Self-Supervised Disentangled Representation Learning for Time Series Anomaly Detection

Liang Zhang, Jianping Zhu, Guangjie Han, *Fellow, IEEE*, Bo Jin, *Senior Member, IEEE*, Pengfei Wang, *Member, IEEE*, and Xiaopeng Wei

***Abstract*—Anomaly detection is a fundamental component of intelligent monitoring in the Internet of Things (IoT), where accuracy, efficiency, and interpretability are critical requirements. However, existing methods often overlook the unique characteristics of IoT signals such as seasonality, trends, and irregular residual components, as well as the complex interactions among them. This oversight can lead to anomaly masking, increased false positives, and reduced interpretability in anomaly identification. Motivated by the effectiveness of disentangled representation learning, we propose TRAdetector, a novel disentangled reconstruction-based framework for IoT signals anomaly detection. TRAdetector explicitly models recurrent and consistent patterns, as well as irregular variations in the latent space by leveraging variational inference strategies, thereby enhancing probabilistic guidance in learning both regular and irregular temporal representations. A sparse coding strategy is incorporated within the latent space of the residual component to directly model inconsistent temporal fluctuations. Finally, a multihead cross-attention mechanism and a gated, decomposition-aware reconstruction strategy are designed to effectively model the complex interactions among different components. Extensive experiments show that our model achieves state-of-the-art performance on multiple benchmark datasets in terms of accuracy, efficiency, and interpretability.**

***Index Terms*—Disentangled representation learning, Internet of Things (IoT) time series anomaly detection, variational inference.**

I. Introduction

EFFECTIVE time series anomaly detection is critical across widespread Internet of Things (IoT) applications, such as pipeline safety early warning [1], industrial equipment monitoring [2], traffic management [3], smart manufacturing [4], and deep trust and reputation management [5]. For instance, in pipeline safety monitoring, detecting anomalies in pressure, flow rate, and temperature data can help prevent leaks, ruptures, and other hazardous incidents, ensuring timely intervention. However, distinguishing abnormal temporal patterns from the overwhelming majority of normal ones remains a significant challenge. This difficulty arises, on the one hand, from the scarcity of labeled anomalies, and on the other hand, from the complex nature of IoT signals, which often exhibit multiple temporal patterns such as seasonality, trends, and other dynamic behaviors [6]. These patterns can lead to anomaly masking and increased false positive rates. For instance, a rise in device temperature during the summer might be considered normal due to seasonal effects, potentially masking true anomalies. Conversely, regular periodic fluctuations may be misinterpreted as abnormal behavior, resulting in a high false alarm rate.

First, to solve the label scarcity problem, unsupervised deep learning paradigms [7] have emerged as the prevailing methods. Current unsupervised deep learning methodologies mainly focus on generating informative pointwise representations or capturing pairwise associations through reconstruction or autoregressive learning tasks. Nevertheless, as highlighted by [8], the rarity of anomalies often results in less informative pointwise representation learning, especially in complex temporal patterns where entangled normal patterns obscure abnormal ones.

Second, recent developments in decomposition-based time series forecasting, such as Autoformer [9], Fedformer [10], CoST [11], and LaST [12], have highlighted the benefits of disentangled representation learning in capturing intricate temporal patterns, such as seasonal trends and long-term variations. By effectively disentangling these components, these methods enhance forecasting accuracy and robustness, reducing the impact of noise interference, spurious correlations. In particular, anomalies in IoT time series data can emerge from various sources, including seasonal fluctuations [13], which arise due to periodic variations in external factors, long-term trend shifts [14], which reflect structural changes in the underlying dynamics, and abrupt event disruptions [15], which result from sudden external shocks or system malfunctions. Therefore, leveraging disentangled representations enables a more structured decomposition of time series data, facilitating the isolation of anomalous patterns from normal fluctuations. Moreover, computing component-wise reconstruction errors provides a fine-grained assessment of deviations from expected behavior, reducing spurious dependencies and improving model interpretability.

Received 1 June 2025; accepted 9 July 2025. Date of publication 18 July 2025; date of current version 25 September 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62202085 and Grant 61772110, and in part by the Fundamental Research Funds for the Central Universities under Grant DUT23RC(3)063. *(Corresponding authors: Bo Jin; Pengfei Wang; Xiaopeng Wei.)*

Liang Zhang, Bo Jin, Pengfei Wang, and Xiaopeng Wei are with the School of Computer Science and Technology and the Key Laboratory of Social Computing and Cognitive Intelligence, Ministry of Education, Dalian University of Technology, Dalian 116024, China (e-mail: liangzhang@dlut.edu.cn; jinbo@dlut.edu.cn; wangpf@dlut.edu.cn; xpwei@dlut.edu.cn).

