Graph Attention Transformer for Unsupervised Multivariate Time Series Anomaly Detection

Tzu Hsuan Hsu[1], Yu Chee Tseng[1,2,3] and Jen Jee Chen[1]

[1] *College of AI, National Yang Ming Chiao Tung University, Taiwan*
[2] *Academia Sinica, Taiwan*
[3] *Kaohsiung Medical University, Taiwan*

Abstract

This paper studies the anomaly detection problem of multivariate time series data. Previous methods may rely on determining positive anomalies by calculating the differences in reconstructed or forecasted results. The challenges include recognition rate, scalability, and lacks of anomaly labels. We propose a self-supervised model that treats the data dependencies of multiple time series as a graph, applies a modified Transformer encoder with graph attention to learn features, and adopts a GRU to predict future data. In addition, a data selection policy with data offsetting and data dropping is designed to filter out outlier data in a self-supervised way, helping to retrieve useful features and avoid data imbalance. The model is validated on two real-world datasets and demonstrates better performance over the state-of-the-art models by about 1.0 in F1-score.

Keywords

Anomaly detection, GRU-forecasting, Graph Neural Network, Multivariate time series data, Spacecraft telemetry data, Transformer

1. Introduction

The rise of IoT and metaverse is a major driving force of sensor applications. Sensors can be installed on industrial motors to collect velocity and torque values and on human bodies to collect electrocardiogram data. Sensor data typically presents in forms of multivariate time series. Among all purposes, anomaly detection is a critical issue, whose goal is to determine whether there is any abnormality, such as heartbeat problems, in a given piece of data.

A major challenge in the anomaly detection task is lack of precise labels. While continuously collecting sensor data is feasible, there is no good basis for judging whether a segment of data is normal or abnormal. Therefore, it is typically assumed that most of the data collected in a normal environment are normal. Then, the characteristics of these data are used to train a model in an unsupervised way for judging abnormality. Another challenge is the scalability issue. When data dimension increases, it becomes more difficult to get an insight into the characteristics of abnormalities. Therefore, a data-driven approach is desired.

"MACLEAN: MAChine Learning for EArth ObservatioN Workshop 2022, in conjunction with ECML/PKDD (European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases) 2022, Grenoble, September 19, 2022"

ivanjj55.ai09@nycu.edu.tw (T. H. Hsu); yctseng@cs.nycu.edu.tw (Y. C. Tseng); jenjee@nycu.edu.tw (J. J. Chen)

Unsupervised anomaly detection solutions can be categorized as reconstruction-based [1, 2, 3, 4] and forecasting-based [5, 6]. Reconstruction-based models regenerate original data to judge abnormality. Forecasting-based models predict future data to judge abnormality. Due to the success of Graph Convolution Network and Graph Attention Network, they have been applied to time-series anomaly detection [7, 8]. More recent papers also apply graph neural networks to learn the correlations between features [7, 8].

Transformer [9] was first proposed in the field of natural language processing for learning the context of words. Recently, it has also been proved useful in the task of time series forecasting [10]. It has been shown in [11] that the transformer can learn long-term dependencies through the attention mechanism, which is suitable for use in time series anomaly detection problems.

In the collection of time series data, the problem of data imbalance sometimes occurs. For example, assuming that most of the data we collect between 0 and 100, a model may be able to detect positive anomaly when data falls in this interval pretty well. However, if time series data is not in this interval, misjudgment may occur. In order to solve this phenomenon, we propose a data offsetting mechanism to handle this issue.

The work [12] adopts the encoder of Transformer [9] and obtains the state-of-the-art scores in Soil Moisture Active Passive satellite (SMAP) [13] and Mars Science Laboratory rover (MSL) dataset. In this work, following the success path of utilizing the Transformer-based architecture, we propose a new graph attention-based Transformer model for anomaly detection. We first design a selection process, whose goal is to reduce data bias and filter out these hidden (unlabeled) abnormal data segments during the training process. We then modify the encoder of Transformer by plugging in a graph attention network. Following the forecasting approach, our model uses a GRU [14] to predict future data. The model is validated on SMAP and MSL, and it outperforms the state-of-the-art [12] in the F1-score by 0.94% and 0.24%, respectively.

