Enhancing Time Series Anomaly Detection Using Discretization and Word Embeddings

Lucas Pérez[1], Nahuel Costa[1], and Luciano Sánchez[1]

Computer Science Department, Polytechnic School of Engineering, University of Oviedo, Gijon, 33202, Asturias, Spain
perezlucas@uniovi.es

Abstract. Time series anomaly detection plays a pivotal role across diverse fields, including cybersecurity, healthcare and industrial monitoring. While Machine Learning and Deep Learning approaches have shown remarkable performance in these problems, finding a balance between simplicity and accuracy remains a persistent challenge. Also, although the potential of NLP methods is heavily expanding, their application in time series analysis is still to be explored, which could benefit greatly due to the properties of latent features. In this paper, we propose WE-TAD, a novel approach for unsupervised anomaly detection based on the representation of time series data as text, in order to leverage the use of well-established word embeddings. To showcase the performance of the model a series of experiments were conducted on a diverse set of anomaly detection datasets widely used in the literature. Results demonstrate our approach can compete and even outperform state-of-the-art approaches with a simple, yet effective model.

Keywords: anomaly detection, word embeddings, time series

1 Introduction

We are currently immersed in a transition to the fourth industrial revolution, also known as Industry 4.0. Some terms that we cannot ignore considering their great relevance nowadays are the Internet of Things (IoT) and Big Data. Through the use of a large number of sensors, massive amounts of data can be collected, processed and transmitted in a distributed manner. The failure of one piece of equipment or machine can affect another machine or process dependent on it, causing a shutdown of the production line. Such failure can usually be detected as an anomaly in the data. Stoppages are often associated with huge costs due to different aspects, such as loss of production, failure to meet delivery deadlines, deterioration of equipment, etc. Therefore, anomaly detection has experienced a great increase of interest for obvious economic reasons. Although it is impossible to eliminate all system failures, it is possible to detect them and, as time permits, be proactive to solve them or minimize damage.

Anomaly detection techniques have typically used a unsupervised approach where algorithms learn from a clean dataset and are evaluated over a set with

anomalous samples. Traditionally, shallow methods based on classical techniques such as autoregressive models, ARIMA and their variants [27] were employed. Additionally, tree-based models like Isolation Forest [29], and adaptations of support vector machines such as One-Class SVM [1], found utility in this domain. Also, approaches based on dimensionality reduction such as PCA [7] and autoencoders have been effectively applied [12]. With the evolution of Deep Learning, more modern models like Recurrent Neural Networks (RNNs) or Convolutional Neural Networks (CNNs) combined with other techniques have been proposed that allow both to improve results and to be applied to larger problems. For example, Omnianomaly [24] uses a stochastic RNN and a planar normalizing flow to generate reconstruction probabilities. MERLIN [18] is a parameter-free method capable of finding anomaly discords by iterating and comparing neighboring subsequences of the time series. MAD-GAN [10] relies in a Long Short-Term Memory network (LSTM) based GAN to model the time series distribution. MTAD-GAT [28] applies two graph attention layers in parallel to model both feature and temporal correlations that are fed to a GRU layer to subsequently pass the outputs to forecasting and reconstruction models. More recent methods, such as USAD [2] are able to achieve fast training by means of an architecture based on adversely trained autoencoders. GDN [4] introduces an embedding vector for each sensor to learn a graph of relationships between data modes and uses attention-based forecasting and deviation scoring to output anomaly scores. More recently, in TranAD [26] the authors proposed a new architecture based on Transformers and is complemented using a two phase adversarial training phase and Meta Learning.

Although recent works such as [4] or [26] use mechanisms originally applied to NLP problems like attention, there are still several NLP techniques that could be useful for anomaly detection problems that have not yet been fully exploited. Applying NLP techniques to time series problem solving is highly interesting, since text and time series have a high number of significant similarities. First, both have a sequential nature: in time series the points are ordered by timestamps while in text the words are ordered to obtain a meaning. Secondly, both exhibit temporal dependence: in time series a point depends on its antecedents while in text, words also often depend on their context. Furthermore, time series usually exhibit trends and patterns, in the same way that repetition of words or keywords occurs in text.