Jianping Zhu is with the Artificial Intelligence Thrust, Hong Kong University of Science and Technology (Guangzhou), Guangzhou 510530, China (e-mail: zhujp@mail.dlut.edu.cn).

Guangjie Han is with the Key Laboratory of Maritime Intelligent Network Information Technology, Ministry of Education, Hohai University, Nanjing 210098, China (e-mail: hanguangjie@gmail.com).

Digital Object Identifier 10.1109/JIOT.2025.3588748

However, directly applying existing time series disentangled representation learning methods to IoT anomaly detection presents significant challenges. On one hand, current approaches [9], [10], [16] primarily disentangle time series into seasonal and trend components, as these recurrent and consistent patterns play a crucial role in long-term temporal dynamics. However, this focus may lead to the omission of localized irregular patterns that are not well captured by these components. On the other hand, both prevailing reconstruction-based methods [17] and autoregressive learning tasks [18] typically assume that hidden representations at each time step follow a Gaussian distribution. This assumption can result in significant reconstruction errors when faced with subtle or abrupt deviations from the normal distribution, ultimately leading to degenerate anomaly detection models.

To tackle these challenges, we propose TRAdetector, a disentangled reconstruction-based method. Specifically, we introduce an explicit residual component to capture additional temporal patterns beyond the predominant seasonal and trend components. For the regular seasonal and trend components, we design two sequentially structured variational inference networks to model their latent recurrent and consistent dynamics, which are assumed to follow Gaussian distributions. In contrast, the residual component, which captures additional irregular or abrupt changes [15], is modeled using a sparse coding-aware variational inference network with a spike-and-slab prior distribution [19], ensuring greater adaptability to non-Gaussian variations. Furthermore, to enhance series association, we introduce a multihead cross-attention mechanism that effectively captures sequential abnormal changes while maintaining robustness to subtle variations. Finally, a gated decomposition-aware reconstruction module facilitates component differentiation, enabling precise identification of anomalous time points.

Our contributions are summarized as follows.

1) We propose a self-supervised disentangled representation learning framework tailored for time series anomaly detection. This framework explicitly captures abnormal temporal patterns inherent in seasonality, trend, and residual components, ensuring a more structured and interpretable decomposition of anomalies.
2) We develop structured variational inference networks to enhance the learning of component-wise latent representations that follow different prior distributions for each component. Furthermore, we introduce a multihead cross attention mechanism and a gated decomposition-aware reconstruction strategy to enhance the interaction between disentangled latent representations.
3) Extensive experiments on multiple benchmark datasets demonstrate that TRAdetector achieves performance comparable or superior to state-of-the-art methods. Moreover, component-specific reconstruction error analysis provides empirical evidence that anomalies can be attributed to different components, offering more insightful and explainable anomaly detection results.

II. Related Work

Time Series Anomaly Detection: Time series anomaly detection aims at determining whether a data point at a certain timestamp is anomalous or not. The existing methodologies can be categorized into three primary groups: 1) traditional statistical methods; 2) classical machine learning approaches; and 3) emerging deep learning methods. Statistical methods primarily analyze the inherent statistical properties of data. Notable examples include ARIMA [20], Bayesian change point detection [21], Sampling, COPOD [22], and ECOD [23]. These methods typically rely on probabilistic models to detect anomalies based on deviations from established statistical patterns. Classical machine learning methods involve training models to recognize typical patterns within regular time series data, subsequently identifying deviations indicative of anomalies. Prominent methods within this category include OCSVM [24], PCA [25], kNN [26], CBLOF [27], HBOS [28], IForest [29], and LODA [30]. These methods utilize various linear transformations, proximity measures, or outlier detection algorithms to identify anomalous behaviors effectively. Recent advances in deep learning have significantly enhanced anomaly detection capabilities, driven by powerful representation learning techniques. These methods encompass a diverse range of architectures and strategies, such as VAE [31], LSTM-AE [32], MTAD-GAT [33], TFAD [16], OmniAnomaly [34], CAE-M [35], GDN [36], CSCAD [37], LSTM-VAE [18], BeatGAN [38], Anomaly Transformer [39], DCdetector [8], and Dynamic Decomposition with diffusion reconstruction (DR) (D^3R) [40]. Despite significant progress, these methods still fall short in terms of accuracy, efficiency, and interpretability.