The rest of this paper is organized as follows. Section 2 reviews some related works. Section 3 presents our model. Experiment results are in Section 4. Conclusions are drawn in Section 5.

2. Related Work

Machine learning methods. Time series data could be univariate or multivariate. Manually labeling positive abnormality is usually infeasible. Therefore, machine learning methods such as KNN [15] and OC-SVM [16] have been proposed. The KNN [15] method tries to find a suitable decision boundary based on categories (normal and abnormal) of those labeled data. The purpose of OC-SVM [16] is similar to KNN. They are all to find the best decision boundary on categories. Difference between KNN is that OC-SVM only needs one category (normal) of data during training. After training, it can be judged whether the testing data and the training data belong to the same category, to achieve the purpose of distinguishing normal and abnormal.

Deep learning methods. There are two types of solutions. The first type is based on reconstruction [1, 3]. InterFusion [1] used a hierarchical variational autoencoder as backbone to model the interdependence, and interdependence among multiple series simultaneously. Omnianomaly [3] stochastic recurrent neural network such as LSTM-VAE [4], extend it with a normalizing flow to generate reconstruction data. This approach uses the Peak-Over-Threshold

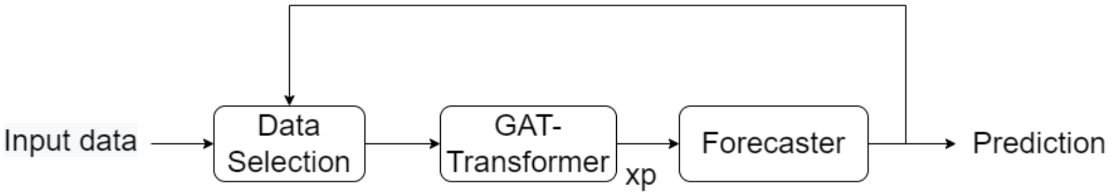


Figure 1: Overall architecture.

to automatically select the threshold. The second type is based on forecasting the future data [5, 6, 8], according to the predicted results as the standard for judging positive and abnormal. LSTM-NDT [5] using LSTM to forecast the data with an input time-series, determine if the data is abnormal by setting the threshold. The disadvantage of this method is it ignored the intermetric correlations. DAGMM [6] uses deep autoencoder Gaussian mixture models for reducing the dimension in feature space. This work predicts the next data point through a mixture of Gaussians. The input of this approach isn't a temporal sequence, it uses the multivariate variables. GDN [8] used Graph Attention Network to learn the relationship in each features and update the graph's adjacency matrix from each features, and forecasting is used to judge positive anomalies. GDN also can judge which feature data is abnormal [8]. These goals are difficult to achieve using machine learning methods. Their also have model using both reconstruction and forecast to detect anomaly. MTAD-GAT [7] used Graph Attention Network to learn the relationship between the time series data in feature direction and the time direction, used GRU to forecasting data, and used variational autoencoder with GRU to reconstructing data. This method uses Peak-Over-Threshold to automatically select the threshold for judging positive anomalies.

3. Method

We consider data in the form of multivariate time series. Each data is represented by $S = (S_1, S_2, \ldots, S_N)$, where N is the length and each $S_i \in R^K$, $i = 1 \ldots N$, is of dimension K. So S is a $K \times N$ array. The original dataset may be a long time series. All S_i are cut from the dataset in the way of a fixed-length sliding window, and the step size of sliding window is 1. Training data are all unlabeled, and only testing data are labeled as 1 (abnormal) or 0 (normal). Note that the training data may contain mostly normal, but very sparse and unknown abnormal data.

Our goal is to develop a model that can learn the characteristics of training data and judge the abnormality of test data. The proposed architecture is shown in Fig. 1, which contains three modules:

1. Data selection: This step is to purify training data by removing those potentially abnormal data. It has two steps: data offsetting and data dropping.
2. GAT-Transformer: This is a combination of Graph Attention Network and the encoder of Transformer [9] for learning the characteristics of data.
3. Forecaster: Similar to [10], we apply GRU-forecasting to predict the next data as a way of judging anomalies.

