1. INTRODUCTION

Anomaly detection in systems is extremely important in practice. In factories and plants, overlooking an anomaly can lead to the manufacture of defective products, machine breakdowns, supply chain failures, etc., resulting in lower profits and loss of reliability. In the case of space systems such as satellites and deep space mission explorers, failure of the mission due to unobservable anomalies must be avoided. With the recent advances in computing power and the ability to acquire large amounts of data, anomaly detection using machine learning, especially deep learning, has become a promising approach.

Deep learning methods for time series anomaly detection are often constructed in an unsupervised or semi-supervised learning setting. This is due to the scarcity of anomalies and the difficulty of creating labels. Therefore, the strategy is based on using anomaly scores as errors based on reconstruction and prediction as self-supervised learning. Based on methods that calculate anomaly scores based on errors, such as reconstruction models by AutoEncoder and prediction models by RNN, there has been development in the direction of models such as variational AutoEncoder, LSTM, and CNN, as well as in the direction of learning methods such as adversarial learning and contrastive learning (Audibert, Michiardi, Guyard, Marti, & Zuluaga, 2020)(Geiger, Liu, Alnegheimish, Cuesta-Infante, & Veeramachaneni, 2020)(Hundman, Constantinou, Laporte, Colwell, & Soderstrom, 2018)(Malhotra, Vig, Shroff, Agarwal, et al., 2015)(Zhang et al., 2019)(Su et al., 2019). And since 2019, we have observed a trend to use Transformer (Vaswani et al., 2017), an Encoder-Decoder model that is essential for pre-training language models in natural language processing (NLP), for time series analysis tasks. In particular, the first paper using Transformer for multivariate time series anomaly detection was published in 2021. (Tuli, Casale, & Jennings, 2022)(Xu, Wu, Wang, & Long, 2022)(Jeong, Yang, Ryu, Park, & Kang, 2023) As has been established as the base model in natural language processing, it remains to be verified whether Transformer is superior to other deep learning models in time series anomaly detection.

As discussed above, Deep learning-based methods for detecting anomalies in time series have demonstrated higher F1 scores compared to traditional methods, and Transformer-based methods have emerged since 2021. However, concerns have been raised regarding the reliability of datasets and evaluation metrics (Doshi, Abudalou, & Yilmaz, 2022)(Kim, Choi, Choi, Lee, & Yoon, 2022), and it is possible that the ap-

parent performance gains observed in the scores may not be indicative of actual improvements generated by deep learning models. The UCR Anomaly Dataset has been identified as a highly reliable resource that has addressed issues such as Triviality, Mislabeling, Run-to-failure bias, and Unrealistic anomaly density found in conventional public datasets.(Wu & Keogh, 2021) Many deep learning approaches for anomaly detection assume multivariate time series data due to the expectation that these models can learn the relationships among multiple variables. The Transformer model, which leverages the Attention mechanism to effectively process multivariate vector sequences, is designed to handle multivariate inputs. Thus, to the best of our knowledge, only one previous study has validated the Transformer model using the entire UCR Dataset (Rewicki, Denzler, & Niebling, 2022). Our research involves evaluating deep learning-based anomaly detection methods using the NASA SMAP/MSL Dataset and UCR Dataset, and we compare the performance of the Transformer model against other models such as CNN, LSTM, and MLP.

2. METHODOLOGY

We worked with two separate methods in validating Transformer's performance in time series anomaly detection. Before describing each method, we define time series anomaly detection.

