An Empirical Analysis of Anomaly Detection Methods for Multivariate Time Series

Dongwen Li[†], Shenglin Zhang[†§], Yongqian Sun[*†], Yang Guo[†], Zeyu Che[†], Shiqi Chen[†], Zhenyu Zhong[†], Minghan Liang[†], Minyi Shao[†], Mingjie Li[†], Shuyang Liu[†], Yuzhi Zhang[†§], Dan Pei[¶]
[†] Nankai University, {lidongwen, guoyang22, zyzhong, minghanliang, shaominyi}@mail.nankai.edu.cn,
{zhangsl, sunyongqian, zyz}@nankai.edu.cn, {chezeyu2022, sicake_c, son_of_the_sun, ShuyangBen}@163.com
[§] Haihe Laboratory of Information Technology Application Innovation
[¶] Tsinghua University, peidan@tsinghua.edu.cn

***Abstract*—Using multivariate time series (MTS) data for anomaly detection is widely adopted in service systems, such as web services and financial businesses. Researchers have recently proposed some well-performed algorithms for MTS anomaly detection from different perspectives. When applied to the real world, we observe that none of the algorithms is adaptable to all scenarios due to the complex data and anomaly characteristics. Moreover, there is currently a lack of comprehensive analysis work of these algorithms to guide operators in selecting the appropriate one in practice. To bridge this gap, we conduct an empirical study using various real-world data to gain an in-depth understanding of state-of-the-art anomaly detection algorithms. First, we provide general recommendations to guide operators in selecting suitable models based on the volume of training data, computational resources, and effectiveness requirements. Then, we summarize the typical data characteristics and types of anomalies and offer tailored model selection suggestions for different data characteristics and anomaly types. At last, we apply the summarized model selection suggestions to all the datasets we collected. The results show that most of our suggestions can achieve better than any single algorithm alone, demonstrating the effectiveness and generalization of our recommendations.**

***Index Terms*—Multivariate Time Series, Anomaly Detection, Practical Challenges, Empirical Analysis**

I. Introduction

Web services, electricity infrastructures, and financial systems have witnessed remarkable success in recent years, resulting in an increasing number of hardware facilities and software systems [1]. One notable example is the vast number of system instances, such as service instances, containers, virtual machines, physical machines, switches, and routers, in a large-scale Web service. The reliability of these facilities and systems is crucial to ensure optimal user experience and maintain service stability [2].

For proactively detecting anomalous behavior of system instances and timely mitigate system failures, operators configure various types of metrics and continuously collect their monitoring data at a predefined time interval [3]–[6]. The monitoring metrics of system instances collectively form multivariate time series (MTS), as shown in Figure 1. MTS anomaly detection algorithms involve learning normal patterns and identifying data as an anomaly when its behavior deviates from the learned normal patterns.

* Yongqian Sun is the corresponding author.

While the recently proposed models have demonstrated good performance on experimental datasets, there is still a lack of empirical studies evaluating their performance in practical applications. Operators often rely on their experience for model selection in practice, as quantitative guidance is unavailable. To gain a better understanding of the strengths and limitations of existing state-of-the-art anomaly detection algorithms, we collect two novel real-world datasets (§III) and conduct a comprehensive empirical investigation utilizing these two datasets (§IV). Our findings reveal that anomaly detection algorithms usually face three notable challenges: 1) Inefficiency in handling large-scale MTS [7], [8]. 2) Inability to handle diverse MTS [9]. 3) Inability to detect various types of anomalies [9].

As far as we know, no comprehensive solution currently addresses all three issues simultaneously. A practical method to address the abovementioned issues is selecting an appropriate algorithm for each category of MTS with similar characteristics. To guide this process, we involve six more public datasets (§III) and compile a summary of common data characteristics and anomaly types based on historical incident reports and insights from interviews with experienced engineers [10]. We conduct extensive experiments using the public datasets from diverse application scenarios, each characterized by different data characteristics and anomaly types (§V). Through the experimental analysis, we obtain valuable insights and provide practical recommendations for applying MTS anomaly detection.

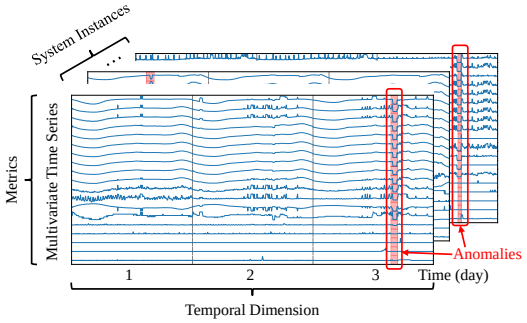

Fig. 1: The MTS of system instances.

The main contributions of this paper can be summarized as follows:

1) We highlight several practical challenges of MTS anomaly detection. To effectively tackle these challenges, we conduct a comprehensive empirical and experimental investigation using large-scale real-world datasets. To the best of our knowledge, this is the first practical investigation work into MTS anomaly detection.
2) We collect two new datasets and develop a label tool for case studies. Additionally, we do accurate labeling for the newly collected and widely used public datasets based on their data characteristics and anomaly types.
3) Through a comprehensive experimental analysis, we come up with some recommendations on choosing suitable algorithms, including a general conclusion from the perspective of effectiveness and efficiency and specific suggestions from the perspectives of MTS characteristics and anomaly types, respectively.
4) We apply the recommendations to all the datasets, and the results show that most of our suggestions can achieve better results than any single algorithm alone.

II. Background

A. MTS Anomaly Detection

Anomalies, such as unexpected fluctuations or rapid deviations from normal patterns, often indicate potential faults, including hardware crashes, service disruptions, and software bugs. The primary objective of MTS anomaly detection is to identify anomalous behavior in both system status and user behavior to prevent system crashes and mitigate potential disruptions to the business [11]. The overall pipeline for MTS anomaly detection can be summarized as follows: data collection, data preprocessing, and anomaly detection [6], [7], [9], [12]–[16]. Additionally, some models incorporate an anomaly interpretation module. Since data collection is a routine module in every system and the interpretation module is only presented in some algorithms, we focus on data preprocessing and anomaly detection modules here.

Data Preprocessing. Ensuring the perfect collection of all monitoring data in large-scale and potentially unstable system instances is challenging, often leading to missing values. MTS data commonly contain anomalies and noise, which can significantly impact the data patterns [17]. Moreover, it is necessary to make different metrics in MTS comparable, despite different metrics often exhibiting variations in amplitude. Data preprocessing is employed to address these issues. The training data and the online data require different preprocessing steps. The training data requires extreme values removal, missing values interpolation, moving average, and normalization. However, for online data, we only apply interpolation and normalization. Common normalization methods include standardization, scaling features to a specific range, and quantile transforms.

