TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 10/39 pp823–836
DOI: 10.26599/TST.2024.9010256
Volume 31, Number 2, April 2026

Research Article

Spectra: Spatial-Temporal Parallel Memory with Agent Attention Fusion and Embedding Alignment for Time-Series Anomaly Detection

Muyan Yao, Dan Tao*, Peng Qi, and Ruipeng Gao

Abstract: Early detection of anomalous events in automated processes within industrial scenarios helps to improve service smoothness, thus becoming critical and urgent. Despite this vision, prior works face challenges in convergence on noisy training materials and insufficient construction of spatial-temporal dependencies, leading to performance limitations. In this work, we propose Spectra, a flexible framework for time-series anomaly detection in industrial scenarios. We employ a pair of parallel memory modules in the generative model to store and purify spatial and temporal knowledge in latent embeddings. As such, Spectra offsets the impact of noise and anomalous components in training materials, and signifies the difference between normals and anomalies. To dynamically integrate cross-domain information, we design an embedding fusion mechanism that comprises an agent attention module and a contrastive embedding alignment technique. This mechanism bridges embeddings from instantiated memory modules, aligns dependencies, and improves the organization of the latent space. Extensive experiments on three large-scale industrial datasets demonstrate Spectra's effectiveness, with an average F1-Score of 0.9083 outperforming the baselines.

Key words: Anomaly Detection (AD); parallel memory; multi-head agent attention; spatial-temporal; contrastive regulation; embedding alignment; Internet of Things (IoTs)

1 Introduction

Recently, the ever-changing landscape of industries has urged the adoption of Internet-of-Things (IoT) technologies[1, 2], enabling various automated processes. Their deployment brings an escalated efficiency in multiple applications, including human-machine interaction[3], optical communication[4], healthcare[5], digital twin[6], and smart city[7, 8]. However, they also contribute to challenges in ensuring the services' smoothness[9, 10], integrity[11, 12], and maintenance[13–15].

Anomaly Detection (AD) solutions aim to identify unusual events in the target time-series data to reduce service downtime and maintenance costs. These solutions consistently check the behavior of concerned assets, enhancing the reliability and efficiency of infrastructures in the production environment[16–18].

Despite the goal, deploying these solutions in practical applications is not straightforward. In industrial scenarios, the data generated during production processes are usually unlabeled due to the cost of labeling[19]. Further, fluctuations and sensor malfunctions in on-site assets may introduce noise or anomalous components in the target data stream[20].

Prior works use generative models to develop algorithms under such circumstances, capturing normal patterns from unlabeled historical observations and comparing deviations. However, due to the self-

• Muyan Yao, Dan Tao, and Peng Qi are with School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100040, China. E-mail: muyanyao@bjtu.edu.cn; dtao@bjtu.edu.cn; pengqi1@bjtu.edu.cn.
• Ruipeng Gao is with School of Cyberspace Science and Technology, Beijing Jiaotong University, Beijing 100040, China. E-mail: rpgao@bjtu.edu.cn.
* To whom correspondence should be addressed.
Manuscript received: 2024-04-20; revised: 2024-07-15; accepted: 2024-12-21

supervised nature of such models, their training process becomes a balancing act, and the model inadvertently learns to reconstruct the noise or anomalous components in the training set. Otherwise, it would fail to capture the intricate knowledge within the data stream.

These situations contribute to less distinguishable reconstructed results, resulting in less effective anomaly detection. As a result of this dilemma, deploying such solutions in practical applications becomes challenging (see Fig. 1).

Challenge 1: Extracting and memorizing typical patterns from unlabeled training materials is difficult in a self-supervised pipeline. The vast deployment of sensors and actuators has led to an ecosystem enabled by seamless interconnections. The target data stream pattern in the production environment varies with time and context[21, 22]. Besides, the complex topologies presented by the connections and the noisy, anomalous components in the data further aggregate the difficulty of their utilization. As such, it becomes challenging to extract a proper template from unlabeled on-site data to identify undetermined samples.

Challenge 2: Bridging and aligning dependencies from spatial and temporal domains in the latent space is not straightforward. Interconnections between different devices result in dependencies that span spatial and temporal domains. These dependencies provide more detailed information about the target data stream, but their utilization remains challenging. Incorporating both dependencies is essential for identifying anomalies caused by different factors[23]. However, the bridging and aligning dependencies that cross domains remain a significant challenge.

To address these challenges, we propose a flexible anomaly detection framework Spectra, tailored for time-series data in industrial scenarios. Spectra incorporates a pair of parallel memory modules that dynamically regulate the extraction of typical patterns, even from training materials with high levels of noise and anomalous components. Besides, an agent attention enhanced fusion mechanism is employed to integrate and reorganize information from the two memory instances. Considering the embeddings from spatial and temporal domains may feature distinct properties, this mechanism improves the alignment of the dependencies and their cross-domain adaptation. Extensive experiments are conducted to validate Spectra's generalization and efficiency. The results on three real-world large-scale datasets demonstrate that Spectra surpasses the baselines in AD performance and verify its design. The contributions of this work are threefold.

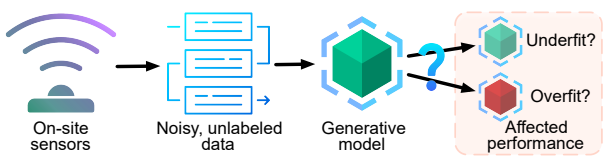

Fig. 1 Challenges associated with deploying AD solutions.

- We propose a parallel memory mechanism that is dynamically updated to bridge the spatial and temporal dependencies and ease the impacts of noise in target materials. In this way, the difference between normals and anomalies gets magnified.
- To integrate knowledge portrayed in different aspects, we design an embedding fusion mechanism that comprises an agent attention module and a contrastive embedding alignment technique. This mechanism helps to reduce redundant components and improve the organization of the latent space.
- We propose Spectra, a novel framework designed for anomaly detection in industrial scenarios. Three large-scale datasets are used for performance validation. Experiment results indicate that Spectra achieves an average F1-Score of 0.9083, outperforming the baselines.

The remainder of this manuscript is organized as follows. We review recent works in Section 2. Section 3 unveils the design of our proposed framework Spectra, and we present the experiments in Section 4. In Section 5, conclusions of this work and possible future directions are given.

2 Related Work

Motivated by the concept of achieving automated device monitoring, researchers from academia and industry have been exploring possible solutions. Prior works have made significant progress in detecting the operational states of deployed devices. Considering their different algorithm implementations, we categorize these solutions into conventional and deep learning based ones.

2.1 Conventional approaches

Given the static nature of task contents in IoT

environments, some researchers use specialized domain knowledge to develop effective approaches for identifying anomalous behaviors of on-site devices[24, 25].

Some research examines the properties of the concerned data stream and analyzes the presence of anomalies. Liu et al.[26] used a smoothed Z-Score algorithm to determine a proper threshold for AD tasks. Researchers in Ref. [27] applied a fusion mechanism composed of K-Medoids and Euclidean Distance to detect anomalies in time series. These handcrafted algorithms heavily rely on data distribution characteristics, limiting their scalability. To create more robust solutions, researchers shift their focus toward machine learning-based approaches.

Kim and Heo[28] selected handicraft-extracted features by calculating their correlation coefficients, and applied a series of machine learning algorithms, e.g., Linear Discriminant Analysis and XGBoost, to spot anomalies in a hydraulic system. Li et al.[29] used Isolation Forest to detect anomalies in power grids. Considering the vulnerabilities of the conventional feature selection process, researchers in Ref. [30] proposed a novel feature selection strategy. Utilizing the selected features, unusual sequences within the target data are thus spotted by a random forest algorithm.

Despite the progress being made, there are still drawbacks to these solutions. Machine learning algorithms require manual specification of feature extraction schemes, which demands prior knowledge in the relevant domain. Moreover, the resulting designs often suffer from poor generalization capabilities. In evolving application landscapes, where distributions and patterns shift over time, sustaining optimal parameter configurations poses significant challenges. This dynamic nature necessitates adaptive approaches to maintain the effectiveness of models in such situations. Otherwise, data distribution and pattern changes may degrade the model's performance.