In this regard, there are some promising papers. In [21], [13] the authors used fuzzy logic to discretize time series and applied text mining techniques to identify patterns related to the health status of aircraft engines. In [19], the method proposed uses the discretization of the time series into symbols to subsequently learn word embeddings using Skip-gram. The SAFE framework [25] also proposes time series classification tasks by means of a new neural network architecture using word embeddings. In [5], different NLP-based techniques such as SVD, a Transformer model and an LSTM network with embeddings are applied to detect anomalies in categorical time series.

Similarly, in this paper we propose to reformulate the problem of time series anomaly detection in order to benefit from NLP techniques. Contrary to existing methods we introduce a novel architecture capable of dealing not only with categorical time series, but with all types of time series. Moreover, this architecture differs from other approaches as it exploits a never before proven concepts of discretization by timestamp and of word similarity by calculating scalar product between embeddings. Thus, we achieve a simple but effective model that is at the same time computationally undemanding.

2 Experimental Study

2.1 Problem formulation

We start from a multivaluated time series $X = \{x_t\}_{t \epsilon T}$ as an ordered set of k-dimensional vectors, where each observation is collected in a specific time period and consists of k observations. It should be noted that an univariate series is a special case where the parameter k is one. The time series is splitted into training X^{train} and test sets X^{test}, of which the training set is assumed to be free of anomalies. The task is to predict whether an anomaly occurred at each time step t in the test set of the time series X^{test}.

2.2 Data preprocessing

As usual in any Machine Learning problem, data is normalized to ease model performance and training stability, for which the min-max normalization was used:

$$x_{kt} = \frac{x_{kt} - min(X_k^{train})}{max(X_k^{train}) - min(X_k^{train}) + \varepsilon'} \tag{1}$$

where x_{kt} is a point in the k channel and in the t timestamp. $min(X_k^{train})$ and $max(X_k^{train})$ are the minimum and maximum values in the k channel of the training data. The ranges obtained from the minimum and maximum of the train set are then applied to the test set. ε' is a small constant to prevent zero-division. Knowing the ranges a-priori, we normalize the data to the range [0, 1). Once the data is normalized, it is necessary to discretize it to convert the time-series into a sequence of symbols. The discretization is a simplified version of SAX (Symbolic Aggregate Approximation) [11] where the previous step of downsampling data with PAA (Piecewise Aggregate Approximation) [8] is omitted and breakpoints are equidistant. For each channel of the time-series the normalized range is divided into as many intervals as symbols we selected. Each point will be discretized as a given symbol according to the range to witch its value belongs. The number of symbols is an hyperparameter of the model wich can be tuned depending of the fine-grained discretization pursued (for which we achieved good results with 7 in our experiments). If the number of symbols is too low, the discretization will be generic but if the number is high the resulting discrete time-series will be more complex. Once discretization is complete the

values of k channels will be concatenated per timestamp. Thus, a single point in a determined timestamp will be converted into a word consisting of k symbols. Each of these words will be treated as the i-th discretized value of the time series. The vocabulary will be formed by the set of words of the train and test sets. In turn, each of these words will be composed of k symbols. In Figure 1 the discretization and generation of words for a time series of 3 channels is illustrated graphically.

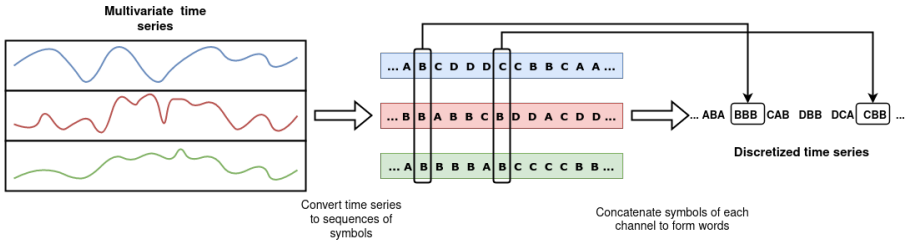

Fig. 1. Illustration of the idea behind converting time series to symbolic data (nº of symbols = 4) and extracting words from it.