3.1. Data Selection

Data offsetting. This is to reduce the data imbalance problem as time series data are prone to have higher variation. The problem of data imbalance will occur in the case of uneven distribution of data (such as the level of data values), and this phenomenon will cause the model can't learn and judge the characteristics of the data well, this phenomenon can't solve by normalize (such as minmaxscaler or standardscaler).

We first preprocess data by offsetting each S_i.

$$\hat{S}_i = \begin{cases} \mathbf{0}, & \text{if } i = 1 \\ S_i - S_1, & \text{otherwise.} \end{cases} \tag{1}$$

Then each S_i is replaced by $\hat{S}_i$. After data offsetting, all data will be moved to the same level, solving the problem from data imbalance.

Data dropping. To filter out outliers, we apply an unsupervised approach during training by using a batch loss threshold. The training steps are as follows:

1. After training loss does not continue to decline, use training loss as a judgment criteria. A threshold $T = q_{75} + 1.5(q_{75} - q_{25})$ is calculated from all batches, where q_{75} (resp. q_{25}) is the data point covering 75% (resp. 25%) of batch losses.
2. Each batch with a loss higher than T is considered abnormal and is discarded in next epoch.
3. The above two steps are repeated for all future epochs.

Similar techniques have been applied in [17, 18] by dropping a piece of training data if its loss is too high in a batch. We adopt a per-batch dropping strategy, in hope of preserving more variants in the training data.

3.2. GAT-Transformer

Fig. 2 shows the architecture of GAT-Transformer, which uses a graph attention network to learn the characteristics of multivariate time series data. After a convolution operation, we treat each S_i as a node v_i, $i = 1 \dots N$, and these N nodes are fully connected. The input to a graph attention layer is a sequence of vectors $[h_1, h_2, \dots, h_N]$, $h_i \in R^K$, $i = 1 \dots N$. For the first layer, $h_i = S_i$.

Following the message-passing model, the hidden state h_i of node v_i after a graph attention layer is

$$h_i = \sigma(\sum_{j=1}^{N} \alpha_{ij} v_j), \tag{2}$$

where σ is a sigmoid activation function, and α_{ij} is the attention score between v_i and node v_j,

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{l=1}^{N} exp(e_{il})}, \tag{3}$$

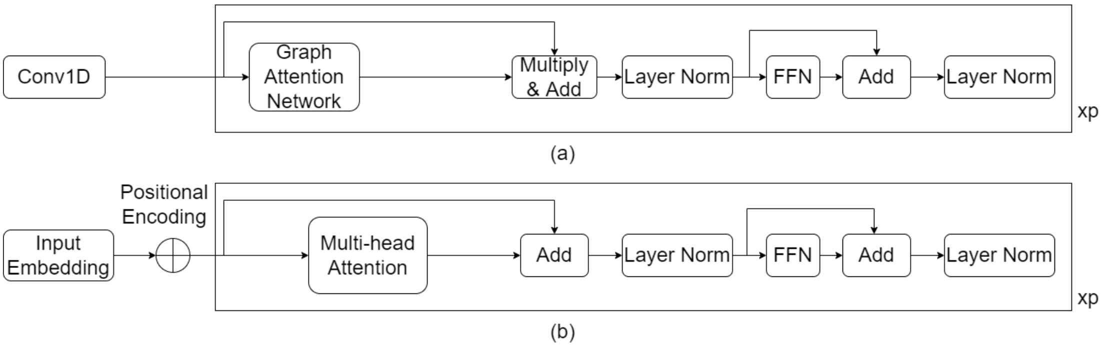


Figure 2: Comparison on the encoders of (a) GAT-Transformer and (b) Transformer.

where

$$e_{ij} = LeakyReLU(w^T \cdot (v_i \oplus v_j)). \tag{4}$$

Here, ⊕ represents concatenation, and w is a learnable column vector $\in R^{2K}$. The original Transformer is designed for natural language processing. We adopt the encoder of Transformer [9] and apply graph attention and residual calculation on it. There are three modifications in GAT-Transformer, as shown in Fig. 2. Here, we set $p = 6$ as in Transformer.