2.2 Deep learning approaches

Researchers have focused on developing adaptive algorithms and techniques to mitigate the adverse effects of parameter and data pattern changes. Deep learning, in particular, has gained significant attention due to its ability to automatically learn hierarchical representations directly from raw data.

Su et al.[31] introduced a stochastic variable connection approach to generate representations from the samples and detect anomalies based on reconstruction probabilities. This work utilizes techniques including stochastic variable connections and planar flows to capture normal patterns from unlabeled data, providing a potential solution to spot anomalies. Lin et al.[32] employed Long Short-Term Memory (LSTM) modules in the autoencoder structure to construct temporal dependencies and generate cross-timestep embeddings. These researches validate the possibility of utilizing autoencoders to process unlabeled data and identify potential anomalies.

Then, researchers investigate the application of other novel components to elevate AD performance. Since the graph network is designed to capture complex dependencies[33, 34], some works introduce these modules to enhance the representation learning process. Zhao et al.[35] proposed a novel framework that constructs spatial dependencies through graph-attention modules, and employed Gated Recurrent Unit (GRU) modules to depict the target data from the temporal perspective. Li et al.[36] designed a hierarchical variational autoencoder approach to model the target data through two stochastic latent variables. Boniol et al.[37, 38] analyzed the embeddings in the latent space produced from graph modules, and spot anomalies according to the distribution properties. Inspired by adversarial frameworks, some other works aim to change how models are trained. Audibert et al.[19] designed an adversarial framework that shares the same encoder to enhance the separation of normal and anomalous embeddings effectively. Deng et al.[39] further introduced graph and LSTM structures to explicitly capture inter-domain correlations in neighboring data points. Chen et al.[40] introduced a hierarchical attention network to explicitly extract the short-term and periodic patterns and model the non-linear dependencies in the target data. These efforts have led to significant improvements in the overall AD performance. However, the unique characteristics of real-world business scenarios result in challenges that hinder their actual deployment[20].

In industrial scenarios, the properties of data are usually complex and dynamic, and anomalies can be produced in diverse ways. Network fluctuation, packet loss, and drifts in device patterns all lead to anomalous readings. While the proposed approaches demonstrate powerful results during their validations, translating these methods to real-world applications may

encounter difficulties. To solve this problem, some researchers propose memory modules to ease over-expression and pattern shift in industrial data. Zhang et al.[20] used an LSTM based memory network to capture temporal dependencies, and applied a maximum mean discrepancy penalty to guide the generation of embeddings and ease the impacts of noise. Gao et al.[41] designed a network that incorporates memory modules into autoencoder networks to store historical patterns, and rectify the embeddings' components. Additionally, researchers in Ref. [42] used different memory modules to capture global and local information, thus creating rich representations for further data reconstruction. There have also been efforts to introduce log analysis[43] and overhead balance[44] to address the needs of industrial anomaly detection. These works discuss the challenges faced in actual applications, presenting potential solutions to enhance model convergence, robustness, and deployment. Despite the progress that has been made, the balance between a more robust representation extraction and the proper convergence of models remains challenging.

3 Framework and Methodology

3.1 Overview

The overall framework of Spectra is presented in Fig. 2. It consists of two essential components: the generative model with an encoder-decoder structure, and the contrastive spatial-temporal memory module.

The encoder in the generative model processes incoming multivariate sequences to generate embeddings within the latent space. As depicted in the upper left part of Fig. 2, the encoder translates target multivariate sequences into the latent space to extract the typical patterns. This procedure forms two types of embeddings: the spatial embedding, derived from the last 1D Convolution layer (Conv1D), and the temporal embedding, output from the Bidirectional Gated Recurrent Unit (Bi-GRU) Layer.

In industrial environments, the target data are usually collected from a process that involves a set of devices that function cooperatively. As such, the data range and distribution properties of the features within the data stream can be inconsistent. To construct a comprehensive representation and understand the long-term correlations between different metrics, analyzing the spatial and temporal embeddings, and modeling the intricate patterns between different metrics are necessary.

To address this need, we introduce a pair of parallel memory modules to bridge the connections between the spatial and temporal embeddings and ease the impacts caused by noise components in the target samples. They work under the guidance of the embedding fusion mechanism, dynamically update the components in the memory modules, and purify contents in the embeddings.

After obtaining the reconstructed sample from the decoder, Spectra compares the discrepancy between the input observation sample and its reconstruction. Based on this discrepancy, Spectra determines if this sample is anomalous.

3.2 Data pre-processing and problem statement

The widespread deployment of devices in industrial

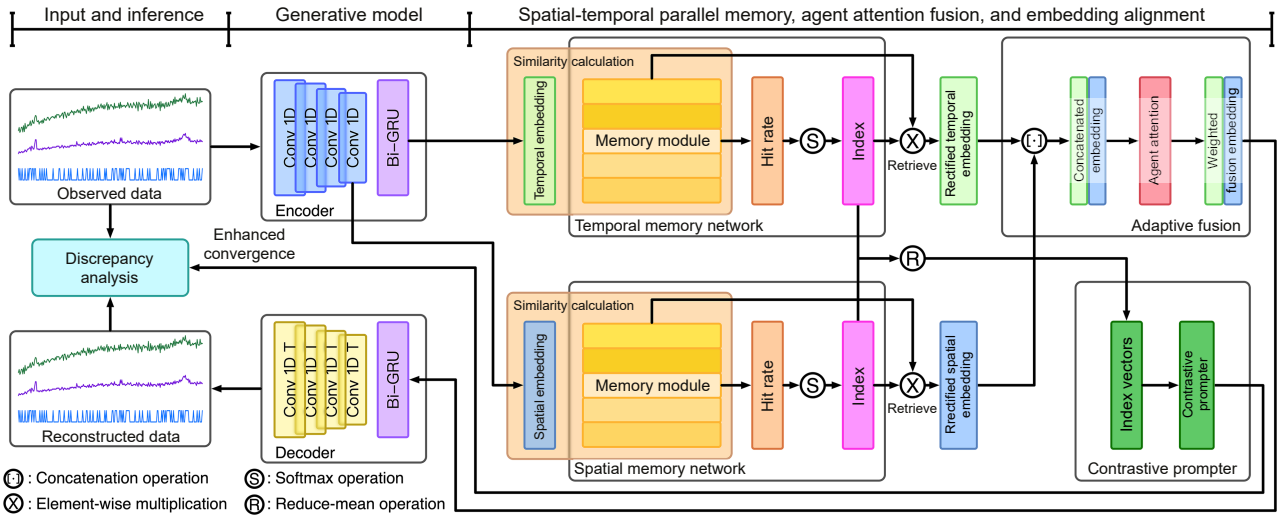

Fig. 2 Framework of spectra.

settings enables a variety of automated processes, enhancing production efficiency on a large scale. These processes are conducted by a set of sensors and actuators, which produce multivariate time series data streams that are collected and stored for further actions. To effectively leverage these data, it is essential to pre-process the raw recordings and ensure they are suitable for subsequent analysis.

Without loss of generality, we denote the 1D multivariate sequence as $\boldsymbol{x} = \{\boldsymbol{x}_{(1)}, \boldsymbol{x}_{(2)}, \ldots, \boldsymbol{x}_{(N)}\}$, $\boldsymbol{x} \in \mathbf{R}^{N \times M}$, where N and M represent the number of timesteps and features in the data, respectively. Each column in the sequence can be denoted as a univariate sequence featuring a size of $\mathbf{R}^{N \times 1}$. Each column stands for one feature space, which comes from readings of one device. A column-wise Max-Min Scaling is performed in both the training and inference process to normalize the target data,

$$\boldsymbol{x} = \frac{\boldsymbol{x} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})} \tag{1}$$

where $\min(\boldsymbol{x})$ and $\max(\boldsymbol{x})$ are the minimum and maximum values of the column being scaled. A sliding window is applied to the scaled data to focus on more detailed information along the time series, forming sequences in segments, i.e., $\boldsymbol{x}_{t-L+1:t} \in \mathbf{R}^{L \times M}$, where L is the window length, which is the number of timesteps contained in a sequence. Since the sliced samples are the minimal units consumed by the deep learning model, in the subsequent sections of this manuscript, we use $\boldsymbol{x}_t$ to represent $\boldsymbol{x}_{t-L+1:t}$ to simplify the notation.