Disentangled Representation Learning: Disentangled representation learning has been widely adopted in modeling distinct factors of variation and explanation [11], [12], [41]. For instance, CDDG [41] was proposed to learn intrinsic representations of semantic concepts that generalize across domains. In time series representation learning tasks, both LaST [12] and CoST [11] were proposed to reduce the degenerated model performance and lack of interpretation brought by a single representation. Disentangled variational autoencoders have been utilized to directly learn different representations with more probabilistic regularization. However, they mainly decomposed time series into seasonality and trend components without paying attention to the residual part, which serves as another irregular temporal dynamic source. For example, in BurstGraph [15], the bursty temporal behavior changes were captured by explicit modeling of residual irregular data.

Deep Self-Supervised Time Series Anomaly Detection: As mentioned in NPSR [42], most self-supervised time series anomaly detection methods typically entail computing an anomaly score at each time point and subsequently comparing this score to a predefined threshold. To obtain such score, temporal context-aware pointwise representation at each time point are required. Current deep self-supervised time series anomaly detection methods mainly utilize reconstruction-based [39] methods or contrastive learning

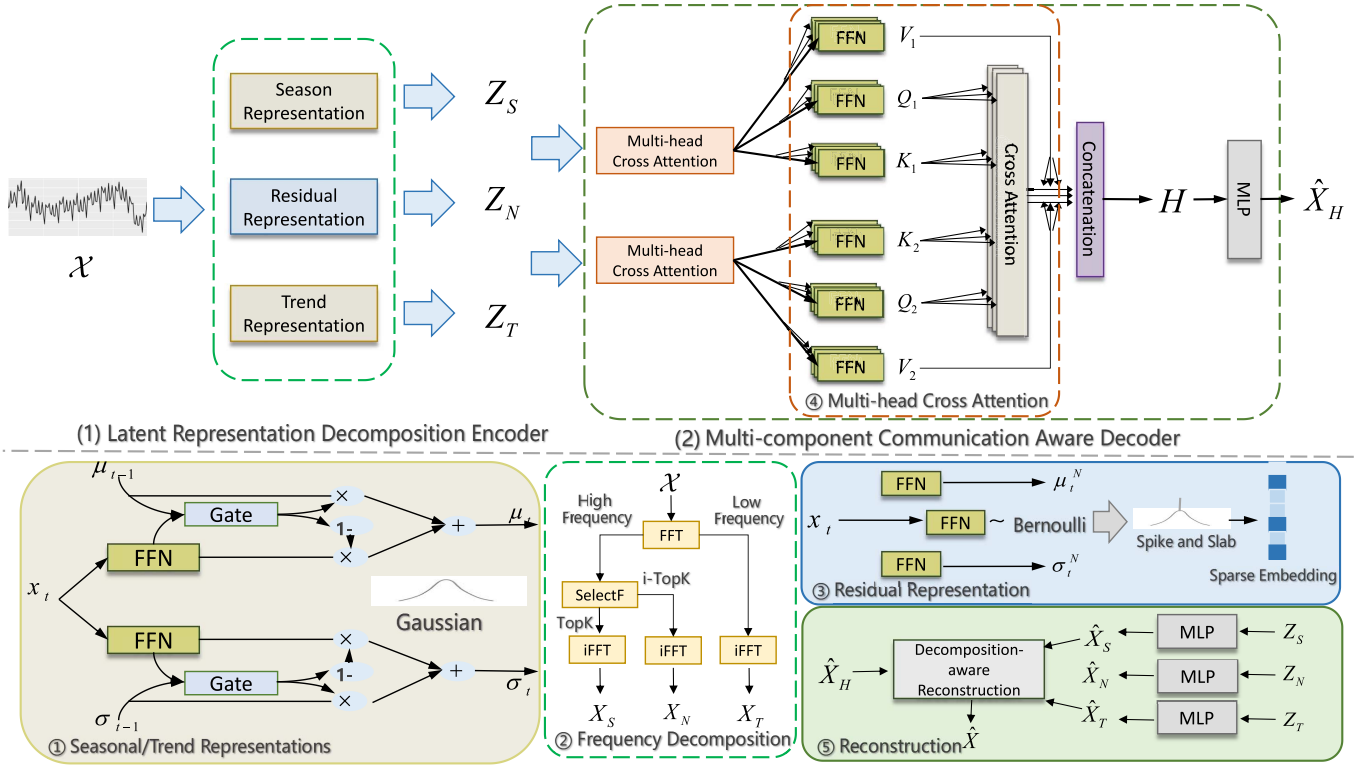

Fig. 1. Overview of proposed TRAdetector architecture. (1) Latent Representation Decomposition Encoder. It entails decomposing the original time series into seasonal, trend, and residual components. The inference of these latent representations rely on variational inference strategies. ① the R-VAEs learn the latent seasonal and trend representations, respectively. ② the frequency domain decomposition module decomposes the original sequence into seasonal, trend, and residual components for reconstruction constraints. ③ the Sparse-VAE is responsible for learning the residual representation. (2) Multicomponent Communication Aware Decoder. The interaction of the three latent representations enhances the representation consistency via ④ a stacked multihead cross attention module, followed by ⑤ a gated decomposition-aware reconstruction module.

methods [8] to get such informative representations. For instance, AnomalyTransformer [39] leveraged an Anomaly-Attention mechanism to compute the association discrepancy, and the parameters were learned in a reconstruction-based task. Despite its effectiveness, the off-the-shelf reconstruction-based methods easily suffer from obstruction of anomalies, making it harder to learn a well-reconstructed model [8]. DCdetector [8] was a multiscale dual attention contrastive representation learning model. However, the underlying assumption is the independence of the channels (or variables), which cannot be satisfied in most scenarios. In comparison, our proposed unsupervised learning method explicitly captures abnormal temporal patterns within the seasonal, trend, and residual components, without sacrificing residual information or relying on strict channel independence assumptions, while achieving improved accuracy, efficiency, and interpretability.