1. Conv1D: Because our data is numerical without semantic meanings, we apply Conv1D which also can solve the position information in position embedding. Using Conv1D for data embedding rather than input embedding and position embedding. data.
2. GAT: The multi-head attention is replaced by graph attention because sensor data has no semantic relation and graph attention can better learn the characteristics of time series data of similar signals.
3. Multiply & Add: Instead of directly adding attention to the original data, this operator multiplies the hidden states of nodes to the original values of S in an element-wise way and then adds S to the former result, i.e.,

$$\hat{S}_i = S_i + S_i \times h_i, \tag{5}$$

   where $i = 1 \dots N$. This is because the final results of GAT-Transformer is compressed to [0,1] by Sigmoid, the level of original data S and attention data h are not equal, compared to directly adding the two vectors in Transformer, this would maintain the outcomes of attention better (refer to Ablation study).

3.3. Forecaster

The forecaster's architecture is shown in Fig. 3. The output $[h_1, h2, \dots, h_N]$ of GAT-Transformer is sent to a GRU unit to learn their relationship. Then the final hidden state of GRU will go through a normalization layer and three fully connected layers to predict the data after S_N, denoted as $\tilde{S}_{N+1}$.

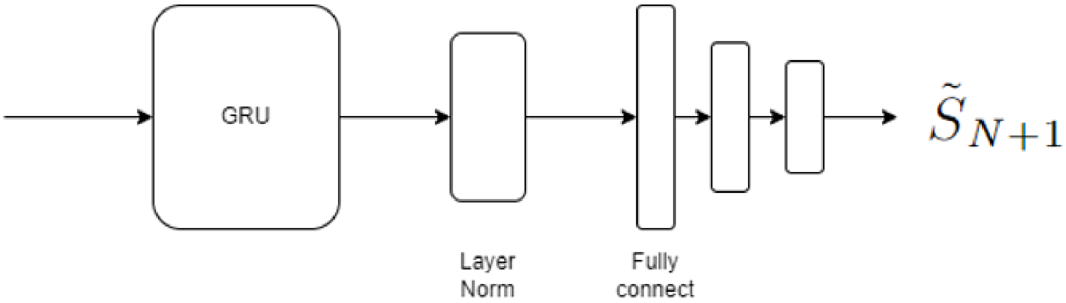


Figure 3: The forecaster.

Table 1
Dataset descriptions.

	SMAP	MSL
Number of features (K)	25	55
Training set size	135,183	58,317
Testing set size	427,617	73,729
Anomaly rate in testing set	13.13%	10.27%

During training, the loss function is defined as:

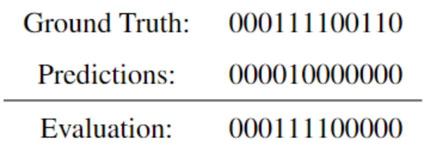
$$Loss = \sqrt{|S_{N+1} - \tilde{S}_{N+1}|^2}. \tag{6}$$

In addition, during deployment, the loss is to be compared to a threshold δ to judge abnormality.

4. Experiment Results

Datasets. We consider SMAP and MSL datasets provided by NASA [13], which are real spacecraft telemetry data from the SMAP satellite and the Curiosity rover, respectively. Both SMAP and MSL datasets are telemetried by different channels. All channel IDs are anonymized and codenamed with a letter, and all telemetry values are scaled by their max/min values. All data are multivariate time series. More information are shown in Table 1.

Metrics. We consider three performance metrics: precision, recall and F1-score. Note that F1-score is more important because abnormal data is sparse, accuracy will make all models have high scores, making it harder to distinguish the discriminant rate of each model. We follow the evaluation strategy in [3]: in a sequence of continuous abnormal observations, if any anomaly is successfully detected, the whole continuous abnormal sub-sequence is considered detected. In Fig. 4, the ground truth contains two anomaly sub-sequences. The prediction contains one anomaly detected. So the evaluation will regard the first sub-sequence as detected. Anomaly



Ground Truth:	000111100110
Predictions:	000010000000
Evaluation:	000111100000

Figure 4: The evaluation strategy. There are 12 contiguous points, first row represents ground truth; the second row represents the predictions after model; and the third line represents the calculated evaluation.