2.1. Problem setting

First, we provide a common definition. This is the problem setup used in the first validation and used in many time-series anomaly detection data sets. Let Test Data be the multivariate time series $\hat{X} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_T) \in \mathbb{R}^{\hat{T} \times m}$ for which we want to predict anomaly labels. Let Train Data be a multivariate time series $X = (x_1, x_2, \ldots, x_T) \in \mathbb{R}^{T \times m}$ that contains no anomalies, where each variable is the same as this Test Data and is generated from the same system. $T, \hat{T}$ is the length of the time series in the time direction and m is the number of variables (channels). Through solving tasks such as reconstruction of partial time series, a deep learning model f that takes X as input is learned. This learned model f is used to infer anomaly score S for the test data $\hat{X}$. The model f may be used as is, or further fine tuning may be performed. Thresholds $\tau = (\tau_1, \tau_2, \ldots, \tau_{\hat{T}})$ are calculated for those anomaly scores, and if $s_t > \tau_t$, the system is considered abnormal ($\hat{y}_t = 1$) at that time t; otherwise, it is considered normal ($\hat{y}_t = 0$).

The second validation was performed under another problem setting. The settings for training and test data and the calculation of anomaly scores are the same. The difference is the method of determining anomalies. If the point with the largest value in the calculated anomaly score is included in the anomaly area, it is considered to be a successful anomaly detection, and if it is not included, it is considered to be a failure. In the previous problem setting, the threshold value is used to determine the anomaly location, but in this problem setting, the most likely anomaly point is determined. The strength of this problem setting is that it can unambiguously determine the success or failure of anomaly detection, but it requires that only one anomaly interval be included in the target data set. Such data sets are limited and are well represented by the UCR Anomaly Benchmark Datasets discussed later.

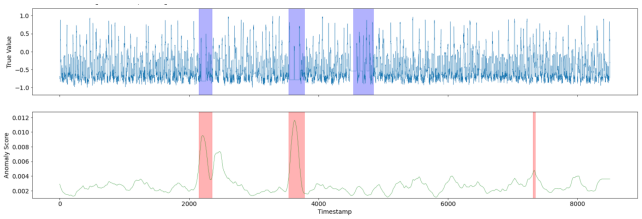

Figure 1. Example of Time Series Anomaly Detection. The upper graph is a univariate time series and the area shown in blue indicates the anomaly range. The lower graph shows the calculated anomaly and the area in red represents the predicted anomaly section.

2.2. Anomaly detection method

The two anomaly detection methods and experimental conditions conducted in our study are described below.

2.2.1. Method1: Simple Reconstruction

This method inputs a time series as a series of vectors at each time point into a deep learning model and reconstructs the original input data.

Fig.2 shows how the method 1, Simple Reconstruction works.

- First, scale the time series to the range $[-1, 1]$ for each variable and divide into windows $[W_1, W_2, \ldots, W_{T-w+1}], W_t = (x_t, x_{t+1}, \ldots, x_{t+w-1})$ where w is the length of the window.
- Random masking before entering the model. Masking is applied to the entire vector at each time point, with a ratio of 50%.
- The deep learning model will learn to reconstruct each window.

$$\mathcal{L} = \|W_t - \mathrm{Model}(\mathrm{mask}(W_t))\|_2$$

Using the model trained as described above, the test data is reconstructed in the same way and the mean squared error in each variable is calculated with respect to the original time series. In the SMAP/MSL data set described in later sections, the 0th variable is telemetry, and the other variables are command information that take values of 0 or 1, so the 0th MSL

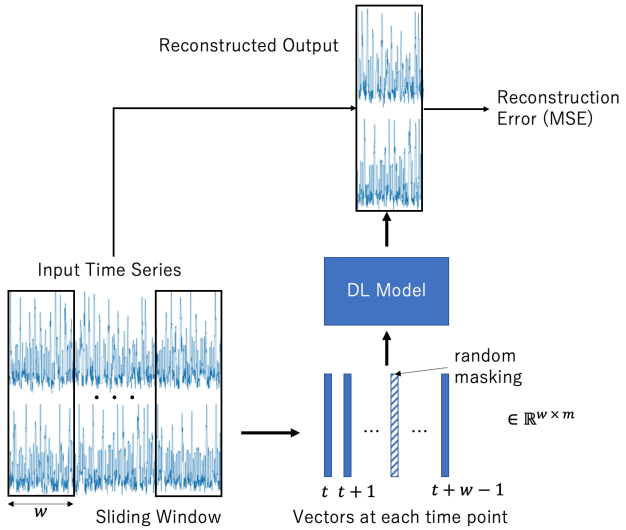

Figure 2. Method1: Simple Reconstruction

is treated as an anomaly score. The dynamic thresholding proposed by (Hundman et al., 2018) is applied to the derived anomaly score S.