For an AD task, the goal is straightforward: to determine whether the incoming event $\boldsymbol{x}_t$[※] is an anomaly. This process can be denoted as follows:

$$y_t = \mathcal{F}(\boldsymbol{x}_t; \boldsymbol{\beta}_{\text{Spectra}}) \tag{2}$$

where $\mathcal{F}(\cdot)$ represents the mapping function of Spectra, and $\boldsymbol{\beta}_{\text{Spectra}}$ is the parameter set of our proposed model.

3.3 Encoder and embedding generation

In this section, we unveil the encoder's design. The convolutional encoder plays a fundamental role in the analysis and reconstruction process of the incoming sequence $\boldsymbol{x}_t$. This structure is designed to form two different embeddings to benefit the extraction and utilization of both spatial and temporal information. i.e., the spatial and temporal ones.

※For illustrative purposes, we refer to a generic timestep as t without assuming any predefined characteristics for this specific timestep.

To accomplish this objective, we incorporate four cascaded Conv1D layers to bridge spatial dependencies in the target data stream. Besides, a Bi-GRU is added after the Conv1D layers to extract temporal features. We use this combination to formulate a comprehensive understanding of the dynamics embedded in two kinds of embeddings as follows:

$$\boldsymbol{z}_t^{\text{spat}} = \text{Conv1D}(\boldsymbol{x}_t) \times 4 \tag{3}$$

$$\boldsymbol{z}_t^{\text{temp}} = \text{Bi-GRU}(\boldsymbol{z}_t^{\text{spat}}) \tag{4}$$

where E is the size of the embeddings. We use $\boldsymbol{z}_t^{\text{spat}} \in \mathbf{R}^{1 \times E}$ and $\boldsymbol{z}_t^{\text{temp}} \in \mathbf{R}^{1 \times E}$ to denote the spatial embedding and the temporal embedding, which contain correlations that are bridged intra- and inter- timestep, respectively.

In the initial stage of the generative model, a sequence of four Conv1D layers is employed to produce spatial dependencies. Each of these layers performs a Conv1D operation with a stride of 2, to compact the feature maps and create a higher level of abstraction. The cascaded layers' hierarchical feature extraction process allows the model to uncover intricate patterns and dependencies within the sample gradually.

On the other hand, applying the Bi-GRU layer enables the model to leverage temporal relationships within the sample selectively. A Bi-GRU layer operates by processing the input sequence in two directions: forward and backward. This bidirectional nature allows Spectra to consider the context from both past and future timesteps simultaneously. As such, Spectra captures long-range dependencies that span across multiple timesteps. This is particularly beneficial in industrial applications where the input sequences exhibit intricate temporal relationships: the model thus needs to understand the context beyond adjacent timesteps for the proper bridging of these connections.

3.4 Spatial-temporal parallel memory mechanism

As discussed in previous studies[20, 45], autoencoders and their variants often suffer overfitting problems when trained on unlabeled noisy datasets. However, the presence of anomalies or noise is common in industrial data. Consequently, the model trained on these materials may learn to reconstruct these irregular patterns, and make anomalies less distinguishable.

To address this issue, we propose a spatial-temporal

parallel memory mechanism, and integrate it into the structure of the conventional autoencoder. The memory mechanism acts as a cache of the patterns, which helps to identify and remove anomalous components from the embedding vectors. In this way, we ease the impacts of noise and anomalies on the generation of embeddings, and enlarge the differences between the input and output for the anomalous samples.

This structure incorporates a pair of memory modules, which retain the most typical patterns in the training materials in a self-supervised, adaptive manner. During the inference process, the incoming embeddings are compared with the knowledge in the memory modules. If the incoming embeddings significantly deviate from the learned knowledge, a rectification process would happen by retrieving information in the memory module.

In specific, we initiate memory modules in both the spatial and temporal memory mechanisms, which are denoted as $m^{\text{spat}} \in \mathbf{R}^{I\times E}$ for the spatial one or $m^{\text{temp}} \in \mathbf{R}^{I\times E}$ for the temporal one. E in its size matches the size of $z_t^{\text{spat}} \in \mathbf{R}^{1\times E}$, and the I is the number of memory items. In this way, the information in the original embeddings can be retrieved from the learned memories contents,

$$\widetilde{z}_t = \text{Softmax}\,(z_t m^{\text{T}})m \tag{5}$$

Equation (5) is the dot-product attention mechanism, where z_t serves as the query ($\boldsymbol{q}$), and both the key ($\boldsymbol{k}$) and the value ($\boldsymbol{v}$) are the corresponding memory module m.

Following, we take the Spatial Memory Mechanism for a more detailed explanation. This mechanism evaluates the correlation between the information stored in each component of the memory module m, and the original spatial embedding z_t^{spat}. Based on the similarity between the embedding and the memory module, the hit rate vector is calculated,

$$\boldsymbol{h}_t^{\text{spat}} = z_t^{\text{spat}}(m^{\text{spat}})^{\text{T}} \tag{6}$$

Then, we use a Softmax function to transfer the range of the hit rate vector and form the index,

$$\boldsymbol{r}_t^{\text{spat}} = \text{Softmax}\,(\boldsymbol{h}_t^{\text{spat}}) \tag{7}$$

To indicate the components in the index vector, we use $r_{t,i}^{\text{spat}}$ to denote the i-th component of the index $\boldsymbol{r}_t^{\text{spat}} \in \mathbf{R}^{1\times I}$. Following the calculation of the index, a Top-K operation is performed to prevent excessive indexing and retain only the most relevant information,

$$\widetilde{\boldsymbol{r}}_t^{\text{spat}} = \text{Top-K}\,(\boldsymbol{r}_t^{\text{spat}}) \tag{8}$$

Then, the Spatial Memory Mechanism retrieves information in m^{spat},

$$\widetilde{z}_t^{\text{spat}} = \widetilde{\boldsymbol{r}}_t^{\text{spat}} m^{\text{spat}} \tag{9}$$

The rectified spatial embedding $\widetilde{z}_t^{\text{spat}} \in \mathbf{R}^{1\times E}$ has the same size of the original embedding z_t^{spat}, but with the components reorganized and rectified. The process for rectified temporal embedding $\widetilde{z}_t^{\text{temp}}$ is similar. During the training process, the memory module's components are updated under the guidance of the compound loss.

3.5 Agent attention enhanced embedding fusion

Upon the generation of rectified embeddings ($\widetilde{z}_t^{\text{spat}}$ and $\widetilde{z}_t^{\text{temp}}$) from both of the memory modules, we introduce an agent attention enhanced fusion mechanism to consolidate the information in these embeddings. To unite the information in these embeddings, we concatenate them as follows:

$$\widetilde{z}_t^{\text{c}} = [\widetilde{z}_t^{\text{spat}}, \widetilde{z}_t^{\text{temp}}] \tag{10}$$

where $[\cdot]$ represents the concatenation operation. This process produces the raw fusion embedding, $\widetilde{z}_t^{\text{c}}$, containing information from both the spatial and temporal characteristics. The motivation of this process is to create a comprehensive representation that contains both perspectives to record the dependencies in the input sequences. In subsequent phases of Spectra, this representation holds the foundation for further analysis and inference. However, due to several issues, the concatenated raw fusion embeddings are not immediately suitable for further processing. These embeddings may face the problem of information redundancy and disorganization, which may hinder their effective utilization.

Considering this, we employ the agent attention mechanism to weigh and organize the components in the fusion embeddings. The mechanism (see Fig. 3) is designed to dynamically adjust the weights of the individual components within the concatenated embeddings, allowing the model to capture information from a range of aspects.