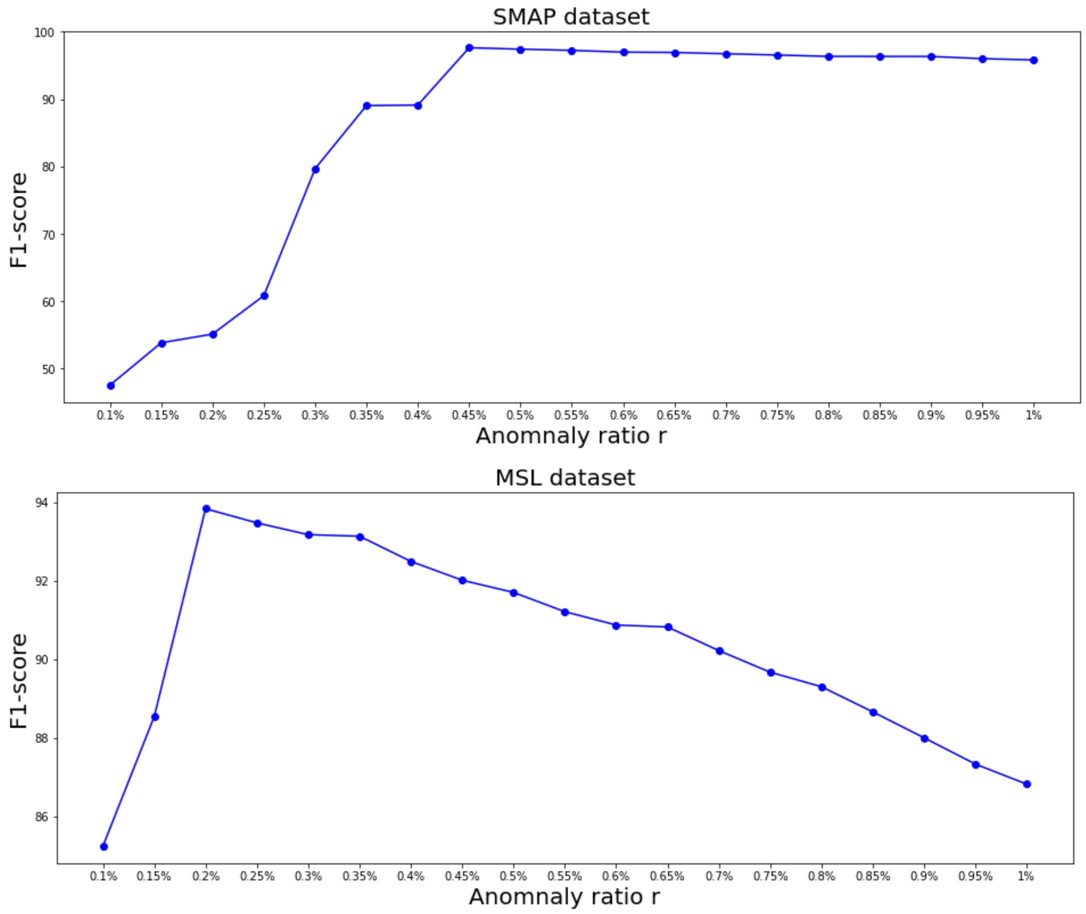


Figure 5: F1-scores under different settings of r.

threshold δ is selected as follows. We assume that there is a certain ratio r of anomaly data in the validation dataset, where $r \in$\{0.1%, 0.15%, 0.2%, 0.25%, 0.3%, 0.35%, 0.4%, 0.45%, 0.5%, 0.55%, 0.6%, 0.65%, 0.7%, 0.75%, 0.8%, 0.85%, 0.9%, 0.95%, 1%\}. Since validation data are also not labeled, given r, we set $\delta(r)$ to the value such that ratio r of data are of loss $\geq \delta(r)$.