Anomaly Detection. Anomaly detection typically consists of two stages: offline model training and online anomaly detection. In the offline model training stage, sufficient data is used to train an anomaly detection model. The trained model outputs an anomaly score for each time point during the online detection stage, indicating the likelihood of being anomalous. We categorize anomaly detection algorithms into three main groups: traditional models, deep learning-based models, and others. 1) Traditional models often rely on simple assumptions, such as anomalies manifesting as extreme values. Representative works include the k-Nearest Neighbor (kNN) algorithm [18], clustering-based algorithms [19], and classification algorithms [20]. These algorithms identify anomalies using techniques like nearest-neighbor analysis, clustering, and one–class support vector machines. 2) Most deep learning-based algorithms for MTS anomaly detection are unsupervised since acquiring labeled MTS data is expensive. Several notable models in this category include MSCRED [6], DAGMM [7], USAD [12], DOMI [9], OmniAnomaly [13], SDFVAE [14], and InterFusion [16], which utilize an autoencoder (AE) or a variational AE (VAE) as their underlying framework. The underlying assumption is that the reconstructed data effectively filters out most noise and anomalies. The difference (*i.e.*, anomaly scores) between the reconstruction and true values is used to determine whether anomalies occur. Moreover, GDN [15] incorporates a structure learning method with graph neural networks (GNNs) to forecast the value of the next time point. The anomaly scores are calculated by comparing the predicted values with the actual data. The anomaly scores are then compared to a predefined threshold to determine anomalies. 3) Some anomaly detection algorithms employ rule-based or alternative methods. One representative work is JumpStarter [21], which introduces the compressed sensing technique to reconstruct data.

B. Practical Challenges

Despite the development of numerous models for MTS anomaly detection, their performance often falls short when applied to various practical scenarios. This can be attributed to several significant challenges, which can be summarized as follows:

1) Large-scale MTS: As systems expand in scale and complexity, the generation rate of MTS data experiences a significant increase. For example, online Web service systems can have several to thousands of services running on different containers, virtual machines, and physical machines. Financial institutions, such as banks, have tens of thousands of terminals operating simultaneously, resulting in massive MTS generated daily. Deep learning-based models often possess complex structures and require high training resources. Training a separate model for each MTS can be a tedious task. Although techniques like clustering and transfer learning can alleviate the training burden, the overhead of online detection remains inevitable.

2) Various MTS Patterns: Data collected from various systems and scenarios often display distinct patterns. For example, soil environment monitoring data exhibit annual periodicity and are influenced by human factors such as watering and

fertilization. Online Web services generate MTS data with a daily periodicity, which is affected by user behavior and system status. Even in the same system, different components can exhibit different MTS patterns. Additionally, service systems, especially Web service systems, experience frequent changes, including business changes and troubleshooting, leading to changes in the underlying patterns of MTS. However, existing models are mainly evaluated on limited public datasets, making their results non-representative and non-generic.

3) Various Anomaly Patterns: MTS exhibits various types of anomalies when different failures occur or when it is subjected to different attacks. For example, in a Web service, surges in page view counts often occur briefly and result in significant changes in metric values. On the other hand, when it faces an access attack, the monitored MTS tends to show longer-lasting changes with minor fluctuations. Detecting these diverse anomaly patterns in a timely and accurate manner is crucial to prevent unnecessary damage to the system. It is important to note that a single algorithm, whether supervised or unsupervised, is usually insufficient to detect all types of anomalies. Fortunately, the inherent properties of the model structure contribute to its varying ability to detect different types of anomalies. However, further comprehensive investigations are still needed to explore the detection capability of existing models in detecting different types of anomalies.

III. Preliminary

This paper aims to comprehensively study MTS anomaly detection and provide insights for practical applications. We primarily focus on investigating the following research questions:

RQ1: What are the characteristics of the most popular unsupervised algorithms?
RQ2: How do the existing algorithms work in practice?
RQ3: What are the data characteristics and anomaly types present in MTS?
RQ4: How to select the most appropriate algorithms based on data characteristics and specific anomaly types?

We must involve large-scale data and conduct thorough experiments to address these questions. In this section, we first introduce two new datasets we encounter in practice (for RQ2), and six public datasets (for RQ3 and RQ4). Then we present the overview of existing models to answer RQ1 and provide the preparation conditions for the experiments. Next, RQ2 will be resolved in §IV, and RQ3 and RQ4 in §V.

A. Dataset Selection

To ensure comprehensiveness and authenticity, we collect two datasets (D1 and D2) from our partner companies and six public datasets from various practical scenarios.

Table I presents a summary of the characteristics of the datasets. The number of entities varies from 1 to 107, while the number of metrics varies from 19 to 123. Notably, the percentage of anomalies ranges from 0.02% to 0.05%, and at least one anomaly occurs nearly every day. This highlights the importance of anomaly detection in various industries, with MTS data serving as a valuable resource for such detection.

B. Dataset Labeling

To ensure reliable labeling, three experienced operators meticulously examined each MTS and assigned anomaly labels based on data changes and incident tickets. These operators have at least three years of experience and a comprehensive understanding of the data. In cases with conflicting annotations, they engaged in discussions to reach a consensus on the labels. Furthermore, these operators dedicated two weeks to labeling the data characteristics and anomaly types for each MTS. More details about the data characteristics and anomaly types can be found in §IV-B.

We have developed a dedicated graphical user interface (GUI) tool for displaying MTS data and assisting operators in efficient data annotation. Moreover, this tool enables visual inspection of anomaly detection results, including the display of the anomaly label and the anomaly score of each time point. The GUI tool greatly facilitates our empirical study. To promote further research and development in anomaly detection, we have made the GUI tool [22] and the labeled dataset [23] openly accessible. The only exception is D2, which is not accessible due to commercial issues.

C. Model Overview

This paper primarily focuses on unsupervised algorithms, mainly due to the difficulty of obtaining sufficient high-quality labeled data for training supervised models in practical scenarios [24], [25]. We select eight popular unsupervised models for MTS anomaly detection, including one compressed sensing-based algorithm and seven deep learning-based algorithms. Table II presents the advantages, data preprocessing methods, and model structures of the eight algorithms, addressing RQ1.

D. Experimental Setup

We prefer to adopt the default hyperparameters provided in the corresponding open-source code for the studied anomaly detection models when conducting experiments. Each instance in the eight datasets has its unique anomaly detection model trained using its corresponding training data. This ensures that the anomaly detection models are specifically tailored to the characteristics of each instance. For each combination of MTS and algorithm, we conduct three times of experiments and select the best result as the final result. For datasets or specific entities that yield unsatisfactory results, we make efforts to adjust the hyperparameters and conduct multiple experiments to achieve improved results. All the experiments are run on a server with two 16C32T Intel(R) Xeon(R) Gold 5218 CPU @ 2.30 GHz, one NVIDIA(R) Tesla(R) V100S, and 192 GB RAM.

E. Evaluation Metrics

We employ two evaluation metrics, time consumption and $F_1 - score$ (F_1 for short), to evaluate the efficiency and effectiveness of the models. F_1 is the most widely used

TABLE I: Detailed information of the experimental datasets. (The symbol '#' denotes the amount of data, while the symbol '%' denotes the percentage of anomalies.)