Transformer, Dilated CNN, LSTM, and MLP were employed as deep learning models and their performance was compared.

2.2.2. Method2: Sub Window to Vector Reconstruction

In this method, within the divided window, the window is further divided into smaller windows, and the smaller windows are embedded in the vector. The embedded vector sequence is then fed into a deep learning model, which reconstructs the original time-series window.

Fig.3 shows how the method 2, Sub Window to Vector Reconstruction works.

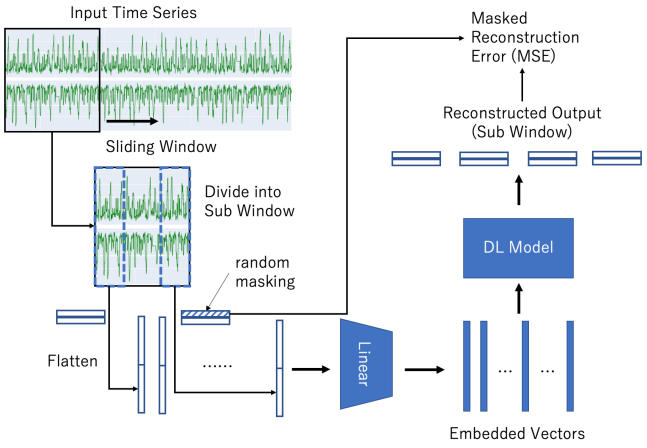

Figure 3. Method2: Sub Window to Vector Reconstruction

- First, normalize the time series and divide it into windows of length w with stride 1.
- Each window is further divided into smaller windows, which are then embedded in vectors. Masking is performed for each channel of each sub-window before being embedded in the vector. Masking can be either random or fixed.

$$\text{Embedding} = \text{Linear}(\text{Flatten}(\text{mask}(\text{SubWindow})))$$

- Input the embedding vectors into the deep learning model and reconstruct each sub window. The reconstruction error (MSE) of the masked part of the sub window is taken as the loss of the model.

The above training is performed on training data, and the model is used to reconstruct the masked areas in the test data. When testing, the selection of the segment to be masked should be fixed rather than random. Based on the evaluation criteria for the UCR dataset described later, the area to be masked should be the central sub-window within the window. This masked reconstruction error is treated as an anomaly score. This masked reconstruction error is defined as the anomaly score at the middle time of the window.

If the maximum anomaly score exists in the anomaly range, the anomaly detection of the test time series is assumed to be successful; if it does not exist, the anomaly detection is assumed to be unsuccessful.

Transformer, Dilated CNN, LSTM, and MLP were prepared as models for this method, and their performance was compared.

2.3. Experiments

We examined Method 1 on the NASA SMAP/MSL dataset, which is a multivariate time series, and Method 2 on the UCR Anomaly Benchmark Dataset, which is a univariate time series.

2.3.1. Experiment1

The SMAP/MSL dataset is a telemetry and command dataset from the Soil Moisture Active Passive (SMAP) earth observation satellite and the Mars Science Laboratory (MSL) rover, affectionately known as Curiosity. Released by Hundman et al. of NASA JPL, it is a multivariate time series consisting of a vector at each time point with the telemetry value in the 0th variable and the send/receive flag (0,1) of the command in the subsequent variables.

Following previous studies, the evaluation will be conducted as follows. Predicted anomaly labels are classified as true positive (TP), false positive (FP), true negative (TN), and false negative (FN) as described below.