2.3 Model architecture

In recent years there has been an overwhelming increase in the popularity of Deep Learning for anomaly detection, in part guided by the great advances made in NLP [9]. New models based on techniques such as Attention and specifically Transformers are beginning to be applied to anomaly detection [2], [26]. However, the use of embedding-based techniques such as Word2Vec or Glove [15], [22] is barely explored while they are a key part of NLP.

In [16], the Skip-Gram methodology was proposed, where given a target word, the model attempts to predict the context, i.e., the neighboring words by using a context window. In most cases this model is simplified by using a single word as the context, so that the skipgrams formed are pairs of words. As explained in the Introduction, there are certain similarities between text and time series. Thus, starting from the discretized time series, we intend to exploit these similarities in order to obtain embeddings that allow us to represent the time series. Subsequently, we aim to detect anomalies, which will correspond to pairs of words that are not usually found together in the same context.

The proposed architecture is a variation of Skip-Gram proposed by Mikolov [16] known as Skip-Gram Negative Sampling. Negative sampling emerged as an improvement to the original model which was computationally very expensive. This was partially solved with the application of Hierarchical Softmax but definitely improved with Negative Sampling, which simplified the problem while maintaining high quality embeddings.

In Negative Sampling, a specific number of negatives samples are randomly drawn from a noise distribution. On the one hand, in the training set there will be positive samples, which will correspond to pairs for which the context word is within the window with respect to the target word. These pairs will be labeled with 1. On the other hand, randomly chosen word pairs will be negative samples and will be labeled with a 0. Instead of using at the end of the model a layer with a Softmax function to compute the probability distribution of observing an output word given an input word, it is replaced by a Sigmoid function, whose output differentiates pairs belonging to the context (positive samples) from random (negative samples), which transforms it into a binary classification problem. In this way, the model will presumably learn embeddings where similar words are close and therefore whose scalar product is high. For further exemplification, in a multivariate time series with 3 channels, the "ABA" event may be common to appear before the "BAB" event, which would result in a high scalar product between their embeddings, so they can be considered as normal events. On the contrary, if the event "ACC" is not common to appear before "BAB", this would mean a low scalar product and may be an anomaly, since during training the generated embeddings of each words were not similar.

Figure 2 illustrates the model architecture. It has two inputs, one for the word context and another for the target. The first layer consists of two embeddings, one for each word, which will be in charge of encoding the "information" of the symbols/words fed to the model. There are different methodologies to select the size of the embeddings, but in our case for the used datasets reasonable results were achieved with a size of 300 dimensions as recommended by Melamud et al. [14]. However, for the selection of this value, successive tests were performed by increasing the size of the embeddings by 50. Therefore, the embeddings are two large matrix of real numbers of size $vocab\_size \times 300$.

Given a pair of input words and once their respective embeddings are obtained, the next step is to perform a dot product between the selected embeddings. The result will finally be passed to the last layer where the Sigmoid function is applied. At the output of the model a score between [0, 1] is obtained which would model the probability that the two words appear in the same context (close to each other) or not. A score, called perplexity[1], which is the inverse of this estimated probability is used for detecting anomalies: word/symbol pairs that have a low perplexity value will be considered normal, while a high perplexity will be associated with an anomaly.

Once the time series was discretized, the vocabulary size is obtained to initialize the embeddings of the model. After that, the pairs of skipgrams are generated from the training set, generating positive and negative samples. The model is trained using Negative Sampling. For evaluation, the pairs of symbols are generated using contiguous symbols. Once the scores are obtained, the perplexity score is calculated for each word and POT [23] is applied to detect possible anomalies through the analysis of extreme values. POT is a statistical method that uses "extreme value theory" to fit the data distribution with a Generalized Pareto

[1] Disambiguation: Do not confuse with the classical meaning in NLP.

Distribution and identify appropriate values at risk to dynamically determine threshold values.

Regarding the parameters used, as already mentioned in the previous section, the number of letters to perform the discretization of the time series was 7, the dimensionality of the embeddings was 300 and the window size for the generation was 7.