This mechanism expands the original self-attention to process an additional agent vector. The agent vector is generated from the original temporal embeddings through a linear transformation, and then used to guide the information utilization during the fusion process. Inside this mechanism, the regulated query vector is

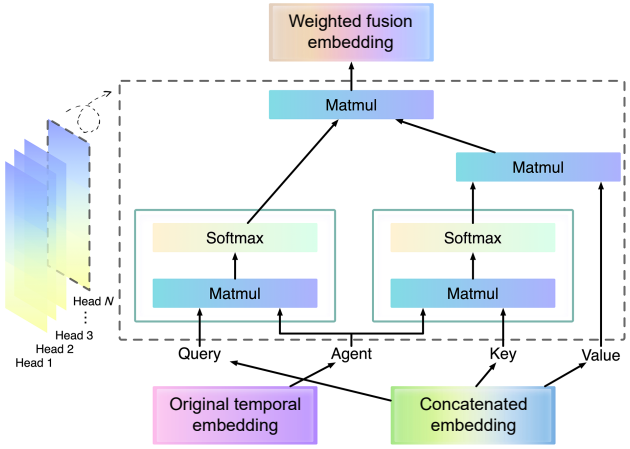

Fig. 3 Agent attention enhanced embedding fusion process.

formed as

$$\boldsymbol{q}_{\text{Reg}} = \text{Softmax}\,(\boldsymbol{q}\boldsymbol{a}^{\text{T}}) \tag{11}$$

where $\boldsymbol{a}$ is the agent vector.

Similarly, we have the regulated key vector as

$$\boldsymbol{k}_{\text{Reg}} = \text{Softmax}\,(\boldsymbol{a}\boldsymbol{k}^{\text{T}}) \tag{12}$$

Then, the regulated key vector interacts with the value vector to generate its regulated version,

$$\boldsymbol{v}_{\text{Reg}} = \boldsymbol{k}_{\text{Reg}}\boldsymbol{v} \tag{13}$$

The final output is thus formed,

$$\boldsymbol{o}_{\text{Aga}} = \boldsymbol{q}_{\text{Reg}}\boldsymbol{v}_{\text{Reg}} \tag{14}$$

To encapsulate information in different modalities of the concatenated embedding, a multi-head version of the agent attention mechanism is employed. During this process, information in each head $\boldsymbol{h}_i$ is updated as follows:

$$\boldsymbol{h}_i = \text{AgentAttention}\,(\boldsymbol{W}_q^i\widetilde{\boldsymbol{z}}_t^{\text{c}}, \boldsymbol{W}_k^i\widetilde{\boldsymbol{z}}_t^{\text{c}}, \boldsymbol{W}_v^i\widetilde{\boldsymbol{z}}_t^{\text{c}}, \boldsymbol{z}_t^{\text{temp}}) \tag{15}$$

where $\boldsymbol{W}_q^i$, $\boldsymbol{W}_k^i$, and $\boldsymbol{W}_v^i$ are the weight arrays for the query, key, and value vectors, respectively, and $\boldsymbol{z}_t^{\text{temp}}$ is taken as the agent vector. AgentAttention$(\cdot)$ is used to present the process described in Eqs. (11)–(14). The outputs from these heads are then fused as

$$\widetilde{\boldsymbol{z}}_t^{\text{f}} = \text{LN}\,[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N] \tag{16}$$

where LN is the layer normalization that helps to ease the internal covariate shift and stabilize the convergence of Spectra, and N is the number of attention heads.

3.6 Contrastive regulation based embedding alignment

Due to the nature of self-supervised training, noise and anomalous components in the training data can negatively affect the generation and retrieval of embeddings. Our proposed parallel memory and agent attention mechanisms provide a powerful tool to adjust embedding generation by applying dynamic information retrieval operations. By comparing and retrieving the typical information on data patterns that are obtained through historical readings, this process filters the components in the embeddings. Apart from implementing a specialized module to adaptively filter the components of the embeddings, we hope to design regulations that directly constrain the latent space and control the generation of embeddings.

To this end, we introduce a contrastive regulation to guide the model's training process and align the distribution of embeddings. Firstly, both the hit rate vectors described in Eq. (6), i.e., $\boldsymbol{h}_t^{\text{spat}}$ and $\boldsymbol{h}_t^{\text{temp}}$, are captured as indicators of the intensity of information retrieval from the memory module.

Based on these index vectors, a contrastive prompter is employed to regulate the parameter optimizations,

$$\begin{aligned}\mathcal{L}_{\text{ctr}} = &\sum_{k=1}^{E/2}\left(\boldsymbol{\mu}_k^{\text{spat}} - \frac{\tau_1}{I}\sum_{i=1}^{I}(h_{t,i}^{\text{spat}})^2\right) + \\ &\sum_{k=1}^{E/2}\left(\boldsymbol{\mu}_k^{\text{temp}} - \frac{\tau_2}{I}\sum_{i=1}^{I}(h_{t,i}^{\text{temp}})^2\right)\end{aligned} \tag{17}$$

where $\boldsymbol{\mu}^{\text{spat}}$ and $\boldsymbol{\mu}^{\text{temp}}$ refer to the mean component of the embeddings $\boldsymbol{z}^{\text{spat}}$ and $\boldsymbol{z}^{\text{temp}}$, respectively; $h_{t,i}^{\text{spat}}$ and $h_{t,i}^{\text{temp}}$ are the i-th component in the hit rate vectors $\boldsymbol{h}_t^{\text{spat}}$ and $\boldsymbol{h}_t^{\text{temp}}$, respectively; τ_1 and τ_2 are the weights for this regulation.

This setting is inspired by the idea of incorporating external information and aligning the distribution of embeddings in the latent space. We use the similarity between the extracted embeddings and contents in the memory module as an implicit measurement to enhance the alignment of embeddings. This measurement represents the extent to which the input sequence matches the significant patterns that have been previously learned and stored in the memory module.

Under this design, embeddings of a potential anomalous sample and the information stored in the memory module would become less relevant. Subsequently, the contrastive prompter $\mathcal{L}_{\text{ctr}}$, as defined in Eq. (17), directs the encoder to adjust its parameters to exhibit a deviation from the clusters of normal instances when processing anomalous samples. This

deviation, in turn, results in less likelihood of these samples' embeddings to match records in the memory module m. A more equalized index for information retrieval would then be obtained, leading to a more intensified information retrieval. We illustrate this process in Fig. 4 to provide a more straightforward demonstration.

3.7 Decoder and parameter optimization

When the $\widetilde{z}_t^{\mathrm{f}}$ is ready, Spectra uses a decoder with a symmetric setting to the encoder (see Section 3.3) for data reconstruction,

$$\widetilde{\boldsymbol{x}}_t = \mathrm{Decoder}\left(\widetilde{z}_t^{\mathrm{f}}\right) \tag{18}$$

The decoder architecture mainly consists of a Bi-GRU layer, followed by a sequence of four successive layers of 1D Transposed Convolution, denoted as Conv1DT layers. The incorporation of Bi-GRU layers in the decoder plays a pivotal role in modeling complex temporal relationships within the sequential data. By facilitating simultaneous information exchange from both forward and backward directions, the Bi-GRU layers enable the decoder to capture complex dependencies that may span across different timesteps. Meanwhile, the Conv1DT layers leverage transposed convolutional operations to expand the feature maps, allowing for a more detailed representation that captures subtle patterns. By combining these two kinds of layers, Spectra leverages both recurrent and convolutional approaches that extract essential dependencies within the target sequences.

To optimize the parameters in the model, Spectra employs a compound loss function that combines the reconstruction loss, the Kullback–Leibler Divergence (KL-Div), and the sparsity loss for parameter regulation,

$$\mathcal{L}_{\mathrm{sum}} = \mathcal{L}_{\mathrm{rec}} + \eta_1 \mathcal{L}_{\mathrm{spr}} + \eta_2 \mathcal{L}_{\mathrm{ctr}} \tag{19}$$

where η_1 and η_2 represent the weights assigned to corresponding component.