Dataset	Source	Scenarios	#Entities	#Metrics	Time Interval	#Train	#Test	Anomalies (%)
D1	A global content service provider	Web services.	26	49	30 sec	14400	23040	0.05
D2	An Internet service provider	Network operation service.	107	22	15 min	672	672	0.02
SMD	An Internet company	/	28	38	1 min	28479	28479	0.04
ASD	An Internet company	/	12	19	5 min	8640	4320	0.05
SMAP	NASA	Global measurements of soil moisture and its freeze-thaw status.	54	25	1 min	2818	7331	0.13
MSL	NASA	The Mars rover Curiosity's operations.	27	55	1 min	4308	6100	0.11
SWaT	A water treatment plant	The real-world industrial water treatment plant operation status.	1	51	1 sec	496800	449919	0.12
WADI	A testbed	A single plant operation status.	1	123	1 sec	1048571	172801	0.06

TABLE II: An overview of unsupervised MTS anomaly detection models.

Model	Advantages	Data Preprocessing Method	Model Structures
DAGMM	• Based on time point. • Preserves the low-dimensional features and reconstruction error for anomaly detection.	Does standardization.	AE + Gaussian Mixture Model (GMM)
USAD	• Leverages the advantages of AE and adversarial training. • A straightforward model structure and a limited number of parameters.	Does standardization.	AE + Generative Adversarial Network (GAN)
OmniAnomaly	• Models the explicit temporal dependence. • Employs a VAE to map input observations to stochastic variables.	Uses zero to fill in missing values and does normalization.	RNN + VAE
DOMI	• Simultaneously extracts both categorical variables and low-dimensional data features. • Works better with MTS data that exhibits multiple normal patterns.	Uses zero to fill in missing values and does standardization.	1D-CNN + Gaussian Mixture Variational AE (GMVAE)
SDFVAE	• Be capable of explicitly learning the representations of time-invariant and time-varying characteristics.	Does normalization	CNN + RNN + VAE
InterFusion	• Employs a hierarchical VAE (HVAE) to learn different features independently. • Learns both low-dimensional inter-metric and temporal embeddings.	Uses zero to fill in missing values, remove extreme values and does standardization.	1D-CNN + RNN+ VAE
JumpStarter	• Clusters univariate time series in MTS. • Reconstructs MTS based on compressed sensing. • Effectively reduces initialization time.	Does normalization.	Clustering + Compressed Sensing
GDN	• Uses an attention-based GNN to learn the inter-metric dependence.	Uses mean values or zero to fill in missing values and does normalization.	Attention + GNN

metric for classification tasks, where True Positives (TP), False Positives (FP), and False Negatives (FN) are taken into account. It is given by: $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}, \text{Recall} = \frac{\text{TP}}{\text{TP+FN}}, F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}}$.

Early anomaly detection algorithms primarily focus on detecting anomaly points. In point-wise F_1, TP represents correctly detected anomalous points, FP represents normal data points incorrectly reported as anomalies, and FN represents anomalous data points overlooked by the algorithm. However, anomalies often occur continuously in practical scenarios and form contiguous segments. It is acceptable for an alert to be triggered in any subset of a ground truth anomaly segment. Thus, a point-adjusted (range-based) evaluation metric is proposed [26]. The entire anomalous segment is deemed correctly detected if any point in a ground truth anomaly segment is detected. In our study, we employ the point-adjusted F_1 to evaluate the performance of the algorithms. We directly count the TP, FP, and FN to calculate the F_1 for each instance. For a dataset, we aggregate the TP, FP, and FN from all entities in the dataset and compute the overall F_1.

IV. Practical Investigation

To address RQ2, we study how existing algorithms work in practice (D1 and D2) and what problems exist.

A. Current Practice and Case Study

The companies we collaborate with use an unsupervised anomaly detection model for all instances in the same service due to the high cost of acquiring a sufficient amount of labeled data and training individual models for each instance. Engineers consistently enhance the model by adjusting hyperparameters and modifying the structure based on practical feedback. In large-scale services, a considerable number of system instances exist, each demonstrating different MTS characteristics and potentially encountering distinct anomalies. Employing one model for all entities can be error-prone and may not yield optimal results. However, selecting the appropriate algorithms for each system instance poses a significant challenge, frustrating engineers.

TABLE III: The performance of different anomaly detection algorithms on D1 and D2. (* denotes an algorithm that performs better on the corresponding dataset, and bold denotes the best algorithm on each dataset.)

Model	D1	D2
DAGMM	0.6333*	0.4106
USAD	0.4982	**0.9051***
OmniAnomaly	0.4270	0.4375*
DOMI	0.5740*	0.1970
SDFVAE	0.6507	0.8886*
InterFusion	0.6369	0.7586*
JumpStarter	0.6420*	0.2119
GDN	**0.7394**	0.7464*

To get a preliminary understanding of the effectiveness of these MTS anomaly detection models, we perform a case study using two real datasets: D1 and D2. In contrast to the practical strategy adopted by the companies mentioned above, we train a dedicated model for each system instance. These datasets contain sufficient training data for each instance, enabling us to evaluate the models' performance optimally under experimental conditions. It is worth noting that training a separate model for each instance can be time-consuming due to the algorithm's training efficiency. In our study, we invest approximately three days to train all the models on D1 and 15 hours on D2.

Table III presents the results of our case study. Overall, the performance of the examined models is unsatisfactory, indicating potential challenges in achieving effective anomaly detection performance in practical scenarios. Specifically, these models perform unstable across different datasets. DAGMM, DOMI, and JumpStarter show better performance on D1, while the other five models perform better on D2. Furthermore, Figure 2 presents the results of the eight algorithms on each instance. No algorithm consistently achieves optimal performance across all entities. For example, while USAD performs well on D2 with a score of 90.51%, it does not achieve the best performance for all entities. InterFusion yields the best result on the 12th instance, whereas OmniAnomaly performs best on the 25th. The algorithms exhibit varying performance on different entities in the same dataset, even when using the same hyperparameters. This indicates that the specific characteristics and types of anomalies present in each instance can impact the effectiveness of the algorithms. In conclusion, none of the existing models always provide optimal results in practical scenarios. The application of unsupervised anomaly detection algorithms remains a challenging task, and it is crucial to carefully select the appropriate algorithm for each instance to achieve the best performance.

B. MTS Characteristics and Anomaly Types

As discussed in § II-B2 and § II-B3, MTS collected from practical scenarios exhibit diverse data characteristics and anomaly types. Drawing on previous research and our practical experience from analyzing over 10,000 historical MTS segments with anomaly labels, we summarize the most valuable data characteristics and types of anomalies in anomaly detection. Then, we conduct a statistical analysis of the collected MTS data to get deeper insights into them.

We identify three key characteristics that significantly impact the performance of anomaly detection algorithms: smoothness, periodicity, and metric correlation. Smoothness refers to the level of fluctuation between adjacent data points. We use $\frac{\#\ smooth\ metrics}{\#\ metrics}$ to quantify the smoothness of MTS. In practical scenarios, the dynamic changes in business operations and low sampling frequencies can result in frequent fluctuations and unclear trends in the data. It is challenging to determine whether the observed changes conform to normal patterns. Comparing the current data with historical data is a valid strategy for determining whether the current data is anomalous. However, detecting anomalies in non-periodic data poses difficulties due to the lack of reliable historical data. We use $\frac{\#\ periodic\ metrics}{\#\ metrics}$ to quantify the periodicity of MTS. Leveraging metric correlation enables the anomaly detection model to use complementary information from multiple metrics. We quantify the correlation between metrics using $1 - \frac{\#metric\ patterns}{\#\ metrics}$. Specifically, we only consider metrics where the patterns and trends closely align, indicating a strong positive correlation.