2.4 Datasets

The datasets used for benchmarking, are widely used in the anomaly detection literature and are open and publicly available. The set is composed of one univariate time series (UCR) and five multivariate time series (MBA, SMD, MSL, SMAP and MSDS), which contain a very low percentage of anomalous data. SMD, MSL and SMAP are multi-entity datasets, which are made up of different entities, corresponding to different physical units of the same type. For these, a different model per entity has been trained to finally aggregate the results by adding true positives, false positives, etc. and calculating the precision, recall and F1 score.

- **Hexagon ML/UCR Time Series** [3] is a large collection of univariate time series which has been growing and being updated over the years.
- **MIT-BIH Supraventricular Arrhythmia Database (MBA)** [17] is a collection of electrocardiogram recordings from four patients, containing multiple instances of two different kinds of anomalies. This dataset has been used for benchmarking purposes in both medical and anomaly detection articles.
- **Soil Moisture Active Passive (SMAP)** [6] is a compilation of data and telemetry from a NASA space mission that measures and maps Earth's soil moisture and freeze/thaw state from a satellite.
- **Mars Science Laboratory (MSL)** [6] is similar to SMAP dataset, but the data and telemetry are collected from the Curiosity rover during its exploration of the planet Mars. Authors such as [18] have analyzed this dataset and the previous one, detecting a large number of trivial sequences, so as in [26] the non-trivial sequences have been chosen.
- **Server Machine Dataset (SMD)** [24] is a dataset collected over 5 weeks from a large Internet company and contains resource utilization traces from 28 different machines in a cluster of computers. Similar to the previous ones, we have chosen to use non-trivial traces.
- **Multi-Source Distributed System (MSDS)** [20] is a recent high-quality multi-source data composed of distributed traces, application logs, and metrics from a complex distributed system.

3 Results

For the experimentation, the chosen metrics were the precision, recall and the F_1-score. Since the data in anomaly detection problems are usually imbalanced it

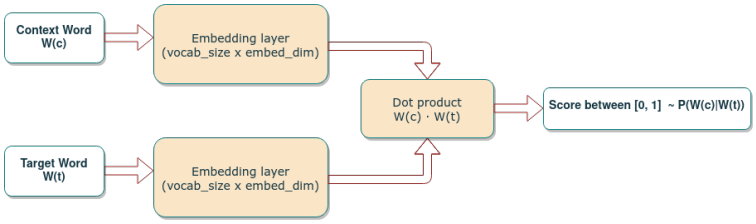

Fig. 2. Architecure for the proposed model.

Table 1. Summary of dataset characteristics used in this paper. Obtained from [26]

Dataset	Train size	Test size	Nº of channels	% of anomalies
UCR	1600	5900	1	1.88
MBA	100000	100000	2	0.14
SMD	135183	427617	38	4.16
MSL	58317	73729	55	10.72
SMAP	135183	427617	25	13.13
MSDS	146430	146430	10	5.37

is recommended to use metrics like ROC. However, since F_1-score is widely used in the literature, we consider that this metric in combination with precision and recall allow us to evaluate WETAD properly against other approaches. Table 1 compares the best precision, recall and F_1-score obtained with our proposed method, labeled as WETAD (Word Embeddings for Time series Anomaly Detection), with the results obtained by 7 anomaly detection methods (described in the Introduction). These methods are TranAD [26], GDN [4], MTAD-GAT [28], USAD [2], MAD-GAN [10], OmniAnomaly [24] and MERLIN [18]. It should be pointed that the experiments have been performed using the implementations available in the Github repository of [26].

In terms of F_1-score our method outperforms the other approaches in UCR, MBA, MSL, SMAP and MSDS datasets. Only in SMD TranAD obtains a higher F_1-score and in SMAP it draws with the same method with a F_1-score of 0.914. In the average ranking both methods also draw with a 1.8 position. The worst method by far seems to be MERLIN which, lacking parameters, seems to have a hard time adapting, especially to high-dimensional multivariate time series.

It is noteworthy mentioning that WETAD on some datasets such as MSDS and MSL reverses the trend of the other models, slightly decreasing precision or recall to improve the inverse metric. This may be due to the fact that there may be certain words that in one context are considered anomalous while in another may be normal, which would decrease precision but increase recall. For exemplification, in collective type anomalies, where all points in a sub-sequence of the time series, a word may appear repeatedly. However, the word may also appear in another non-anomalous context. In this case, the problem would be

that the word in the non-anomalous context could be detected as a false positive, leading to alterations in the metrics.