- If an anomaly is predicted at any one point in a true

Table 1. SMAP/MSL statistics

	SMAP	**MSL**
Dimensions	25	55
Unique telemetry channels	55	27
All Train data length	140825	58317
All Test data length	444035	73729
Average Train data length	2560	2160
Average Test data length	8073	2731
Total anomaly sequences	69	36
Point anomalies	43(62%)	19(53%)
Contextual anomalies	26(38%)	17(47%)
Anomalies length (%)	13.1%	10.7%

anomaly section, TP is recorded for the entire anomaly section.

- Based on these total numbers, prediction (TP / (TP+FP)) and recall (TP / (TP + FN)) are calculated, and F1 score, which is the harmonic mean of these numbers, is used as the evaluation metric.

Hyper parameters in each model are shown in Table.2. For the common parameters, the window siize is 128, the batch size is also 128. The dropout ratio is 0.1.

The optimization method was AdamW. The learning rate was 0.001 and the weight decay was set to 10^{-5}. The number of epochs was 100, and the learning rate was scheduled to decrease by a factor of 0.9 every 30 epochs.

Table 2. Hyper parameters in Experiment1

	Transformer	Dilated CNN	LSTM	MLP
d_model	64	-	-	-
d_feedforward	64	-	-	-
Multi-Head	8	-	-	-
EncoderLayer	3	-	-	-
DecoderLayer	3	-	-	-
hidden dim	-	64	64	512
kernel size	-	3	-	-
depth	-	7	-	-
LSTM Layer	-	-	1	-

As for Transformer, Dilated CNN, and LSTM, the outputs of these models were converted to a vector sequence of the original time series dimensions using a linear layer. MLP's model Flatten all input vectors into one-dimensional vectors, then transform them in the linear layer and return the result to the original shape of the input.

2.3.2. Experiment2

Public datasets commonly used in time series anomaly detection research such as Yahoo, Numenta, SMAP, MSL, SDM, MBA-ECG, SWAT include the problems of Triviality, Mislabeling, Run-to-failure bias, and Unrealistic anomaly density (Wu & Keogh, 2021). The UCR Time Series Anomaly Archive Dataset are introduce by them as reliable dataset for time series anomaly detection. However, this dataset is a univariate time series and is not suitable for applying deep learning methods for multivariate time series. Hence, in Experiment 1 we used the NASA SMAP/MSL, while in Experiment 2 we will use the UCR dataset for a more meaningful comparison.

The dataset consists of 250 univariate time series of various types. Each time series has a training period in the first half and a test period in the second half, with only one anomaly within the test period.

Table 3. UCR Anomaly Dataset statistics

total train size	5302449
total test size	14051317
average train size	21210
average test size	56205
total anomaly length	49363
average anomaly length	197
anomaly ratio	0.35%

In the UCR dataset, the success or failure of anomaly detection is measured by whether the time with the highest anomaly score is included in the anomaly range. In order to calculate this, UCR Score is defined as follows.

- Let L be length of anomaly

$$L = \text{end} - \text{begin} + 1$$

- Let p be the index representing the time of the highest anomaly score.
- If $p \in (\text{begin} - \max(100, L), \text{end} + \min(100, L))$, the anomaly detection is correct.
- The UCR score is the percentage of correct answers in all 250 time series.

Hyper parameters in each model are shown in Table.4. For the common parameters, the window size is 512, the batch size is also 512. The dropout ratio is 0.2. The sub-window size is 16, this stride is 8, and the model input size is 16.

The optimization method was AdamW. The maximum learning rate was 0.0001 and one cycle LR scheduler was used with default parameters. The number of epochs was 20.

Table 4. Hyper parameters in Experiment2

	Transformer	Dilated CNN	LSTM	MLP
d_feedforward	64	-	-	-
Multi-Head	8	-	-	-
EncoderLayer	3	-	-	-
DecoderLayer	3	-	-	-
hidden dim	-	64	64	64
kernel size	-	3	-	-
depth	-	3	-	-
LSTM Layer	-	-	1	-

3. RESULTS

To summarize the results, the Dilated CNN performed best in almost all experiments. Transformer did not perform as well as expected.