Setup. Models are developed in PyTorch 1.7.0 with CUDA 11.0, and are trained on a server with Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz and NVIDIA GeForce RTX 3080 graphics cards. We

Table 2
Comparison results.

Dataset	SMAP			MSL		
Method	Precision	Recall	F1-score	Precision	Recall	F1-score
Clustering-based model						
ITAD	82.42	66.89	73.85	69.44	84.09	76.07
THOC	92.06	89.34	90.68	88.45	90.97	89.69
Deep-SVDD	89.93	56.02	69.04	91.92	76.63	83.58
Reconstruction-based model						
InterFusion	89.77	88.52	89.14	81.28	92.70	86.62
BeatGAN	92.38	55.85	69.91	89.75	85.42	87.53
OmniAnomaly	92.49	81.99	86.92	89.02	86.37	87.67
LSTM-VAE	92.20	67.75	78.10	85.49	79.94	82.62
Anomaly-Transformer	94.13	99.40	96.69	92.09	95.15	93.59
Forecasting-based model						
LSTM-NDT	89.65	88.46	89.05	59.34	53.74	56.40
DAGMM	86.45	56.73	68.51	89.60	63.93	74.62
ours	**95.93**	**99.41**	**97.63**	**92.38**	**95.32**	**93.83**
Reconstruction & Forecasting-based model						
MTAD-GAT	89.06	91.23	90.13	87.54	94.40	90.84

adopt the Adam optimizer with learning rate=0.0001 and batch size=256. We set the window size as 25 and the kernel size=7 for Conv1D, for GRU unit, we set the hidden size=25 and set the dropout=0.3. Models are trained for up to 300 epochs. The ratio of training to validation data is set to 9:1.

4.1. Comparisons

First, we use different threshold $\delta(r)$ with the above stated values of r to identify anomaly in the testing data of SMAP and MSL. The F1-scores are shown in Fig. 5. As can be seen, the choices of $\delta(r)$ lead to different outcomes. Below, we choose the best $r = 0.45\%$ and 0.2% for SMAP and MSL, respectively, to make further comparisons. In Table 2, we compare our model against several other models, including the current state-of-the-art Anomaly-Transformer [12].

For clustering-based models, we choose ITAD [19], THOC [20] and Deep-SVDD [21]. For reconstruction-based models, we choose InterFusion [1], BeatGAN [2], OmniAnomaly [3] and LSTM-VAE [4]. For forecasting-based models, we choose LSTM-NDT [5] and DAGMM [6]. The design of MTAD-GAT [7] includes both reconstruction and forecasting. Our GAT-transformer has the highest F1-score in both datasets, improving over Anomaly-Transformer by 0.94 and 0.24 on SMAP and MSL, respectively. GAT-Transformer also outperforms Anomaly Transformer in precision by 1.80 and 0.20 on SMAP and MSL, respectively, and in recall by 0.01 and 0.17 on SMAP and MSL, respectively.

Table 3
Ablation study.

Model	SMAP	MSL
ours	97.63	93.83
w/o data offsetting	85.55	92.89
w/o data dropping	96.46	92.00
with multi-head attention	96.65	92.81
with residual	96.57	90.32

4.2. Ablation Study

Our model contains four main designs: data offsetting, data dropping, graph attention, and multiply-and-add of attention. To understand how each design impacts performance, we conduct four ablation studies as shown in Table 3. We make observations as follows:

- The impact of data offsetting is the largest. It drops down 12.08 in F1-score for SMAP when data offestting is removed. This is because the problem of data imbalance occurs in SMAP dataset, data offsetting proves that the problem of data imbalance can be solved, the reason that is less obvious in the MSL dataset is because the data's level of different channels are relatively similar, data imbalance not phenomenon.
- The impact of data dropping is 1.17 and 1.83 for SMAP and MSL, respectively. Though the inprove isn't large, this method also proves that removing data with high loss can improve the accuracy of the model.
- The third study is to replace graph attention by the typical multi-head attention. We see that using graph attention improves F1-score by 0.98 and 1.02 on SMAP and MSL, respectively.
- The fourth study is to replace the multiply & add operator (Eq. 5) by a residual operator (i.e., $\hat{S}_i = S_i + h_i$). The improvements on F1-score are 1.17 and 1.83 on SMAP and MSL, respectively, validating the effectiveness of our design. As mentioned in the method, the data's value between [-1, 1], but the data calculated by graph attention is a value between [0, 1], in the case of different data levels, used multiply method can better than use the efficiency of graph attention.