Our method in general seems to be quite stable both in precision and in recall which makes it get good results. The combination of the perplexity score together with the dynamic thresholding of POT seems to have an effect, since it allows to adjust the threshold helps set more accurate values by also considering the localized peak values in the data sequence. OmniAnomaly and MTAD-GAT seem unable to detect anomalies in the only univariate time series set, perhaps because they focus on the complex dependencies between different channels. Unlike other methods that use computationally expensive techniques such as CNNs or Transformers, WETAD achieves competitive results with a simple model.

Table 2. Precision, Recall and F_1-scores of all models with POT dynamical thresholding.

	UCR			MBA			SMD		
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1
WETAD	0.634	**0.957**	**0.763**	0.981	0.985	**0.983**	**0.979**	0.410	0.578
TranAD	**0.649**	0.472	0.547	0.957	1.000	0.978	0.927	0.646	**0.761**
GDN	0.049	0.043	0.045	0.844	**1.000**	0.915	0.717	0.495	0.586
MTAD-GAT	0.000	0.000	0.000	0.901	**1.000**	0.948	0.805	0.628	0.705
USAD	0.346	0.477	0.401	0.895	**1.000**	0.945	0.710	**0.646**	0.676
MAD-GAN	0.595	0.949	0.731	0.940	**1.000**	0.969	0.520	0.489	0.504
OmniAnomaly	0.000	0.000	0.000	0.859	**1.000**	0.924	0.775	0.555	0.646
MERLIN	0.374	0.698	0.487	**0.985**	0.049	0.094	0.132	0.539	0.213

	MSL			SMAP			MSDS			Avg.
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Ranking
WETAD	**0.960**	0.653	**0.777**	0.841	**1.000**	**0.914**	0.825	**1.000**	**0.904**	**1.8**
TranAD	0.247	**1.000**	0.396	0.842	**1.000**	**0.914**	**1.000**	0.803	0.890	**1.8**
GDN	0.241	**1.000**	0.389	**0.848**	0.985	0.912	**1.000**	0.803	0.890	4.2
MTAD-GAT	0.144	**1.000**	0.252	0.782	**1.000**	0.878	**1.000**	0.611	0.758	5.0
USAD	0.239	**1.000**	0.385	0.819	**1.000**	0.900	**1.000**	0.796	0.886	4.2
MAD-GAN	0.232	**1.000**	0.377	0.821	**1.000**	0.901	**1.000**	0.611	0.758	4.3
OmniAnomaly	0.237	**1.000**	0.383	0.818	**1.000**	0.900	**1.000**	0.796	0.887	4.8
MERLIN	0.140	0.372	0.204	0.157	0.999	0.273	0.726	0.311	0.435	6.7

4 Conclusions and future work

In this paper we propose WETAD, a new approach based on NLP techniques to identify anomalies in time series. We first apply a discretization on the time series and then train a Skip-Gram Negative Sampling model by generating word pairs and attempting to learn representations of these using word embeddings.

Experiments on a large set of datasets and a selection of state-of-the-art algorithms have shown that our method can compete with current techniques of much higher complexity and even improve results. In future work, we aim to improve the model by adding more advanced NLP techniques such as Attention combined with the discretized time series. It may also be of interest to test the model with SAX discretization, the inclusion of fuzzy logic or the improvement of signal preprocessing as in [21]. Another aspect to consider for future work is the use of other metrics also suited for imbalanced datasets (such as ROC score) in combination with those currently included. Finally, since WETAD is a straightforward model, it is easy to install in real environments, so an analysis on training times and computational complexity could be performed in view of implementations in power-limited industrial IoT devices.