We categorize anomalies into six types: global anomalies, contextual anomalies, pattern anomalies, frequency anomalies, trend anomalies, and others. Figure 3 presents examples of different types of anomalies. From a duration perspective, global and contextual anomalies are relatively short-lived, lasting less than one period. Pattern anomalies, frequency anomalies, and trend anomalies often span longer segments equal to or greater than one period. To be more specific, global anomalies represent segments that exhibit extreme values when compared to almost all the remaining time points. Contextual anomalies represent segments with values that deviate from the neighboring time points or differ from corresponding time points in other cycles. Pattern anomalies and frequency anomalies typically occur in periodic metrics. Pattern anomalies refer to a segment with different basic patterns compared to normal patterns. Frequency anomalies are characterized by segments displaying unusual frequency compared to the overall frequency. Trend anomalies refer to segments that significantly deviate from the underlying trend of the time series, *i.e.*, segments that consistently deviate from the mean value. The category "others" encompasses anomalies

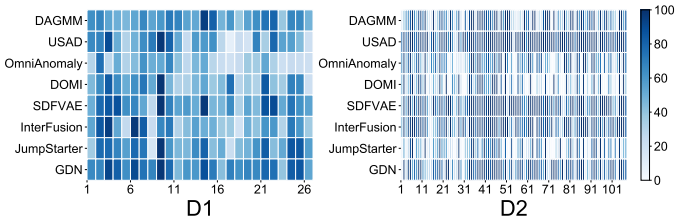

Fig. 2: The performance of anomaly detection algorithms on different entities.

that lack distinct features or clear patterns.

Figure 4 illustrates the diversity of data characteristics across the datasets. Most datasets exhibit a relatively unsmooth characteristic. Only ASD displays a roughly equal distribution between smooth and unsmooth MTS, while D2 has predominantly smooth MTS. Furthermore, even in the same dataset, each MTS exhibits unique characteristics. For example, in D2, approximately 60% of the entities have a smoothness level exceeding 80%. However, the remaining 40% of entities display varying degrees of smoothness, ranging from 26% to 80%. The periodicity of the data also varies across the datasets. SMAP, MSL, and WADI predominantly comprise non-periodic MTS, while D2 exhibits relatively strong periodicity. As for the degree of metric correlation, all datasets except SMAP, SWaT, and WADI exhibit some level of metric correlation. According to Figure 5, we confirm that the anomalous patterns present in MTS exhibit significant diversity. Among the labeled anomalies, the global anomaly is the predominant type across all datasets. However, each dataset has noticeable variations in the proportion of anomaly types. Specifically, global anomalies dominate D2, MSL, SWaT, and WADI anomalies. For D1 and SMAP, there is a more balanced distribution of all types of anomalies. In SMD, both global anomalies and pattern anomalies are dominant. While ASD primarily consists of global, contextual, and pattern anomalies.

V. Experimental and Empirical Analysis

In this section, we aim to answer RQ3 and RQ4. We conduct an experimental study using six publicly available datasets to comprehensively understand existing models across various scenarios. In the following three sections, we evaluate the performance of existing algorithms from three different perspectives: the general case, MTS with different data characteristics, and different types of anomalies. In each section, we also provide recommendations on selecting the appropriate algorithm based on the specific scenario being considered. Lastly, we validate the effectiveness of our recommended algorithms using all the collected data.

A. Overall Performance

Table IV presents the performance of the eight studied algorithms on the public datasets. The experimental results reconfirm the conclusions drawn in §IV-A. Using one algorithm for all datasets is prone to errors. The algorithm outperforming others on one dataset often produces inconsistent results on different datasets. Based on the experimental results, we recommend operators preferentially use SDFVAE, InterFusion, and GDN. These algorithms demonstrate superior performance on most datasets and consistently produce reliable results across all datasets. They can effectively learn inter-metric relationships and temporal dependencies in MTS, enabling them to perform well in MTS anomaly detection tasks.

Table V presents the training and online detection overhead. The training overhead is the average time required for the model to converge on one instance, while the online detection overhead is the average time taken to detect anomalies either in a time window or at a data point. DAGMM, OmniAnomaly, and InterFusion have relatively high overheads due to their complex structures and a larger number of parameters. On the other hand, models like USAD, SDFVAE, DOMI, and GDN demonstrate relatively low training overhead. USAD employs a simple AE structure and adversarial training technique, requiring only a few training epochs. SDFVAE, DOMI, and

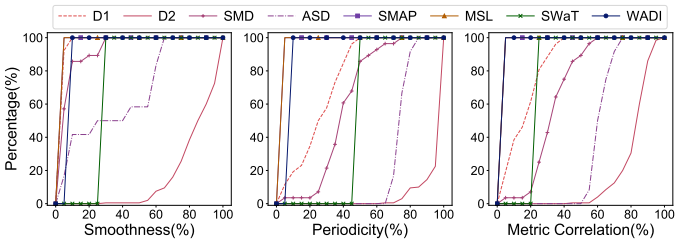

Fig. 4: The cumulative distribution function of different features.

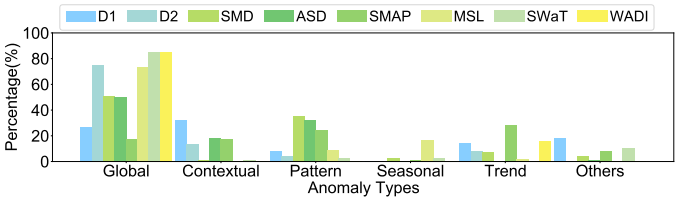

Fig. 5: The distribution of anomaly segments.

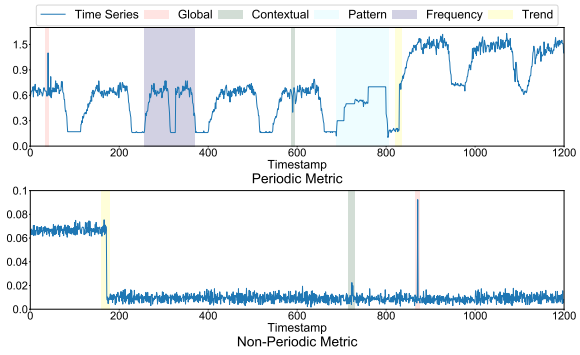

Fig. 3: Common anomaly patterns.

TABLE IV: The overall F_1 of the studied algorithms. (Underline denotes the algorithm that performs worst on the dataset, and bold denotes the algorithm that performs best on the dataset.)