The first component, i.e., the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$, is calculated as follows:

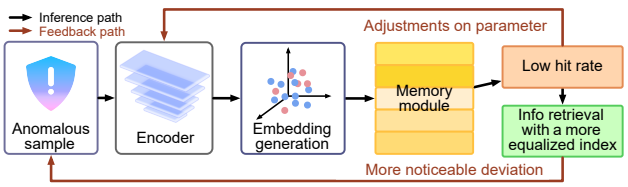

Fig. 4 Contrastive regulation on the latent space.

$$\mathcal{L}_{\mathrm{rec}} = E_{q_\phi(\widetilde{z}_t^{\mathrm{f}}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\widetilde{z}_t^{\mathrm{f}})] - \mathrm{KL}\left[q_\phi(\widetilde{z}_t^{\mathrm{f}}|\boldsymbol{x}) \| p_\theta(\widetilde{z}_t^{\mathrm{f}})\right] \tag{20}$$

where $p_\theta(\cdot)$ is the approximate posterior distribution, and $q_\theta(\cdot)$ is the likelihood function. The term $\mathrm{KL}\left[q_\phi(\boldsymbol{z}|\boldsymbol{x}) \| p_\theta(\boldsymbol{z})\right]$ is calculated as follows:

$$\begin{aligned}&\mathrm{KL}[q_\phi(\widetilde{z}_t^{\mathrm{f}}|\boldsymbol{x}) \| p_\theta(\widetilde{z}_t^{\mathrm{f}}|\boldsymbol{x})] = \\ &E_{q_\phi(\widetilde{z}_t^{\mathrm{f}}|\boldsymbol{x})}[\log q_\phi(\boldsymbol{z}|\boldsymbol{x}) - \log p_\theta(\boldsymbol{x}, \widetilde{z}_t^{\mathrm{f}})] + \log p_\theta(\boldsymbol{x})\end{aligned} \tag{21}$$

The sparsity loss $\mathcal{L}_{\mathrm{spr}}$ is formulated as follows:

$$\mathcal{L}_{\mathrm{spr}} = \sum_{i=1}^{I} -\log\left(1 + (r_i^{\mathrm{spat}})^2 + (r_i^{\mathrm{temp}})^2\right) \tag{22}$$

The contrastive prompter $\mathcal{L}_{\mathrm{ctr}}$ is defined in Eq. (17).

3.8 Anomaly detection

After obtaining the reconstructed data $\widetilde{\boldsymbol{x}}_t$, Spectra uses Mean Square Error (MSE) to compare the discrepancies between the scaled observations and the reconstructed sequences,

$$e_t = \frac{1}{L}\sum_{i=1}^{L}(\boldsymbol{x}_{(i)} - \widetilde{\boldsymbol{x}}_{(i)})^2 \tag{23}$$

where $\widetilde{\boldsymbol{x}}_{(i)}$ and $\widetilde{\boldsymbol{x}}_{(i)}$ are the i-th timestep in $\widetilde{\boldsymbol{x}}_t$ and $\boldsymbol{x}_t$, respectively. During the inference process, the on-site data are scaled (Eq. (1)) according to the rule derived from historical observations, thus ensuring consistency in the data range.

We denote the training dataset as $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_K\}$. The threshold to spot anomalies is calculated as follows[20]:

$$\lambda = \frac{1}{K}\sum_{t=1}^{K} e_t + \sqrt{\frac{1}{K}\sum_{t=1}^{K}(e_t - \mu)^2} \tag{24}$$

where μ is the average discrepancy of all samples in the training set. The anomalies are thus determined by conducting the discrepancy analysis,

$$l_t = \begin{cases} 0, & \text{if } e_t < \lambda; \\ 1, & \text{otherwise} \end{cases} \tag{25}$$

As presented in Eq. (25), the samples with more significant discrepancies than the threshold are labeled anomalous.

4 Evaluation

4.1 Experiment settings

Metrics. In data collected from industrial applications,

anomalies exhibit a notably lower occurrence rate. The accuracy alone thus becomes insufficient for evaluating the performance of an AD solution.

In the experiment section, we use the metric system widely adopted by other anomaly detection works[19, 20, 31, 32, 36] as follows:

$$\begin{aligned}\text{Precision} &= \frac{\text{TP}}{\text{TP}+\text{FP}},\\ \text{Recall} &= \frac{\text{TP}}{\text{TP}+\text{FN}},\\ \text{F1-Score} &= \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}.\end{aligned} \tag{26}$$

In the above formulations, True Positive (TP) stands for positive instances that the model correctly inferred. False Positive (FP), and False Negative (FN) denote positive or negative instances that the model incorrectly identifies.

Test environment. The experiments are conducted on a server with computational capabilities similar to those typically used in industrial settings. Its main configurations are listed in Table 1.

Parameters. The following parameters in Table 2 are used in the experiments.

4.2 Baselines and datasets

Baselines. We reproduce the performance of several deep learning state-of-the-art works, i.e., VAE-LSTM[32], OmniAnomaly[31], USAD[19], InterFusion[36], and CAE-M[20], to serve as benchmarks. Considering machine learning algorithms are also widely deployed in industrial applications, a classical machine learning classification algorithm Isolation Forest (IF) is also included for evaluation.

Datasets. We use three large-scale industrial datasets during the evaluation. The datasets include the SMD[31], an AD dataset containing service center statistics, and two other AD datasets sampled from industrial process recordings: SMAP[46], and SWaT[47]. We list their specifications in Table 3 for reference.

Table 1 Test environment configuration.

Item	Configuration
CPU	Intel Xeon 6142
RAM	64 GB DDR4
Storage	1 TB SSD
GPU	NVIDIA RTX A5000
ML Env	TensorFlow 2.13.1, CUDA 11.8, cuDNN 8.6

Table 2 Parameters.

Item	Value
Batch size	64
Window length	128
Epochs	500 (with early stopping)
Optimizer	Adamax (lr = 2.5×10^{-3})

4.3 Overall performance

In this section, we evaluate the overall performance of Spectra, as compared to the baselines on three large-scale datasets collected from real-world industrial scenarios. During the evaluation process, three random trials on each dataset are performed and the average performance. The results are presented in Table 4 for a direct comparison.

It is clear from Table 4 that Spectra achieves a noticeable advancement in the overall anomaly detection performance. Specifically, the observed advance in the F1-Score is of particular significance. This advancement is quantified by an increase of 2.15% in the F1-Score compared to the best baseline results. This rise in the F1-Score underscores the elevated ability of Spectra to strikingly enhance the system's capacity to accurately detect anomalies within the data from real-world applications.

Generally speaking, different degrees of growth can be observed in these datasets. For these large-scale datasets, the most salient increase in F1-Score can be observed in SMAP, with a boost of up to 4.43%. As for the dataset with higher dimensionality, i.e., the SWaT, Spectra also gets an increase of 2.19% in F1-Score. This increment is mainly attributed to Spectra's ability to effectively utilize features extracted in the spatial and temporal domain while rectifying the embeddings, which is essential under industrial scenarios. As for the service center dataset SMD, Spectra also performs better than the best baseline.

Table 3 Properties of the datasets.

Dataset	Scenarios	Number of monitored entities	Number of metrics of each entity	Record duration	Anomaly ratio (%)	Noise in training set
SMD[31]	Service Center	28	38	5 weeks	4.16	Yes
SMAP[46]	Satellite Telemetry	55	25	12 months	13.13	Yes
SWaT[47]	Water Treatment	–	51	11 days	12.14	Yes

Table 4 Overall evaluation. The best results are in bold, while the second-best ones are underlined.

Works	SMD			SWaT			SMAP			Average			Comp. (%)
	mP	mR	mF	mP	mR	mF	mP	mR	mF	mP	mR	mF	
IF	0.5909	0.8302	0.6904	0.7476	0.6444	0.6922	0.4622	0.5354	0.4961	0.6002	0.6700	0.6262	68.95
VAE-LSTM[32]	0.8181	0.8962	0.8554	0.8909	0.6035	0.7196	0.8318	0.8610	0.8462	0.8470	0.7869	0.8070	88.86
OmniAnomaly[31]	0.9340	0.9528	0.9433	0.9805	0.6558	0.7859	0.7893	**0.9312**	0.8544	0.9013	0.8466	0.8612	94.82
USAD[19]	0.9274	0.9709	0.9486	0.9805	0.6953	0.8137	0.8466	0.8881	0.8668	0.9182	0.8515	0.8764	96.49
InterFusion[36]	0.9438	**0.9825**	0.9627	**0.9869**	0.6957	0.8161	0.8551	0.9247	0.8885	0.9286	0.8676	0.8891	97.89
CAE-M[20]	0.9293	0.9435	0.9364	0.9762	0.6520	0.7818	0.8940	0.8815	0.8877	0.9332	0.8257	0.8686	95.64
Spectra	**0.9624**	0.9633	**0.9629**	0.9483	**0.7442**	**0.8340**	**0.9287**	0.9271	**0.9279**	**0.9465**	**0.8782**	**0.9083**	**100.00**

Note: mP: mean Precision; mR: mean Recall; mF: mean F1-Score; Comp.: a numerical comparison derived from the average mF metric for the baselines，taking the average mF of Spectra as 100%.