III. Methodology

Given an IoT multivariate time series $\mathcal{X} = \{x_1, x_2, \ldots, x_L\} \in \mathbb{R}^{L \times D}$, where L denotes the length of the time series and D represents the dimensions or number of channels at each timestamp. Our goal is to identify anomalous signals in an unsupervised setting through a reconstruction-based approach, i.e., after we learn a function f to obtain the reconstructed output $\hat{\mathcal{X}} = f(\mathcal{X})$, the anomaly score at each timestamp can be calculated based on some predefined metrics and thresholds, such as SPOT [40], [43]. Finally, we can generate a corresponding label sequence $\mathcal{Y} = \{y_1, y_2, \ldots, y_L\} \in \mathbb{R}^L$, which indicates the abnormal state at each timestamp. Specifically, if $y_i = 0$, it denotes that the data is normal at the ith timestamp; if $y_i = 1$, it indicates that the data is anomalous at the ith timestamp.

A. Overall Architecture

Fig. 1 illustrates the overall architecture of the TRAdetector, which consists of two parts: 1) latent representation decomposition encoder and 2) multicomponent communication aware decoder. Some major notations are illustrated in Table I. Detailed descriptions are provided in the following sections.

B. Latent Representation Decomposition Encoder

Disentangled latent representation learning enables the separation of various temporal patterns in time series forecasting tasks [9], [11]. For example, the seasonal component reflects the periodic changes in the time series, while the trend component signifies the long-term trajectory of the time series. Such explicit decomposition helps capture predominant temporal patterns while being robust to small interventions on the error variable [11]. In addition, the residual (or error) component represents the uncertainty and noise in the time series. Nevertheless, predicting and eliminating the residual component, often resembling white noise or abrupt changes,

TABLE I
MAJOR NOTATIONS IN THIS ARTICLE

Symbol	Description
$\mathcal{X}$	Multivariate time series
L	The length of the time series
D	Dimensions or number of channels at each timestamp
$\mathcal{Y}$	Label sequence
X_S, X_T, X_N	Seasonal, trend, and residual corresponding decomposed series
$\hat{X}_S$, $\hat{X}_T$, $\hat{X}_N$	Seasonal, trend, and residual corresponding reconstructed series
F_r	Real parts
F_i	Imaginary parts
mu_t	Mean vector of the seasonal or trend component at time t
$sigma_t$	Variance vector of the seasonal or trend component at time t
Z_S, Z_T, Z_N	Seasonal, trend, and residual corresponding representations
H_S, H_T	The seasonal representation Z_S and trend representation Z_T are each computed with the residual representation Z_N using cross-attention calculations.
H	H_S and H_T are computed using cross-attention calculations.

can be challenging. Many existing methods concentrate solely on the seasonal and trend components [10], [11], [12]. On the contrary, in the task of time series anomaly detection, the residual component might be the key to the generation of abnormal signals [15].