	SMD	ASD	SMAP	MSL	SWaT	WADI
DAGMM	0.9492	0.8615	0.9098	0.9433	0.8663	0.6952
USAD	<u>0.9038</u>	**0.9125**	**0.9812**	0.9471	0.8336	0.4129
OmniAnomaly	**0.9748**	0.8751	0.9402	0.9202	<u>0.6147</u>	0.7101
DOMI	0.9141	<u>0.5350</u>	0.9299	0.9550	**0.9422**	<u>0.1993</u>
SDFVAE	0.9365	0.9087	0.9016	0.9365	0.9052	0.8851
InterFusion	0.9601	0.9101	0.9580	**0.9611**	0.8949	0.8999
JumpStarter	0.9233	0.7001	<u>0.7540</u>	<u>0.8451</u>	0.8694	0.8012
GDN	0.9494	0.8968	0.9380	0.9093	0.8463	**0.9258**

GDN require a small number of parameters, resulting in low training overhead. JumpStarter, as a compressed sensing-based algorithm, does not require a training stage and can directly detect anomalies in real time. Consequently, it takes longer for online detection. However, the detection overhead of JumpStarter is acceptable as it is significantly shorter than the time required to collect the data. JumpStarter is recommended when there is insufficient training data or resource constraints for training detection models. For situations where detection resources are limited or detection speed is crucial, we suggest using USAD, which only requires 0.02ms to 0.03ms for detection.

Considering the balance between effectiveness and efficiency, we recommend prioritizing the GDN and SDFVAE algorithms. These algorithms consistently yield reliable results across all datasets while maintaining low training and detection overhead. InterFusion demonstrates satisfactory overall performance and even outperforms SDFVAE or GDN on certain datasets. However, the training cost of InterFusion is relatively high. On the other hand, the remaining algorithms show poor performance on one or more datasets. Therefore, we do not recommend directly using these models without analyzing the data characteristics or the specific anomaly types.

B. Performance on Various MTS

We conduct three experiments to explore the performance of the algorithms in the presence of MTS with different characteristics. The MTS is divided into ten groups with an interval of 0.1. Table VI presents the results. Due to the limited number of MTS in the public datasets, the coverage of data characteristics is limited.

A lower degree of smoothness indicates more noise and greater fluctuations between adjacent data points. Consequently, the model must possess strong anti-noise capabilities to effectively handle MTS data with low smoothness (ranging from 0 to 0.5). For such cases, we recommend employing the following algorithms: DAGMM, SDFVAE, InterFusion, and GDN. The smoothness is typically observed in the time series. DAGMM learns from and makes predictions on individual time points, so whether MTS is smooth does not directly impact its learning and prediction process. SDFVAE and InterFusion explicitly factorize inter-metric relationships in the MTS, making them resilient to fluctuations and variations. GDN adopts a simple moving average (SMA) [27] method to generate smoothed anomaly scores, helping to prevent normal data fluctuations from being erroneously identified as anomalies. When dealing with MTS characterized by high smoothness (ranging from 0.5 to 0.7), we recommend using USAD, OmniAnomaly, SDFVAE, and InterFusion. These models utilize the reconstruction error for anomaly detection. When the data smoothness is high, the noise in the data is relatively small. In such cases, the reconstruction error can accurately reflect anomalies.

A lower degree of periodicity indicates fewer metrics exhibiting periodic patterns in the MTS, resulting in more intricate patterns across the MTS. The model must possess strong abilities to learn complex patterns and perform few-shot learning. When the data exhibits a low degree of periodicity (ranging from 0 to 0.5), we recommend utilizing DAGMM, SDFVAE, InterFusion, and GDN. DAGMM focuses on capturing the patterns of individual data points, allowing it to effectively identify anomalies based on learned patterns. In cases where MTS lacks periodicity, it becomes challenging to use patterns learned from historical time series to assist in anomaly detection. SDFVAE, InterFusion, and GDN leverage the inter-metric relationships to achieve robust performance even in the presence of data with low levels of periodicity. When the data demonstrate a high degree of periodicity (ranging from 0.5 to 0.9), we recommend utilizing the following models: DAGMM, USAD, OmniAnomaly, and SDFVAE. These models leverage historical data to make accurate predictions for newly-coming data, particularly in scenarios where the data exhibits a more obvious periodic pattern.

A lower metric correlation indicates the presence of more unique univariate time series patterns in the MTS. The model must possess strong abilities to learn complex patterns. When the degree of metric correlation is low (ranging from 0 to 0.5), we recommend employing DAGMM, InterFusion, and GDN. DAGMM is particularly suitable as it focuses on capturing patterns in individual data points, and the number of complex patterns to be learned is low. Moreover, DAGMM utilizes a fully connected layer to learn relationships between all metrics, avoiding incomplete consideration of metric relationships. Despite weak correlations between metrics, InterFusion still demonstrates strong performance by effectively modeling both temporal and inter-metric dependencies. InterFusion adopts a sequential learning strategy to capture both temporal and inter-metric features. By filtering out most of the temporal anomalies in the first step, InterFusion enhances the clarity of inter-metric relationships. GDN leverages an attention structure to learn the relationships between metrics. In cases where the relationships between metrics are not particularly strong, GDN tends to focus on each metric independently and utilize historical data to make forecasts. When the degree of metric correlation is high (ranging from 0.5 to 0.8), we recommend utilizing USAD, OmniAnomaly, and SDFVAE. This high degree of correlation facilitates the detection of anomalies because related metrics tend to change simultaneously or in a close interval when anomalies occur. These models can accurately capture anomalies in data with a high degree of correlation.

C. Performance on Different Anomaly Types

We categorize the detection results based on the type of anomalies present in the MTS. Specifically, when computing the F_1 for the global anomalies, we focus on whether the global anomalies in the MTS are detected, disregarding other types of anomalies. The detailed results can be found in Table VII.

The performance of most models is superior in detecting global anomalies. Except for OmniAnomaly, which achieves

TABLE V: The training time (T) and online detection time (D) of the studied algorithms.

	SMD		ASD		SMAP		MSL		SWaT		WADI	
	T (s)	D (ms)	T (s)	D (ms)	T (s)	D (ms)	T (s)	D (ms)	T (s)	D (ms)	T (s)	D (ms)
DAGMM	709.91	0.95	396.25	0.94	413.80	1.16	707.10	0.91	4403.37	1.17	4520.62	1.15
USAD	620.34	0.02	75.99	0.03	60.33	0.02	83.74	0.02	10352.00	0.03	47457.65	0.03
OmniAnomaly	599.57	0.35	356.75	0.41	320.67	0.35	348.82	0.35	4208.06	0.37	5108.29	0.38
DOMI	9.92	0.31	2.16	0.25	1.03	0.28	2.09	0.44	112.41	0.38	342.83	0.98
SDFVAE	101.59	0.10	117.67	0.03	73.98	0.03	71.82	0.03	1070.42	0.03	2252.28	0.13
InterFusion	2988.92	21.93	906.78	19.60	572.00	23.22	299.92	34.59	55919.08	25.25	118290.53	22.33
JumpStarter	–	4.60	–	2.33	–	3.00	–	6.65	–	6.6	–	16.69
GDN	178.42	0.13	30.78	0.14	32.99	0.25	30.38	0.12	855.38	0.28	1220.14	0.22

TABLE VI: The performance of the studied algorithms on various MTS.

(a) The performance on MTS with different degrees of smoothness.

	0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
DAGMM	0.9029	1.0000	0.8647	–	0.8571	0.9577	0.8406	–	–	–
USAD	0.9144	1.0000	0.8385	–	0.8263	0.9585	0.9652	–	–	–
OmniAnomaly	0.9245	1.0000	0.6244	–	0.8589	0.9252	0.9702	–	–	–
DOMI	0.8799	0.9846	0.9423	–	0.3819	0.5424	0.6868	–	–	–
SDFVAE	0.8928	1.0000	0.9079	–	0.8398	0.9472	0.9599	–	–	–
InterFusion	0.9527	1.0000	0.8975	–	0.8942	0.9513	0.9265	–	–	–
JumpStarter	0.8009	0.9846	0.8709	–	0.5379	0.8574	0.8166	–	–	–
GDN	0.9364	0.9950	0.8514	–	0.8364	0.8416	0.9542	–	–	–

(b) The performance on MTS with different degrees of periodicity.