4.4 Validation of the parallel memory

We introduce a novel parallel memory design to Spectra, and incorporate it within the conventional generative model structure. To validate its effectiveness, the memory modules in Spectra are partially masked to form different variants (Table 5). The anomaly performance of these variants is then compared, and the results are presented in Fig. 5.

From Fig. 5, we can observe that the introduction of the parallel memory module helps Spectra to extract and bridge the complex dependencies in the target industrial data from different aspects. With the complete parallel memory module activated, the Variant-O achieves the best performance compared to the other two variants. This finding is consistent with the design of this module, as it helps the model to leverage and portray the patterns more effectively. With only the temporal memory activated, the Variant-T achieves the second-best result. This observation suggests that the temporal memory mechanism is more robust in capturing the temporal dependencies in the time series data, which may hold a more significant role in industrial scenarios. The input of the spatial memory mechanism is truncated from the encoder module. As the information from the encoder module is not fully utilized, the performance of Variant-S is relatively limited. Despite the performance in record, we should note that the utilization of spatial memory is still essential in the Spectra model, as it helps to clean and organize the learned patterns, and contributes to the overall performance improvement.

Table 5 Settings of the variants.

Variant	Setting
Variant-O	Spectra with full model structure
Variant-S	Temporal memory mechanism and all Bi-GRU not enabled
Variant-T	Spatial memory mechanism not enabled

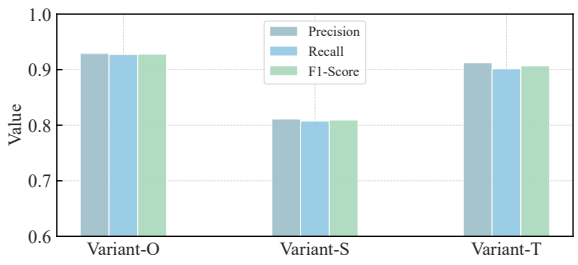

Fig. 5 F1-Score on SMAP of different variants.

4.5 Visualization of the embeddings

We have validated the effectiveness of the parallel memory module in Section 4.4 and concluded that this unique design contributes to improvements in AD performance. To gain insight into the module's effectiveness, we discuss the impacts produced by this design by plotting the distribution of the generated embeddings in latent space using t-SNE. The visualization is presented in Fig. 6.

In the visualization results, the proximity of these dots indicates their similarity in the latent space, while clusters of the dots suggest shared characteristics or patterns among the samples. Figure 6 indicates that the memory design contributes to more compact and converged embeddings in the latent space.

Figure 6a indicates the embedding distribution when no memory mechanism is applied. It can be observed that the embeddings are more dispersed across the latent space. This finding suggests that the absence of engagement with the parallel memory mechanism poses challenges for the remaining generative model's ability to effectively extract and encapsulate patterns from incoming sequences. The reconstruction quality can be affected as the captured patterns may deviate from the normal ones. It becomes more challenging to

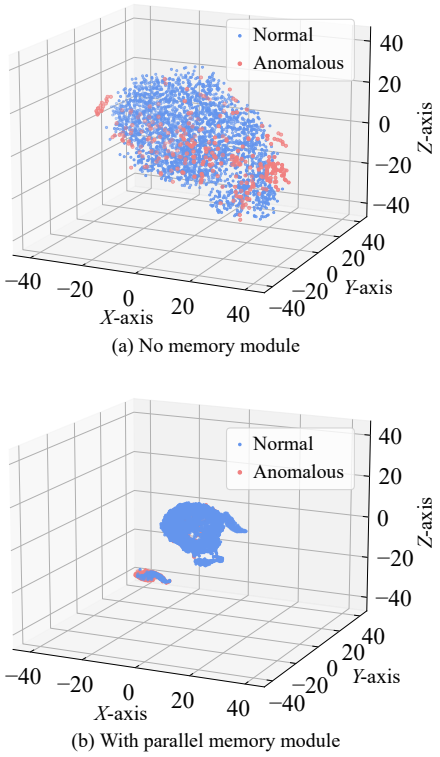

Fig. 6 Visualization of Spectra's embedding distribution. Blue/red dots are embeddings of normal/anomalous samples, respectively.

spot anomalous samples accurately.

Conversely, Fig. 6b visualizes the distribution of the embeddings rectified by the parallel memory module. As we can see, both normal and abnormal samples use a shared encoding approach, resulting in a similar distribution in the latent space for both the blue and red dots. This finding indicates that the memory mechanism helps the generative model in Spectra to focus on the encoding approach for normal samples, thus weakening the reconstruction quality for anomalous samples. Besides, the overall distribution of the embeddings is more compact, validating the effectiveness of the memory module in cleaning and organizing the learned patterns.

In conclusion, this experiment demonstrates that the introduction of the parallel memory module leads to an enhanced encoding of normal patterns, and consequently, an improvement in AD performance. This unique design enables Spectra to leverage stored information, guiding the embedding process and promoting the formation of more structured and distinguishable embeddings in the latent space.

4.6 Validation of anomaly portions

As discussed in prior sections, anomalies are common in industrial scenarios. To address this concern, Spectra integrates a pair of memory modules to store the typical patterns from historical observations, thus being able to handle data with anomalous components.

We conduct an ablation study to test and validate the effectiveness of Spectra in handling data under different noise levels and data irregularities. Considering that the training sets of the selected datasets are unlabeled, we process the SMD dataset as follows to satisfy the need to form a dataset with known anomaly portions.

The original test set of SMD is divided into three parts in temporal order: the (labeled) training set, the validation set, and the new test set. Upon the newly formed training set, the anomalies contained in it are manually sampled so as to adjust the proportion of anomalies in the training set. We visualize the results in Fig. 7.

We can observe from Fig. 7 that Spectra's AD performance remains robust to different noise levels, even with a portion of 20% of anomalies mixed in the training material. Since it is "unrealistic" that 20% of the samples are unnoticed anomalies during an extended period, it is much less likely that 20% of the observations are generated from failures in industrial scenarios. However, the results indicate that despite Spectra sacrificing precision to maintain recall in this extreme scenario, the overall F1-Score remains sustained.

Generally speaking, as the noise level increases, the F1-Score of Spectra shows a slight decrease but remains relatively high. We attribute this robustness to the embedding rectification empowered by the memory mechanism in Spectra, which ensures the robust learning of the normal patterns in the data.

4.7 Validation of window length

Segmenting original time series data into fixed-length windows is crucial in applying Spectra and other deep

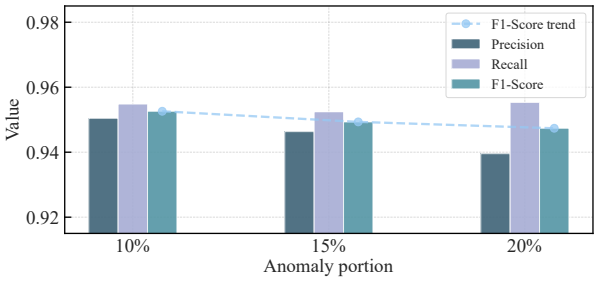

Fig. 7 F1-Score on SMD under different anomaly portions.

learning AD models. The segmented sequences represent the temporal context or history the model considers to extract complex dependencies. In this section, we present a detailed experimental analysis of the impact of different window lengths to investigate its effects. In this experiment, we evaluate the AD performance of Spectra under different window lengths.