3.1. Results for experiment1

Table.5 shows the averages of the evaluation metrics of anomaly detection for each channel in SMAP and Table.6 shows the evaluation metrics calculated from the sum of TP, TN, FP, and FN in each channel. n is the number of detected anomalies.

Table 5. Evaluation metrics on average of each channel in SMAP

	F1	precision	recall	ROU/AUC	n
Transformer	0.7126	0.6763	0.8187	0.901	55
Dilated CNN	0.7262	0.6881	0.854	0.9211	57
LSTM	0.7038	0.6667	0.8187	0.9002	55
MLP	**0.7419**	**0.7235**	**0.8599**	**0.9238**	**58**

Table 6. Evaluation metrics calculated on sum of TPs in SMAP

	F1	precision	recall
Transformer	0.8374	0.884	0.8631
Dilated CNN	**0.9329**	**0.918**	**0.9484**
LSTM	0.8917	0.8777	0.9062
MLP	0.9141	0.9116	0.9166

Table.7 shows the averages of the evaluation metrics of anomaly detection for each channel in MSL and Table.8 shows the evaluation metrics calculated from the sum of TP, TN, FP, and FN in each channel. n is the number of detected anomalies.

Table 7. Evaluation metrics on average of each channel in MSL

	F1	precision	recall	ROU/AUC	n
Transformer	0.5787	0.5636	0.6742	0.8267	23
Dilated CNN	**0.6559**	**0.6501**	0.7309	0.8566	25
LSTM	0.6536	0.6261	**0.7631**	**0.8701**	**26**
MLP	0.5414	0.531	0.6121	0.7953	21

3.2. Results for experiment2

At inference time, the central sub-window in the window was masked. Table.9 compares the UCR Score for the two masking methods used in training, 10% random masking and the same central masking used in inference.

4. DISCUSSION

The purpose of this study is to verify the performance of the Transformer-based reconstruction method for time series anomaly detection.

Table 8. Evaluation metrics calculated on sum of TPs in MSL

	F1	precision	recall
Transformer	0.7083	0.7548	0.6671
Dilated CNN	**0.7574**	**0.7893**	**0.7279**
LSTM	0.7462	0.7656	0.7277
MLP	0.6828	0.7771	0.6089

Table 9. UCR Score on masking strategy on training

	random mask (0.1)	middle mask
Transformer	0.2	0.24
Dilated CNN	**0.32**	0.3
LSTM	0.236	0.244
MLP	0.236	**0.304**

Time series anomaly detection methods based on deep learning have produced higher F1 scores than conventional methods, and Transformer-based methods have been emerging since 2021. However, problems with the reliability of datasets and evaluation metrics have been mentioned, and it is possible that deep learning models are not generating the performance improvements that are visible in the scores. The UCR Anomaly Dataset is considered to have resolved the Triviality, Mislabeling, Run-to-failure bias, and Unrealistic anomaly density observed in conventional public datasets, and can be judged to be highly reliable. One of the reasons why deep learning is being sought for anomaly detection is the expectation that it can successfully learn the relationships among multivariates, so many deep learning methods are based on the assumption of multivariate time series anomaly detection. The Transformer is a model that effectively processes multivariate vector sequences using the Attention mechanism, and it assumes multivariate input. Therefore, to the best of our knowledge, there is only one case in which the Transformer model has been validated using the entire UCR Dataset (Rewicki et al., 2022). In our study, we validated anomaly detection methods based on deep learning reconstruction against the UCR Dataset and verified Transfromer's performance against CNN, LSTM, and MLP.