Frequency Decomposition: The Frequency Decomposition module disentangles the original time series into seasonal, trend, and residual components. Fast Fourier transform (FFT) is employed to convert the time series into the frequency domain. Subsequently, we identify the high-frequency segments as the seasonal component $\tilde{X}_S$ and the low-frequency segments as the trend component X_T

$$\tilde{X}_S, X_T = \mathcal{F}^{-1}(\mathcal{F}(\mathcal{X})[:\beta], \mathcal{F}(\mathcal{X})[\beta:]) \tag{1}$$

where $\mathcal{F}$ denotes the FFT, $\mathcal{F}^{-1}$ is its inverse operation, $\mathcal{F}(\mathcal{X}) \in \mathbb{R}^E$, $E = \lfloor L/2 \rfloor + 1$ is the number of frequencies, and β is a hyperparameter representing the number of low frequency components selected. As the seasonal component $\tilde{X}_S$, composed of high-frequency signals, often contains a notable amount of noise, we find it out by selecting the Top-K frequencies with the highest amplitudes. This process yields the ultimate seasonal component X_S and the residual component X_N. It can be formalized as

$$\begin{aligned} X_S &= \mathcal{F}^{-1}(\text{SelectF}(\text{TopK}(\mathcal{A}(\mathcal{F}(\tilde{X}_S))))) \\ X_N &= \mathcal{F}^{-1}(\text{SelectF}(i{-}\text{TopK}(\mathcal{A}(\mathcal{F}(\tilde{X}_S))))) \end{aligned} \tag{2}$$

where $\mathcal{A}(F) = \sqrt{F_r^2 + F_i^2}$, F_r and F_i represent the corresponding real and imaginary parts, respectively. *SelectF*() denotes selection of the frequency corresponding to the foremost amplitudes.

Recurrent Variational Inference: In accordance with the Gibbs phenomenon as discussed by Gottlieb and Shu [44], the approximation error incurred by the Fourier Transform tends to reach a limit of approximately 9% of the jump discontinuity in time series. Therefore, directly extracting seasonal and trend components from original time series containing anomalous signals via frequency decomposition (Fourier Transform) is a challenging problem. Inspired by effectiveness of recent generative work [45] which utilizes recurrent inference in generating temporal dynamics, we adopt variational inference strategies to capture the inherent regularity (seasonality and trend) of the time series and reduce the impact of anomalous signals. Such design (variational autoencoders) makes the learning of latent representations more tractable. To be specific, we propose two sequentially structured variational inference networks (R-VAEs) to learn the seasonal and trend latent representations, respectively.

As depicted in Fig. 1, in the encoders of the seasonal and trend R-VAEs, we utilize two gated recurrent units to incorporate the latent representations from previous timestamps. For the sake of formula simplicity, we have omitted the superscripts S/T for seasonality and trend in the formula. The encoder of the R-VAE is defined as follows:

$$\begin{aligned} \tilde{\mu}_t &= W_\mu \cdot x_t + b_\mu \\ o_t^\mu &= \text{Sigmoid}\big(W_\mu^o \cdot [\mu_{t-1}, \tilde{\mu}_t] + b_\mu^o\big) \\ \mu_t &= \big(1 - o_t^\mu\big) \odot \mu_{t-1} + o_t^\mu \odot \tilde{\mu}_t \\ \tilde{\sigma}_t &= W_\sigma \cdot x_t + b_\sigma \\ o_t^\sigma &= \text{Sigmoid}\big(W_\sigma^o \cdot [\sigma_{t-1}, \tilde{\sigma}_t] + b_\sigma^o\big) \\ \sigma_t &= \big(1 - o_t^\sigma\big) \odot \sigma_{t-1} + o_t^\sigma \odot \tilde{\sigma}_t \\ z_t &\sim N(\mu_t, \sigma_t) \end{aligned} \tag{3}$$

where W_μ, W_μ^o, W_σ, W_σ^o, b_μ, b_μ^o, b_σ, and b_σ^o are all trainable parameters, and $[\cdot, \cdot]$ denotes the concatenation operation of two variables. After obtaining the decomposed representations of seasonality and trend, Z_S and Z_T, they are, respectively, passed through their corresponding decoders for reconstruction, resulting in $\hat{X}_S$ and $\hat{X}_T$. Both decoders are constructed using a multilayer perceptron. Specifically

$$\begin{aligned} \hat{X}_S &= \text{MLP}_S(Z_S) \\ \hat{X}_T &= \text{MLP}_T(Z_T) \end{aligned} \tag{4}$$

where $\text{MLP}_S()$ and $\text{MLP}_T()$ represent two different multilayer perceptrons.