	0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
DAGMM	0.8842	–	0.9451	0.9437	0.8726	0.9911	0.9297	0.9067	0.8329	–
USAD	0.9208	–	0.9269	0.9186	0.8281	0.9839	0.9489	0.9401	0.8883	–
OmniAnomaly	0.9090	–	0.9719	0.9796	0.6360	0.9989	0.9251	0.8927	0.8906	–
DOMI	0.8736	–	0.9588	0.8956	0.9353	0.9819	0.8584	0.6618	0.3291	–
SDFVAE	0.8808	–	0.9793	0.8950	0.9091	0.9988	0.8983	0.9370	0.8642	–
InterFusion	0.9499	–	0.9573	0.9678	0.8982	0.9901	0.9485	0.9455	0.7787	–
JumpStarter	0.7674	–	0.8670	0.9541	0.8722	0.9557	0.8452	0.8214	0.3478	–
GDN	0.9330	–	0.9478	0.9428	0.8552	0.9772	0.9066	0.9334	0.7545	–

(c) The performance on MTS with different degrees of metric correlation.

	0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	0.4-0.5	0.5-0.6	0.6-0.7	0.7-0.8	0.8-0.9	0.9-1.0
DAGMM	0.8842	0.9655	0.8759	0.9450	0.9645	0.9680	0.8452	0.8329	–	–
USAD	0.9208	0.9399	0.8515	0.8692	0.7714	0.9722	0.9296	0.8883	–	–
OmniAnomaly	0.9090	0.9951	0.6500	0.9735	0.9939	0.9489	0.8855	0.8906	–	–
DOMI	0.8736	0.9847	0.9402	0.8704	0.8705	0.8270	0.6886	0.3291	–	–
SDFVAE	0.9037	0.9879	0.9136	0.8510	0.9835	0.9606	0.9405	0.8642	–	–
InterFusion	0.9499	0.9902	0.9056	0.9503	0.9326	0.9810	0.8980	0.7787	–	–
JumpStarter	0.7674	0.8626	0.8761	0.9551	0.9028	0.9123	0.7942	0.3478	–	–
GDN	0.9330	0.9742	0.8615	0.9180	0.9805	0.9428	0.9552	0.7545	–	–

TABLE VII: The performance of the studied algorithms on various anomaly types.

	Global	Contextual	Pattern	Frequency	Trend	Others
DAGMM	0.8977	0.6325	0.2703	0.7015	0.8491	0.2402
USAD	0.8685	0.7712	0.5107	0.4802	0.2881	0.2012
OmniAnomaly	0.7088	0.2580	0.9672	0.1109	0.8050	0.1857
DOMI	0.8667	0.5669	0.9009	**0.8092**	0.2661	0.2886
SDFVAE	0.8939	0.5395	**0.9731**	0.6596	**0.9355**	**0.3654**
InterFusion	**0.9432**	0.2719	0.9536	0.7533	0.8759	0.2588
JumpStarter	0.8564	0.5872	0.8032	0.5080	0.6694	0.2167
GDN	0.8828	**0.9646**	0.9208	0.7276	0.8754	0.2747

an F_1 of 0.7088, all other models achieve an F_1 of 0.85 or higher. Notably, InterFusion demonstrates the best performance, with a score of 0.9424. Global anomalies are generally easily detected due to obvious deviations from normal values. They can be identified using a relatively short window of data. Even if these anomalies persist beyond the window size, they can be detected once the anomalies occur. InterFusion leverages contextual information in the window data and extracts features separately for each metric during the temporal feature extraction stage, mitigating interference between metrics. These factors contribute to the excellent performance of InterFusion in detecting global anomalies.

The detection performance for contextual anomalies varies across the studied algorithms, ranging from 0.25 to 0.96. GDN achieves the highest performance with an impressive F_1 of 0.9646. It is often challenging to determine contextual anomalies by comparing data at a single time point with the data at

surrounding context time points. Contextual anomaly detection heavily relies on the model's ability to learn the temporal relationships and inter-metric correlations in the historical MTS data. GDN excels in modeling and detecting MTS with short windows. It can effectively leverage the historical MTS data that exhibit similar patterns by utilizing the window data. Short windows allow GDN to focus more on capturing the differences with historical MTS rather than solely relying on the information in a single window. Furthermore, GDN models the relationships between metrics. It is adept at capturing these correlation changes between metrics, enabling it to effectively detect anomalies in the data.

Five of the eight algorithms achieve a satisfactory F_1 of 0.9 or higher in detecting pattern anomalies. Pattern anomalies are generally easily detectable due to their long durations and significant deviations from historical patterns. When a pattern change occurs, there are simultaneous alterations in the temporal relationships between adjacent points and the relationships between metrics. Among the algorithms, SDFVAE stands out with an impressive F_1 of 0.9731. SDFVAE is particularly well-suited for detecting pattern anomalies because it can effectively handle long-term time series data. Additionally, SDFVAE explicitly models both temporal and inter-metric relationships, enabling it to capture the pattern anomalies.

The detection performance of frequency anomalies varies across the studied algorithms, ranging from 0.11 to 0.81. DOMI performs the best and significantly outperforms the other algorithms with an F_1 of 0.8092. Distinguishing frequency anomalies from normal data is often challenging when only relying on contextual data and the relationships between metrics. Moreover, in real-world MTS, multiple normal patterns are typically present, and different normal window data may exhibit distinct patterns in the same MTS. DOMI addresses these challenges by incorporating a categorical variable to obtain robust data features. This method allows DOMI to divide the MTS into finer categories, enabling more precise anomaly detection ability.

Four models perform strongly in detecting trend anomalies, with SDFVAE scoring 0.9355. Detecting trend anomalies is also challenging when relying solely on contextual data and the relationships between metrics. Fortunately, trend anomalies exhibit distinct mean values that deviate from the mean value observed in normal data. Deep learning-based models are inherently sensitive to numerical changes, making them well-suited for detecting trend anomalies. The ability to capture and analyze long-term data enables SDFVAE to excel in detecting trend anomalies in the MTS data.

The overall performance of the studied algorithms in detecting anomalies in the "others" type is relatively low. Even the best-performing algorithm, SDFVAE, achieves a score of only 0.3654. This is because anomalies in this category are usually labeled based on true backtracking and do not have clear changes in the data. Detecting anomalies in this category remains a challenging task, and further research is needed to improve anomaly detection performance in identifying anomalies in this specific category.