In detail, we select four specific window lengths: 48, 64, 80, 96, 112, and 128. These lengths are chosen to cover a range of short to medium window sizes, allowing us to evaluate the effectiveness of Spectra under various temporal resolutions. The experimental results are depicted in Fig. 8 for a clear comparison.

By default, we use a window length $L = 128$ for data pre-processing. This setting is based on the consideration that the span of more timesteps enables the generative model to better capture global patterns and long-range dependencies within its temporal scope of the target data stream. This is beneficial for detecting anomalies that occur over an extended period, such as gradual drifts or trends. The experiment results also validate this point: Spectra achieves the most effective AD performance with this setting.

With the decrease in window length, a relative, mild performance impact can be observed in the value of F1-Score. This result can be attributed to limited contextual information, which limits the analysis along the temporal axis. However, it is also important to note that the AD performance of Spectra generally remains high, regardless of the window length.

5 Conclusion

Modern industrial operations rely on a vast network comprising devices with varying patterns and intricate topologies. To ensure operation smoothness, reduce downtime, and improve efficiency, we propose Spectra, a novel and effective anomaly detection framework for industrial time-series data. We incorporate a pair of parallel memory modules in the architecture of generative models to capture and store typical dependencies. We then design an embedding fusion mechanism comprising an agent attention module and a contrastive embedding alignment technique to integrate knowledge portrayed in different aspects, and help distinguish between normals and anomalies. Extensive results on three large-scale, real-world datasets suggest that the anomaly detection performance of Spectra outperforms the baselines.

For future works, we suggest that researchers may focus on fine-grained analysis of discrepancies, a more advanced training protocol, and a more robust structure of the generative models.

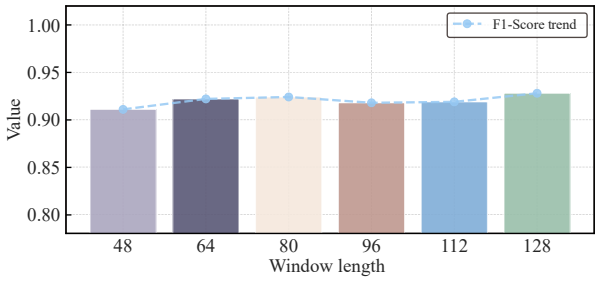

Fig. 8 F1-Score on SMAP under different window lengths.

Acknowledgment

This project was supported by the Fundamental Research Funds for the Central Universities (No. 2023YJS009), the National Natural Science Foundation of China (Nos. 62472023, 62402027, and 62072029), the Natural Science Foundation of Beijing Municipality (No. L221003), the Beijing Nova Program (Nos. 20230484263 and 20240484607), and the DiDi Research Collaboration Plan.

References

[1] W. N. He, X. L. Huang, Z. Xu, F. Hu, and S. Yu, Robust localization for mobile targets along a narrow path with Los/NLoS interference, *IEEE Internet Things J.*, vol. 11, no. 11, pp. 20853–20866, 2024.

[2] C. Hazman, A. Guezzaz, S. Benkirane, and M. Azrour, Enhanced ids with deep learning for iot-based smart cities security, *Tsinghua Science and Technology*, vol. 29, no. 4, pp. 929–947, 2024.

[3] Q. Zeng, F. Li, Z. Zhao, Y. Li, and Y. Wang, Acouwrite: Acoustic-based handwriting recognition on smartphones, *IEEE Trans. Mobile Comput.*, vol. 23, no. 8, pp. 8557–8568, 2024.

[4] L. Zhang, L. Zhang, R. Xie, Y. Ni, X. Wu, Y. Yang, F. Xing, X. Zhao, and Z. You, Highly tunable cascaded metasurfaces for continuous two-dimensional beam steering, *Adv. Sci.*, vol. 10, no. 24, p. 2300542, 2023.

[5] R. K. Dwivedi, R. Kumar, and R. Buyya, Gaussian distribution-based machine learning scheme for anomaly detection in healthcare sensor cloud, *Int. J. Cloud Appl. Comput.*, vol. 11, no. 1, pp. 52–72, 2021.

[6] Y. Qu, S. Yu, L. Gao, K. Sood, and Y. Xiang, Blockchained dual-asynchronous federated learning services for digital twin empowered edge-cloud continuum, *IEEE Trans. Serv. Comput.*, vol. 17, no. 3, pp. 836–849, 2024.

[7] W. Cheng, Z. Jiang, C. Xiang, and J. Fu, Marginal effect-aware multiple-vehicle scheduling for road data collection: A near-optimal result, *ACM Trans. Sen. Network.*, vol. 20,

no. 6, p. 116, 2024.

[8] M. Yao, D. Tao, J. Wang, R. Gao, and K. Sun, MARVAir: Meteorology augmented residual-based visual approach for crowdsourcing air quality inference, *IEEE Trans. Instrum. Meas*., vol. 71, p. 2514310, 2022.

[9] J. Wang, H. Du, D. Niyato, J. Kang, Z. Xiong, D. Rajan, S. Mao, and X. Shen, A unified framework for guiding generative AI with wireless perception in resource constrained mobile edge networks, *IEEE Trans. Mobile Comput.*, vol. 23, no. 11, pp. 10344–10360, 2024.

[10] W. Cheng, H. Lu, C. Xiang, D. Liu, and T. Xiang, Breaking "chicken-egg": Cross-city battery swap demand prediction via knowledge-guided diffusion, in *Proc. 44th Annu. IEEE Int. Conf. Computer Communications*, 2025, London, UK, 2025, pp. 1–10.

[11] J. Zhang, Z. Xu, D. Lv, Z. Shi, D. Shen, J. Jin, and F. Dong, Dig-in-GNN: Discriminative feature guided GNN-based fraud detector against inconsistencies in multi-relation fraud graph, in *Proc. 38th AAAI Conf. Artificial Intelligence*, Vancouver, Canada, 2024, pp. 9323–9331.

[12] Z. Cao, X. Zheng, J. Guo, W. Jia, Y. Wu, and T. Wang, Cost-effective dynamic alliance pricing mechanism based on distributed edge intelligence, *IEEE Internet Things J.*, vol. 11, no. 21, pp. 34471–34481, 2024.

[13] H. Ye, X. Xian, J. R. C. Cheng, B. Hable, R. W. Shannon, M. K. Elyaderani, and K. Liu, Online nonparametric monitoring of heterogeneous data streams with partial observations based on Thompson sampling, *IISE Trans.*, vol. 55, no. 4, pp. 392–404, 2023.

[14] X. Jiang, H. Luo, Y. Sun, and M. Guizani, Fast anomaly detection for IoT services based on multisource log fusion, *IEEE Internet Things J.*, vol. 11, no. 6, pp. 9405–9419, 2024.

[15] J. Dai and Y. Wang, Autoencoder anomaly detection method enhanced by hash memory network, (in Chinese), *J. Chin. Comput. Syst.*, vol. 45, no. 6, pp. 1301–1310, 2024.

[16] D. D. N. Nguyen, K. Sood, Y. Xiang, L. Gao, L. Chi, G. Singh, and S. Yu, Design and robust evaluation of next generation node authentication approach, *IEEE Trans. Depend. Secure Comput.*, vol. 21, no. 6, pp. 5311–5323, 2024.

[17] C. Tang, Y. Ding, S. Xiao, H. Wu, and R. Li, Joint optimization of service caching task offloading and resource allocation in cloud-edge cooperative network, in *Proc. IEEE Int. Conf. Communications*, Denver, CO, USA, 2024, pp. 4036–4041.

[18] T. Zhang, C. Xu, Y. Lian, H. Tian, J. Kang, X. Kuang, and D. Niyato, When moving target defense meets attack prediction in digital twins: A convolutional and hierarchical reinforcement learning approach, *IEEE J. Sel. Areas Commun.*, vol. 41, no. 10, pp. 3293–3305, 2023.

[19] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, USAD: Unsupervised anomaly detection on multivariate time series, in *Proc. 26th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Virtual Event, 2020, pp. 3395–3404.

[20] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, Unsupervised deep anomaly detection for multi-sensor time-series signals, *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 2118–2132, 2023.