5. Conclusions

This paper proposes an unsupervised anomaly detection model for multivariate time series data. An unsupervised data selection is first applied to purify training data. This enhances the model's capability in forecasting future data. Then we apply graph attention on Transformer with several enhancements for multivariate time series data. Our validations on two real-world datasets justify these enhancement's effectiveness.

Acknowledgments

This research is co-sponsored by ITRI, Pervasive Artificial Intelligence Research (PAIR) Labs, National Science and Technology Council (NSTC), and Advantech. This work is also financially supported by the Higher Education Sprout Project of the National Yang Ming Chiao Tung University (NYCU) and Ministry of Education (MOE), Taiwan.

References

[1] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, D. Pei, Multivariate Time Series Anomaly Detection and Interpretation using Hierarchical Inter-Metric and Temporal Embedding, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021.

[2] B. Zhou, S. Liu, B. Hooi, X. Cheng, J. Ye, Beatgan: Anomalous rhythm detection using adversarially generated time series., in: IJCAI, 2019.

[3] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, D. Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the ACM SIGKDD international conference on knowledge discovery & data mining, 2019.

[4] D. Kaur, S. N. Islam, M. Mahmud, et al., A VAE-Based Bayesian Bidirectional LSTM for Renewable Energy Forecasting, arXiv:2103.12969 (2021).

[5] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the ACM SIGKDD international conference on knowledge discovery & data mining, 2018.

[6] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, H. Chen, Deep Autoencoding Gaussian Mixture model for unsupervised anomaly detection, in: International Conference on Learning Representations, 2018.

[7] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, Q. Zhang, Multivariate time-series anomaly detection via graph attention network, in: IEEE International Conference on Data Mining, 2020.

[8] A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series, in: AAAI, 2021.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in Neural Information Processing Systems (2017).

[10] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: Beyond efficient transformer for long sequence time-series forecasting, in: AAAI, 2021.

[11] N. Kitaev, Ł. Kaiser, A. Levskaya, Reformer: The efficient transformer, arXiv:2001.04451 (2020).

[12] J. Xu, H. Wu, J. Wang, M. Long, Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy, arXiv:2110.02642 (2021).

[13] P. O'Neill, D. Entekhabi, E. Njoku, K. Kellogg, The NASA soil moisture active passive (SMAP) mission: Overview, in: IEEE International Geoscience and Remote Sensing Symposium, 2010.

[14] R. Dey, F. M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, in: IEEE international midwest symposium on circuits and systems, 2017.

[15] J. M. Keller, M. R. Gray, J. A. Givens, A fuzzy k-nearest neighbor algorithm, IEEE transactions on systems, man, and cybernetics (1985).

[16] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, Support vector method for novelty detection, Advances in Neural Information Processing Systems (1999).

[17] B. Liu, D. Wang, K. Lin, P.-N. Tan, J. Zhou, RCA: A Deep Collaborative Autoencoder Approach for Anomaly Detection, in: IJCAI, 2021.

[18] V. Shah, X. Wu, S. Sanghavi, Choosing the sample with lowest loss makes SGD robust, in: International Conference on Artificial Intelligence and Statistics, 2020.

[19] Y. Shin, S. Lee, S. Tariq, M. S. Lee, O. Jung, D. Chung, S. S. Woo, ITAD: Integrative Tensor-based Anomaly Detection system for reducing false positives of satellite systems, in: Proceedings of the ACM international conference on information & knowledge management, 2020.

[20] L. Shen, Z. Li, J. Kwok, Timeseries anomaly detection using temporal hierarchical one-class network, Advances in Neural Information Processing Systems (2020).

[21] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: International Conference on Machine Learning, 2018.