References

1. Arenas-García, J., Gómez-Verdejo, V., Navia-Vazquez, A.: RLS adaptation of one-class SVM for time series novelty detection. (2004)
2. Audibert, J., Michiardi, P., Guyard, F., Marti, S., Zuluaga, M.A.: Usad: Unsupervised anomaly detection on multivariate time series. KDD '20, New York, NY, USA, Association for Computing Machinery (2020) 3395–3404
3. Dau, H.A., Bagnall, A., Kamgar, K., Yeh, C.C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Keogh, E.: The ucr time series archive (2019)
4. Deng, A., Hooi, B.: Graph neural network-based anomaly detection in multivariate time series (2021)
5. Horak, M., Chandrasekaran, S., Tobar, G.: Nlp based anomaly detection for categorical time series (2022)
6. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery, ACM (07 2018)
7. Jin, Y., Qiu, C., Sun, L., Peng, X., Zhou, J.: Anomaly detection in time series via robust pca. In: 2017 2nd IEEE International Conference on Intelligent Transportation Engineering (ICITE). (2017) 352–355
8. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD '00, New York, NY, USA, Association for Computing Machinery (2000) 285–289
9. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: state of the art, current trends and challenges. Multimedia Tools and Applications **82**(3) (07 2022) 3713–3744
10. Li, D., Chen, D., Shi, L., Jin, B., Goh, J., Ng, S.K.: Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks (2019)
11. Lin, J., Keogh, E., Wei, L., Lonardi, S.: Experiencing sax: A novel symbolic representation of time series. Data Min. Knowl. Discov. **15** (08 2007) 107–144
12. Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., Shroff, G.: Lstm-based encoder-decoder for multi-sensor anomaly detection (2016)
13. Martínez, A., Sánchez, L., Couso, I.: Engine health monitoring for engine fleets using fuzzy radviz. In: 2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE). (2013) 1–8

14. Melamud, O., McClosky, D., Patwardhan, S., Bansal, M.: The role of context types and dimensionality in learning word embeddings. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego, California, Association for Computational Linguistics (June 2016) 1030–1040
15. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space (2013)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality (2013)
17. Moody, G., Mark, R.: The impact of the mit-bih arrhythmia database. IEEE Engineering in Medicine and Biology Magazine **20**(3) (2001) 45–50
18. Nakamura, T., Imamura, M., Mercer, R., Keogh, E.J.: Merlin: Parameter-free discovery of arbitrary length anomalies in massive time series archives. 2020 IEEE International Conference on Data Mining (ICDM) (2020) 1190–1195
19. Nalmpantis, C., Vrakas, D. In: Signal2Vec: Time Series Embedding Representation. (05 2019) 80–90
20. Nedelkoski, S., Bogatinovski, J., Mandapati, A.K., Becker, S., Cardoso, J., Kao, O.: Multi-source distributed system data for ai-powered analytics. In Brogi, A., Zimmermann, W., Kritikos, K., eds.: Service-Oriented and Cloud Computing, Cham, Springer International Publishing (2020) 161–176
21. Palacios, A., Martínez, A., Sánchez, L., Couso, I.: Sequential pattern mining applied to aeroengine condition monitoring with uncertain health data. Engineering Applications of Artificial Intelligence **44** (2015) 10–24
22. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP). (2014) 1532–1543
23. Rosso, G.: Extreme value theory for time series using peak-over-threshold method. (09 2015)
24. Su, Y., Zhao, Y., Niu, C., Liu, R., Sun, W., Pei, D.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. KDD '19, New York, NY, USA, Association for Computing Machinery (2019) 2828–2837
25. Tabassum, N., Menon, S., Jastrzębska, A.: Time-series classification with safe: Simple and fast segmented word embedding-based neural time series classifier. Information Processing & Management **59**(5) (2022) 103044
26. Tuli, S., Casale, G., Jennings, N.R.: Tranad: Deep transformer networks for anomaly detection in multivariate time series data (2022)
27. Yu, Q., Jibin, L., Jiang, L.: An improved arima-based traffic anomaly detection algorithm for wireless sensor networks. International Journal of Distributed Sensor Networks **2016** (01 2016) 1–9
28. Zhao, H., Wang, Y., Duan, J., Huang, C., Cao, D., Tong, Y., Xu, B., Bai, J., Tong, J., Zhang, Q.: Multivariate time-series anomaly detection via graph attention network (2020)
29. Zhong, S., Fu, S., Lin, L., Fu, X., Cui, Z., Wang, R.: A novel unsupervised anomaly detection for gas turbine using isolation forest. (06 2019) 1–6