Experimental results show that Transformer does not perform as well as other deep learning models (Dilated CNN, LSTM, MLP). Rather, Dilated CNN scored the best in almost all experimental results, especially in UCR Score, which was 60% higher in random masking and 25% higher in middle masking for anomaly detection. There are two possible reasons for this.

- Compared to natural language and images, time series have sparse information. Therefore, understanding the semantic structure by Self-Attention may not be suitable for capturing the features of time series. The reason why the performance of the Dilated CNN, which gathers information sparsely, was good is because it expands the receptive field in order from the neighborhood, so it may be able to extract the information near and far as features

in a good balance. Self-Attention takes inner products in parallel, so it may not be efficient for time series with sparse information.

- Currently, each time-series data is trained in the Train section and anomaly detection is verified in the Test section. Some time series data have a data length of 1500, which is a very small amount of data for deep learning. Transformer performance could be improved by learning long time-series data. As the Transformer's scaling law has been confirmed in natural language processing (Kaplan et al., 2020), performance may improve as the scale of training is increased. In time series anomaly detection, it is necessary to verify the relationship between the amount of data and the scale of model parameters.

5. Conclusion

A time series anomaly detection method based on reconstruction error was tested on multivariate time series of NASA SMAP/MSL and univariate time series of UCR, respectively. Transformer, Dilated CNN, LSTM, and MLP were compared as deep learning models for reconstructing the input. The Dilated CNN performed the best in almost all experimental results, scoring 25% higher than the Transformer on the UCR dataset for training with random masking, and 60% higher than the Transformer for training with middle masking. Transformer did not perform as well as expected. The performance of the Transformer was not high for anomaly detection based on time series reconstruction errors, and its inferiority to the Dilated CNN may be due to the nature of the time series and the small training scale. In the future, it is necessary to develop a Transformer model that incorporates the properties of time series and to verify the relationship between the performance of the Transformer model and the amount of data and the scale of the model.

References

Audibert, J., Michiardi, P., Guyard, F., Marti, S., & Zuluaga, M. A. (2020). Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th acm sigkdd international conference on knowledge discovery & data mining* (pp. 3395–3404).

Doshi, K., Abudalou, S., & Yilmaz, Y. (2022). Tisat: Time series anomaly transformer. *arXiv preprint arXiv:2203.05167*.

Geiger, A., Liu, D., Alnegheimish, S., Cuesta-Infante, A., & Veeramachaneni, K. (2020). Tadgan: Time series anomaly detection using generative adversarial networks. In *2020 ieee international conference on big data (big data)* (pp. 33–43).

Hundman, K., Constantinou, V., Laporte, C., Colwell, I., & Soderstrom, T. (2018). 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).

Jeong, Y., Yang, E., Ryu, J. H., Park, I., & Kang, M. (2023). Anomalybert: Self-supervised transformer for time series anomaly detection using data degradation scheme. *arXiv preprint arXiv:2305.04468*.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., ... Amodei, D. (2020). Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Kim, S., Choi, K., Choi, H.-S., Lee, B., & Yoon, S. (2022, Jun.). Towards a rigorous evaluation of time-series anomaly detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, *36*(7), 7194-7201. doi: 10.1609/aaai.v36i7.20680

Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al. (2015). Long short term memory networks for anomaly detection in time series. In *Proceedings* (Vol. 89, pp. 89–94).

Rewicki, F., Denzler, J., & Niebling, J. (2022). Is it worth it? an experimental comparison of six deep- and classical machine learning methods for unsupervised anomaly detection in time series. *arXiv preprint arXiv:2212.11080*.

Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., & Pei, D. (2019). 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).

Tuli, S., Casale, G., & Jennings, N. R. (2022). TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data. *Proceedings of VLDB*, *15*(6), 1201-1214.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wu, R., & Keogh, E. (2021). Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*.

Xu, J., Wu, H., Wang, J., & Long, M. (2022). Anomaly transformer: Time series anomaly detection with association discrepancy. In *International conference on learning representations.*

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., ... Chawla, N. V. (2019). A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 1409–1416).