D. Performance of Recommended Algorithms

In the previous three sections, we provide recommendation algorithms for different scenarios based on the performance of the algorithms on public datasets. In this section, we evaluate the effectiveness of recommended algorithms in various scenarios using all the collected MTS data. Specifically, when we select algorithms based on the data characteristics for each instance, we use the corresponding recommendation model. If multiple models are recommended, we use the model that performs best in the characteristic value interval of the instance. Table VI provides the performance of each algorithm for different features. Please note that when selecting algorithms based on Table VI, we only consider recommendation algorithms that match the data characteristics. When applying algorithms based on the types of anomalies for each instance, we usually utilize multiple algorithms since an instance can have multiple types of anomalies. If any algorithms detect an anomaly, the data point will be reported as an anomaly. The experimental results are shown in Table VIII.

The recommended SDFVAE, InterFusion, and GDN all demonstrate outstanding anomaly detection performance that meets the performance requirements and are among the top three algorithms. Regarding reduced model training overhead and detection overhead, our recommended algorithms, JumpStarter and USAD, achieved scores of 0.7855 and 0.8269, respectively. Although these results may not be ideal, they are still acceptable considering the resource constraints. For a balance between effectiveness and efficiency, we recommend SDFVAE and GDN. They offer satisfactory performance while maintaining low training and detection overhead.

InterFusion stands out with the highest score of 0.8878 when utilizing a single algorithm. However, our data characteristics-based solutions consistently outperform the best results achieved using a single algorithm, with performance gains of up to 1.3%. Furthermore, it is worth noting that InterFusion, despite its high performance, has the highest training and testing overhead among all the algorithms. Our recommended solution effectively addresses this overhead concern by combining multiple algorithms with lower overhead. Overall, our recommended solution offers significant advantages in performance improvement and cost savings.

Our anomaly types-based solution achieves a score of 0.8624 on all MTS data, slightly lower than the best single algorithm but still better than most. Upon analysis, we observed a significant improvement in the recall of anomalies using our recommended algorithms. However, there is a decrease in precision. This decrease in precision can be attributed to the aggregation of detection results from multiple models for a single instance, resulting in a higher number of false positives. Nevertheless, we firmly believe that algorithm recommendation based on specific anomaly types remains meaningful in practical applications. This is because different systems and services typically face a limited number of attack and failure types. Operators can easily utilize our recommended models based on their practical application without validating

TABLE VIII: The performance of using the recommended algorithms.

Model/Strategy	F_1
DAGMM	0.8499
USAD	0.8269
OmniAnomely	0.7380
DOMI	0.8372
SDFVAE	0.8657
InterFusion	0.8878
JumpStarter	0.7855
GDN	0.8823
Smoothness	0.9008
Periodicity	0.8906
Metric correlation	0.8889
Anomaly types	0.8624

all algorithms, streamlining the process.

Overall, our recommended solution is highly practical and significantly reduces the workload for operators while ensuring effective and efficient anomaly detection. In cases where all data characteristics and anomaly types are known, we suggest that operators prioritize the selection of algorithms based on data smoothness and finally consider the specific anomaly types.

VI. Threats to validity

As an empirical study, our research is subject to various common threats that can impact the validity and reliability of the results. These threats encompass several factors, including the datasets, the models used, the evaluation metrics employed, and the methods used for inspection and analysis.

Datastes. Despite our efforts to collect publicly available datasets and real-world data from two partner companies, the scenarios and volume of collected MTS remain constrained, which may impact the validity and generalizability of our empirical conclusions. Fortunately, the datasets used in our study are derived from real-world scenarios or sophisticated testbeds, and they have been meticulously labeled by experts based on fault feedback. Furthermore, the gained insights have been applied to all MTS, resulting in satisfactory performance. While the number of MTS is limited, we have a high level of confidence in the effectiveness of the data and our conclusions.

Studied models. We focus on evaluating the performance of eight unsupervised algorithms in this work. It is important to note that numerous MTS anomaly detection algorithms still need to be evaluated. Moreover, we maintain default hyperparameter settings for each algorithm when testing them on different datasets. However, different algorithms often use different hyperparameter configurations, even if the hyperparameter has similar meanings. These hyperparameters can significantly impact the performance of the algorithms. While using default hyperparameters is still an acceptable method, as each model has its unique structure, employing different hyperparameter settings may be necessary to achieve optimal performance for them.

Evaluation metrics. We use the widely used metric, F_1, to evaluate the performance of the anomaly detection models. F_1 is considered a suitable metric even when there is an imbalance in the data categories. However, different domains have different preferences for the capabilities of anomaly detection algorithms. Many applications prioritize accurately predicting large portions, detecting anomalies early, and minimizing the number of false alarms. Analyzing these preferences is crucial in specific scenarios. This paper aims to study the general performance of unsupervised algorithms, and therefore, F_1 score is a suitable choice for evaluation.

Inspection methods. To ensure the validity of our conclusions, we make efforts to mitigate the impact of various factors on our experimental results. For efficiency experiments, we conduct them on a dedicated server that allows only one program to run at a time. Moreover, with limited data, we perform separate analyses of the data characteristics and anomaly types. This helps minimize the impact of inadequate experimental data under multiple factors on our experimental results and conclusions.

VII. Conclusion

This empirical study analyzes current practices and provides recommendations for selecting appropriate algorithms in specific scenarios. Our insights successfully address significant practical challenges of applying MTS anomaly detection models in real-world scenarios. Firstly, we examine eight algorithms using two real-world datasets, allowing us to intuitively understand current practices. This analysis also helps us identify common data characteristics and types of anomalies that are crucial for effective anomaly detection. The accurate labels for data characteristics and types of anomalies are now publicly available, which serves as a valuable resource for this study and future research in anomaly detection. Following that, we evaluate the effectiveness and efficiency of these algorithms using publicly available datasets and provide general recommendations for selecting the appropriate algorithms in various scenarios. Lastly, we provide clear guidance on choosing the appropriate algorithm for MTS with different data characteristics and specific anomaly types. We apply the guidance to all the datasets we collected. The results show that most of our suggestions can achieve better than any algorithm alone. Overall, we derive key findings and valuable insights that aim to guide and advance future research in anomaly detection.

Acknowledgment

We thank Fangyuan Liu, Yizhen Zhang, Xijie Pan, and Ziyi Liu for their contributions to this work. The work was supported in part by the National Key Research and Development Program of China (No.2021YFB0300104), Advanced Research Project of China (No.31511010501), National Natural Science Foundation of China (Grant No.62272249, 62072264), and Natural Science Foundation of Tianjin (Grant No.21JCQNJC00180).