[21] M. Yao, D. Tao, P. Qi, and R. Gao, Rethinking discrepancy analysis: Anomaly detection via meta-learning powered dual-source representation differentiation, *IEEE Trans. Autom. Sci. Eng*., doi: 10.1109/TASE.2024.3486688.

[22] Z. Zheng, H. Ye, and K. Liu, Online nonparametric monitoring for asynchronous processes with serial correlation, *IISE Trans.*, vol. 57, no. 2, pp. 172–185, 2025.

[23] J. Fan, G. Tang, K. Wu, Z. Zhao, Y. Zhou, and S. Huang, Score-VAE: Root cause analysis for federated-learning-based IoT anomaly detection, *IEEE Internet Things J.*, vol. 11, no. 1, pp. 1041–1053, 2024.

[24] J. Ramos, N. Nedjah, L. de Macedo Mourelle, and B. B. Gupta, Visual data mining for crowd anomaly detection using artificial bacteria colony, *Multimed. Tools Appl.*, vol. 77, no. 14, pp. 17755–17777, 2018.

[25] P. Gulihar and B. B. Gupta, Anomaly based mitigation of volumetric DDoS attack using client puzzle as proof-of-work, in *Proc. 3rd IEEE Int. Conf. Recent Trends in Electronics, Information & Communication Technology*, Bangalore, India, 2018, pp. 2475–2479.

[26] W. Liu, H. Jiang, D. Che, L. Chen, and Q. Jiang, A real-time temperature anomaly detection method for IoT data, in *Proc. 5th Int. Conf. Internet of Things, Big Data and Security*, Prague, Czech Republic, 2020, pp. 112–118.

[27] N. B. Chikodili, M. D. Abdulmalik, O. A. Abisoye, and S. A. Bashir, Outlier detection in multivariate time series data using a fusion of K-medoid, standardized Euclidean distance and Z-score, in *Proc. 3rd Int. Conf. Information and Communication Technology and Applications*, Minna, Nigeria, 2020, pp. 259–271.

[28] D. Kim and T. Y. Heo, Anomaly detection with feature extraction based on machine learning using hydraulic system IoT sensor data, *Sensors*, vol. 22, no. 7, p. 2479, 2022.

[29] N. Li, X. Liu, Z. Liu, L. Mao, L. Zhao, and X. Wang, Anomaly detection in power grid IoT system based on isolated forest, in *Proc. IEEE/WIC/ACM Int. Conf. Web Intelligence and Intelligent Agent Technology*, Melbourne, Australia, 2022, pp. 9–12.

[30] R. Pavaiyarkarasi, T. Manimegalai, S. Satheeshkumar, K. Dhivya, and G. Ramkumar, A productive feature selection criterion for Bot-IoT recognition based on random forest algorithm, in *Proc. 11th IEEE Int. Conf. Communication Systems and Network Technologies*, Indore, India, 2022, pp. 539–545.

[31] 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 *Proc. 25th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 2019, pp. 2828–2837.

[32] S. Lin, R. Clark, R. Birke, S. Schönborn, N. Trigoni, and S. Roberts, Anomaly detection for time series using VAE-LSTM hybrid model, in *Proc. 2020 IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Barcelona, Spain, 2020, pp. 4322–4326.

[33] L. Bai, L. Cui, Y. Wang, M. Li, J. Li, P. S. Yu, and E. R.

Hancock, HAQJSK: Hierarchical-aligned quantum Jensen-Shannon kernels for graph classification, *IEEE Trans. Knowledge Data Eng.*, vol. 36, no. 11, pp. 6370–6384, 2024.
[34] T. Zhang, Y. Liu, Z. Shen, R. Xu, X. Chen, X. Huang, and X. Zheng, An adaptive federated relevance framework for spatial–temporal graph learning, *IEEE Trans. Artif. Intell.*, vol. 5, no. 5, pp. 2227–2240, 2024.
[35] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, Multivariate time-series anomaly detection via graph attention network, in *Proc. 2020 IEEE Int. Conf. Data Mining*, Sorrento, Italy, 2020, pp. 841–850.
[36] 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 *Proc. 27th ACM SIGKDD Conf. Knowledge Discovery & Data Mining*, Virtual Event, 2021, pp. 3220–3230.
[37] P. Boniol and T. Palpanas, Series2Graph: Graph-based subsequence anomaly detection for time series, *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 1821–1834, 2020.
[38] P. Boniol, T. Palpanas, M. Meftah, and E. Remy, GraphAn: Graph-based subsequence anomaly detection, *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 2941–2944, 2020.
[39] L. Deng, D. Lian, Z. Huang, and E. Chen, Graph convolutional adversarial networks for spatiotemporal anomaly detection, *IEEE Trans. Neural Network. Learn. Syst.*, vol. 33, no. 6, pp. 2416–2428, 2022.
[40] Z. Chen, D. Jia, Y. Sun, L. Yang, W. Jin, and R. Liu, Univariate time series anomaly detection based on hierarchical attention network, *Tsinghua Science and Technology*, vol. 29, no. 4, pp. 1181–1193, 2024.
[41] H. Gao, B. Qiu, R. J. Durán Barroso, W. Hussain, Y. Xu, and X. Wang, TSMAE: A novel anomaly detection approach for internet of things time series data using memory-augmented autoencoder, *IEEE Trans. Network Sci. Eng.*, vol. 10, no. 5, pp. 2978–2990, 2023.
[42] Y. Zhang, J. Wang, Y. Chen, H. Yu, and T. Qin, Adaptive memory networks with self-supervised learning for unsupervised anomaly detection, *IEEE Trans. Knowledge Data Eng.*, vol. 35, no. 12, pp. 12068–12080, 2023.
[43] X. Chai, H. Zhang, J. Zhang, Y. Sun, and S. K. Das, Log sequence anomaly detection based on template and parameter parsing via BERT, *IEEE Trans. Depend. Secure Comput.*, doi: 10.1109/TDSC.2024.3428538.
[44] M. Yao, D. Tao, R. Gao, and P. Qi, Anomaly detection for MEC enabled hierarchical industrial IoT with transformer enhanced variational auto encoder, *IEEE Trans. Ind. Inf.*, vol. 21, no. 1, pp. 40–48, 2025.
[45] F. Barreto, S. Yadav, D. S. Patnaik, and D. J. Sarvaiya, MMCD VAE model for deep facial scale-invariant-feature-translation, in *Proc. 5th Int. Conf. Intelligent Computing and Control Systems*, Madurai, India, 2021, pp. 526–530.
[46] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding, in *Proc. 24th ACM SIGKDD Int. Conf. Knowledge Discovery & Data Mining*, London, UK, 2018, pp. 387–395.
[47] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, A dataset to support research in the design of secure water treatment systems, in *Proc. 11th Int. Conf. Critical Information Infrastructures Security*, Paris, France, 2017, pp. 88–99.

Muyan Yao received the BEng degree from Beijing Jiaotong University, China in 2020, where he is currently a PhD candidate. His research interests include ubiquitous computing and representation learning.

Dan Tao received the BEng and MEng degrees from Jilin University, China in 2001 and 2004, respectively, and the PhD degree from Beijing University of Posts and Telecommunications, China in 2007. She was a visiting scholar at Illinois Institute of Technology, USA from 2010 to 2011. She is currently a professor at School of Electronic and Information Engineering, Beijing Jiaotong University, China. She is also a senior member of the CCF. Her research interests include Internet of Things (IoT), crowdsensing, map construction and positioning, and big data and intelligent information processing.

Peng Qi received the PhD degree from Beijing University of Posts and Telecommunications, China in 2022. He is currently working at School of Electronic and Information Engineering, Beijing Jiaotong University, China. His research interests include IoT, recommendation systems, and Large Language Model (LLM).

Ruipeng Gao received the BEng degree from Beijing University of Posts and Telecommunications, China in 2010, and the PhD degree from Peking University, China in 2016. He was a visiting scholar at Purdue University, USA from 2018 to 2019. He is currently a professor at School of Cyberspace Science and Technology, Beijing Jiaotong University, China. His research interests include mobile computing and applications, IoT, and intelligent transportation systems.