In (15), $KL(q_{\theta^{S^E}}(Z_S|\mathcal{X})||p(Z_S))$ and $KL(q_{\theta_T^E}(Z_T|\mathcal{X})||p(Z_T))$ represent the Kullback–Leibler (KL) divergence constraints for the seasonal and trend components, respectively. We use the Monte Carlo sampling method to estimate the KL divergence, assuming that the prior distribution follows a standard normal distribution $\mathcal{N}(0, 1)$, to improve tractability of latent representations and computational efficiency. Our goal is to learn more informative seasonal and trend representations from the original time series data, for which we impose corresponding reconstruction constraints through X_S and X_T. Based on the LaST [12] setup, the seasonal and trend reconstruction loss $\mathcal{L}_{\text{rec}}^{ST}$ is redefined as

$$\begin{aligned} \mathcal{L}_{\text{rec}}^{ST} = &- \sum_{\tau=1}^{L-1} \left\| A_{\hat{X}_S\hat{X}_S}(\tau) - A_{X_SX_S}(\tau) \right\|^2 \\ &+ \frac{\sum_{i=1}^{L-1} \big(\hat{x}_{T,i} - \hat{x}_{T,i-1}\big)\big(x_{T,i} - x_{T,i-1}\big)}{\sqrt{\sum_{i=1}^{L-1} \big(\hat{x}_{T,i} - \hat{x}_{T,i-1}\big)}\sqrt{\sum_{i=1}^{L-1} \big(x_{T,i} - x_{T,i-1}\big)}} \\ &- \left\| \hat{X}_S - X_S \right\|^2 - \left\| \hat{X}_T - X_T \right\|^2 \end{aligned} \tag{5}$$

where $A_{X_SX_S}(\tau) = \sum_{t=1}^{L-\tau} (x_{S,t} - \bar{x}_S)(x_{S,t+\tau} - \bar{x}_S)$ denotes the autocorrelation coefficient with lagged value τ. Overall, the first term reflects the similarity between every instant value

in time series data and its τ-lagged value. It captures the periodic characteristic similarities between two time series. The second term represents temporal correlation coefficient, which measures simultaneity of two temporal periods' rising or falling patterns. The final two terms denote point-wise reconstruction losses of seasonal and trend components.

Sparse Variational Inference: In the traditional VAE framework, the prior distribution of the latent variable Z follows a standard Gaussian distribution. However, for residual data containing anomalous signals, the occurrence of sudden events is discrete and random, making traditional methods unsuitable. We adopt the spike-and-slab distribution, of which sparse coding assumption (Sparse-VAE) better fits the characteristics of the residual component, to fit the latent variables. The spike component places high probability mass at zero, effectively suppressing most residuals to reflect normal fluctuations, while the slab component endows the model with the capacity to represent infrequent but large deviations. To be specific

$$\begin{aligned} c &\sim \text{Bernoulli}(\text{Sigmoid}(W_c \cdot \mathcal{X} + b_c)) \\ \mu_N &= W_\mu^N \cdot \mathcal{X} + b_\mu^N \\ \sigma_N &= W_\sigma^N \cdot \mathcal{X} + b_\sigma^N \\ r_1 &\sim N\left(0, \frac{1}{s}\sigma_N\right) \\ r_2 &\sim N(\mu_N, \sigma_N) \\ Z_N &= (1-c) \odot r_1 + c \odot r_2 \end{aligned} \tag{6}$$

where W_c, W_μ^N, W_σ^N, b_c, b_μ^N, and b_σ^N are all trainable parameters, s is a hyperparameter, and r_1 and r_2 follow the spike and slab distributions, respectively. After obtaining the residual representation Z_N, we accomplish the residual reconstruction through a multilayer perceptron

$$\hat{X}_N = \text{MLP}_N(Z_N). \tag{7}$$

Meanwhile, we apply reconstruction constraints using X_N, which is generated via a frequency domain decomposition

$$\mathcal{L}_{\text{rec}}^N = -\left\|\hat{X}_N - X_N\right\|^2. \tag{8}$$

In (15), the residual component's KL divergence constraint $KL(q_{\theta_N^E}(Z_N|\mathcal{X})||p(Z_N))$ is implemented. Different from traditional VAE, the prior distribution of Z_N is set as: $p(Z_N) \sim (1-\alpha) \cdot N(0, [1/100]I) + \alpha \cdot N(0, I)$, where $\alpha = \{\alpha_i\}$, and each α_i is drawn from a Bernoulli distribution, i.e., $\{\alpha_i\} \sim \text{Bernoulli}(1/2)$.