REFERENCES

[1] Y. Su, Y. Zhao, W. Xia, R. Liu, J. Bu, J. Zhu, Y. Cao, H. Li, C. Niu, Y. Zhang, Z. Wang, and D. Pei, "Coflux: robustly correlating kpis by fluctuations for service troubleshooting," in *Proceedings of the International Symposium on Quality of Service, IWQoS 2019, Phoenix, AZ, USA, June 24-25, 2019*. ACM, 2019, pp. 13:1–13:10. [Online]. Available: https://doi.org/10.1145/3326285.3329048

[2] Y. Lee, D. Juan, X. Tseng, Y. Chen, and S. Chang, "Dc-prophet: Predicting catastrophic machine failures in datacenters," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18-22, 2017, Proceedings, Part III*, ser. Lecture Notes in Computer Science, Y. Altun, K. Das, T. Mielikäinen, D. Malerba, J. Stefanowski, J. Read, M. Zitnik, M. Ceci, and S. Dzeroski, Eds., vol. 10536. Springer, 2017, pp. 64–76. [Online]. Available: https://doi.org/10.1007/978-3-319-71273-4_6

[3] X. Wang, J. Lin, N. Patel, and M. W. Braun, "Exact variable-length anomaly detection algorithm for univariate and multivariate time series," *Data Min. Knowl. Discov.*, vol. 32, no. 6, pp. 1806–1844, 2018. [Online]. Available: https://doi.org/10.1007/s10618-018-0569-7

[4] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S. Ng, "MAD-GAN: multivariate anomaly detection for time series data with generative adversarial networks," in *Artificial Neural Networks and Machine Learning - ICANN 2019: Text and Time Series - 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17-19, 2019, Proceedings, Part IV*, ser. Lecture Notes in Computer Science, I. V. Tetko, V. Kurková, P. Karpov, and F. J. Theis, Eds., vol. 11730. Springer, 2019, pp. 703–716. [Online]. Available: https://doi.org/10.1007/978-3-030-30490-4_56

[5] G. G. González, P. Casas, A. Fernández, and G. Gómez, "On the usage of generative models for network anomaly detection in multivariate time-series," *SIGMETRICS Perform. Evaluation Rev.*, vol. 48, no. 4, pp. 49–52, 2021. [Online]. Available: https://doi.org/10.1145/3466826.3466843

[6] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, and N. V. Chawla, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 1409–1416. [Online]. Available: https://doi.org/10.1609/aaai.v33i01.33011409

[7] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. [Online]. Available: https://openreview.net/forum?id=BJJLHbb0-

[8] W. Liao, Y. Guo, X. Chen, and P. Li, "A unified unsupervised gaussian mixture variational autoencoder for high dimensional outlier detection," in *IEEE International Conference on Big Data (IEEE BigData 2018), Seattle, WA, USA, December 10-13, 2018*, N. Abe, H. Liu, C. Pu, X. Hu, N. K. Ahmed, M. Qiao, Y. Song, D. Kossmann, B. Liu, K. Lee, J. Tang, J. He, and J. S. Saltz, Eds. IEEE, 2018, pp. 1208–1217. [Online]. Available: https://doi.org/10.1109/BigData.2018.8622120

[9] Y. Su, Y. Zhao, M. Sun, S. Zhang, X. Wen, Y. Zhang, X. Liu, X. Liu, J. Tang, W. Wu, and D. Pei, "Detecting outlier machine instances through gaussian mixture variational autoencoder with one dimensional CNN," *IEEE Trans. Computers*, vol. 71, no. 4, pp. 892–905, 2022. [Online]. Available: https://doi.org/10.1109/TC.2021.3065073

[10] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano, "A review on outlier/anomaly detection in time series data," *ACM Comput. Surv.*, vol. 54, no. 3, pp. 56:1–56:33, 2022. [Online]. Available: https://doi.org/10.1145/3444690

[11] D. Xu, W. Cheng, J. Ni, D. Luo, M. Natsumeda, D. Song, B. Zong, H. Chen, and X. Zhang, "Deep multi-instance contrastive learning with dual attention for anomaly precursor detection," in *Proceedings of the 2021 SIAM International Conference on Data Mining, SDM 2021, Virtual Event, April 29 - May 1, 2021*, C. Demeniconi and I. Davidson, Eds. SIAM, 2021, pp. 91–99. [Online]. Available: https://doi.org/10.1137/1.9781611976700.11

[12] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: unsupervised anomaly detection on multivariate time series," in *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, R. Gupta, Y. Liu, J. Tang, and B. A. Prakash, Eds. ACM, 2020, pp. 3395–3404. [Online]. Available: https://doi.org/10.1145/3394486.3403392

[13] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "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, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds. ACM, 2019, pp. 2828–2837. [Online]. Available: https://doi.org/10.1145/3292500.3330672

[14] L. Dai, T. Lin, C. Liu, B. Jiang, Y. Liu, Z. Xu, and Z. Zhang, "SDFVAE: static and dynamic factorized VAE for anomaly detection of multivariate CDN kpis," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*, J. Leskovec, M. Grobelnik, M. Najork, J. Tang, and L. Zia, Eds. ACM / IW3C2, 2021, pp. 3076–3086. [Online]. Available: https://doi.org/10.1145/3442381.3450013

[15] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 4027–4035. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/16523

[16] Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, and D. Pei, "Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding," in *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, F. Zhu, B. C. Ooi, and C. Miao, Eds. ACM, 2021, pp. 3220–3230. [Online]. Available: https://doi.org/10.1145/3447548.3467075

[17] H. Liang, L. Song, J. Wang, L. Guo, X. Li, and J. Liang, "Robust unsupervised anomaly detection via multi-time scale dcgans with forgetting mechanism for industrial multivariate time series," *Neurocomputing*, vol. 423, pp. 444–462, 2021. [Online]. Available: https://doi.org/10.1016/j.neucom.2020.10.084

[18] V. Hautamäki, I. Kärkkäinen, and P. Fränti, "Outlier detection using k-nearest neighbour graph," in *17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, August 23-26, 2004*. IEEE Computer Society, 2004, pp. 430–433. [Online]. Available: https://doi.org/10.1109/ICPR.2004.1334558

[19] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, no. 9-10, pp. 1641–1650, 2003. [Online]. Available: https://doi.org/10.1016/S0167-8655(03)00003-5

[20] L. M. Manevitz and M. Yousef, "One-class svms for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, 2001. [Online]. Available: http://jmlr.org/papers/v2/manevitz01a.html

[21] M. Ma, S. Zhang, J. Chen, J. Xu, H. Li, Y. Lin, X. Nie, B. Zhou, Y. Wang, and D. Pei, "Jump-starting multivariate time series anomaly detection for online service systems," in *2021 USENIX Annual Technical Conference, USENIX ATC 2021, July 14-16, 2021*, I. Calciu and G. Kuenning, Eds. USENIX Association, 2021, pp. 413–426. [Online]. Available: https://www.usenix.org/conference/atc21/presentation/ma

[22] 2023. [Online]. Available: https://github.com/ldwen/MTS-Checking-Tool-Data-Result

[23] 2023. [Online]. Available: https://github.com/ldwen/MTS_Data

[24] G. Pang, C. Shen, and A. van den Hengel, "Deep anomaly detection with deviation networks," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, and G. Karypis, Eds. ACM, 2019, pp. 353–362. [Online]. Available: https://doi.org/10.1145/3292500.3330871

[25] M. E. Villa-Pérez, M. Á. Á. Carmona, O. Loyola-González, M. A. Medina-Pérez, J. C. Velazco-Rossell, and K. R. Choo, "Semi-supervised anomaly detection algorithms: A comparative summary and future research directions," *Knowl. Based Syst.*, vol. 218, p. 106878, 2021. [Online]. Available: https://doi.org/10.1016/j.knosys.2021.106878

[26] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, and H. Qiao, "Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds. ACM, 2018, pp. 187–196. [Online]. Available: https://doi.org/10.1145/3178876.3185996

[27] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Söderström, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Y. Guo and F. Farooq, Eds. ACM, 2018, pp. 387–395. [Online]. Available: https://doi.org/10.1145/3219819.3219845