C. Multicomponent Communication Aware Decoder

In addition to the aforementioned component-specific reconstructions, we propose to enhance the representation consistency via a multicomponent communication aware decoder, which is composed of a multihead attention module and a gated decomposition-aware reconstruction module.

Multihead Cross Attention: To better utilize the residual representations for time series reconstruction, we propose three multihead cross-attention modules to fuse the features of the seasonal, trend, and residual components. Specifically, the seasonal/trend representation and the residual representation

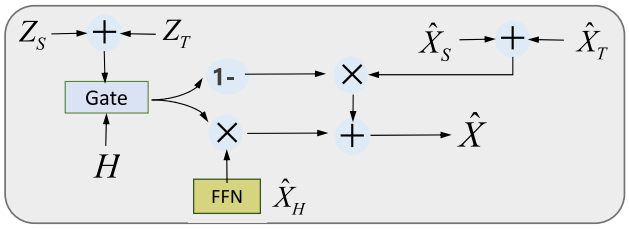

Fig. 2. Overview of proposed decomposition-aware reconstruction architecture.

are regarded as queries and keys, respectively, obtaining interaction-aware attention scores. The final aggregation process can be represented as

$$H_{S/T} = \text{Softmax}\left(\frac{Q_{S/T}K_N^\top + Q_N K_{S/T}^\top}{\sqrt{d}}\right) \cdot \left(V_{S/T} + \omega \cdot V_N\right) \tag{9}$$

where $\{Q_S, Q_T, Q_N\}$, $\{K_S, K_T, K_N\}$, and $\{V_S, V_T, V_N\}$ represent the query matrices, key matrices, and value matrices of the seasonal, trend, and residual components, respectively. They are all obtained by mapping $\{Z_S, Z_T, Z_N\}$ through a linear layer, and ω is an adjustable hyperparameter. H_S and H_T are further aggregated via a multihead cross-attention layer to obtain the final aggregation

$$H = \text{Softmax}\left(\frac{Q_{HS}K_{HT}^\top + Q_{HT}K_{HS}^\top}{\sqrt{d}}\right) \cdot (V_{HS} + V_{HT}), \tag{10}$$

where $\{Q_{HS}, Q_{HT}\}$, $\{K_{HS}, K_{HT}\}$, and $\{V_{HS}, V_{HS}\}$ represent the cross query matrices, cross key matrices, and cross value matrices, respectively. These matrices are derived from H_S and H_T through a linear layer mapping.

Finally, the representation cross-reconstruction is accomplished using a multilayer perceptron

$$\hat{X}_H = \text{MLP}_H(H). \tag{11}$$

The representation cross-reconstruction loss is defined as

$$\mathcal{L}_{\text{rec}}^H = -\left\|\hat{X}_H - \mathcal{X}\right\|^2. \tag{12}$$

Decomposition-Aware Reconstruction: Both regular and irregular changes jointly influence the modeling of time signals. We leverage latent representation decomposition to separately model the regular components (seasonal and trend components) and irregular components (residual components) in the time signal. $\hat{X}_H$ captures the overall temporal dynamics by reconstructing it with the fusion of representations. $\hat{X}_S$ and $\hat{X}_T$ capture the regularity of the time series. To capture the contributions of different components, a gated decomposition-aware aggregation strategy is designed to obtain the final reconstructed time series data. The structure can be seen in Fig. 2. More specifically, we obtain a gating signal η through a sigmoid function, then achieving the final gated reconstruction aggregation

$$\begin{aligned} \eta &= \text{Sigmoid}\left(W_\eta \cdot [H, Z_S + Z_T] + b_\eta\right) \\ \hat{X} &= \eta \odot \hat{X}_H + (1-\eta) \odot \left(\hat{X}_S + \hat{X}_T\right) \end{aligned} \tag{13}$$

where W_η and b_η are trainable parameters. The gating mechanism allows for distinguishing the importance of different reconstructed components. The loss function is defined as

$$\mathcal{L} = \left\| \hat{\mathcal{X}} - \mathcal{X} \right\|^2 - \mathcal{L}_{\text{ELBO}}. \tag{14}$$

TRAdetector performs representation decomposition modeling by separately learning the representations for seasonal, trend, and residual components. Under the framework of time series decomposition, we maximize the likelihood function:

Finally, we have the evidence lower bound (ELBO)

$$\begin{aligned}
&\mathcal{L}_{\text{ELBO}} = \\
&\frac{1}{L}\sum_{t=1}^{L}\Bigg(\underbrace{E_{q_{\theta_S^E}(z_t^S|x_{0:t})q_{\theta_T^E}(z_t^T|x_{0:t})q_{\theta_N^E}(z_t^N|x_t)}\left[\log p_{\theta^D}(x_t|z_t^S, z_t^T, z_t^N)\right]}_{\text{Representation Cross-Reconstruction Loss}} \\
&\underbrace{+E_{q_{\theta_S^E}(z_t^S|x_{0:t})}\left[\log p_{\theta_S^D}(x_t^S|z_t^S)\right] + E_{q_{\theta_T^E}(z_t^T|x_{0:t})}\left[\log p_{\theta_T^D}(x_t^T|z_t^T)\right]}_{\text{Seasonal and Trend Reconstruction Loss}}\Bigg) \\
&\underbrace{+E_{q_{\theta_N^E}(Z_N|\mathcal{X})}\left[\log p_{\theta_N^D}(X_N|Z_N)\right]}_{\text{Residual Reconstruction Loss}} \\
&\underbrace{-KL\left(q_{\theta_S^E}(Z_S|\mathcal{X})||p(Z_S)\right) - KL\left(q_{\theta_T^E}(Z_T|\mathcal{X})||p(Z_T)\right) - KL\left(q_{\theta_N^E}(Z_N|\mathcal{X})||p(Z_N)\right)}_{\text{KL Divergence Constraint}}
\end{aligned} \tag{15}$$

where θ_S^E, θ_T^E, and θ_N^E, respectively, represent the encoder parameters for the seasonal, trend, and residual components; θ_S^D, θ_T^D, and θ_N^D, respectively, denote the decoder parameters for the seasonal, trend, and residual components; θ^D represents the parameters for the cross-representation decoder. The derivations are provided in the Appendix.

The overall procedures are summarized in Algorithm 1.

Algorithm 1 Overall Algorithm of TRAdetector

Input: Multivariate time series $\mathcal{X} \in \mathbb{R}^{L\times D}$ for the training set, and Multivariate time series $\mathcal{X}' \in \mathbb{R}^{L\times D}$ for the test set; max epoch *Epoch*; batch size B.

Use the uniform distribution to initialize model parameters $\theta \sim U(-1, 1)$.

Output: Anomaly detection results $\hat{\mathcal{Y}}'$ for the test set time series.

/* Training */

for $epoch = 1, 2, \ldots, Epoch$ **do**

  $X_S, X_T, X_N = Frequency{-}Decomp(\mathcal{X})$.

  $\hat{X}_S, Z_S = R{-}VAE(\mathcal{X})$.

  $\hat{X}_T, Z_T = R{-}VAE(\mathcal{X})$.

  $\hat{X}_N, Z_N = Sparse{-}VAE(\mathcal{X})$.

  $H_S = Cross{-}Attention(Z_S, Z_N)$.

  $H_T = Cross{-}Attention(Z_T, Z_N)$.

  $H = Cross{-}Attention(H_S, H_T)$.

  Generate the reconstructed sequence $\hat{X}_H$ using Eq. (12).

  Implementing the gated aggregation of reconstruction item $\hat{X}$ using Eq. (14).

  Update model parameters by optimizing the loss function in Eq. (15).

end for

/* Testing */

$X'_S, X'_T, X'_N = Frequency{-}Decomp(\mathcal{X}')$.

$\hat{X}'_S, Z'_S = R{-}VAE(\mathcal{X}')$.

$\hat{X}'_T, Z'_T = R{-}VAE(\mathcal{X}')$.

$\hat{X}'_N, Z'_N = Sparse{-}VAE(\mathcal{X}')$.

$H'_S = Cross{-}Attention(Z'_S, Z'_N)$.

$H'_T = Cross{-}Attention(Z'_T, Z'_N)$.

$H' = Cross{-}Attention(H'_S, H'_T)$.

Generate the reconstructed sequence $\hat{X}'_H$ using Eq. (12).

Implementing the gated aggregation of reconstruction item $\hat{X}'$ using Eq. (14).

Calculate the anomaly score at each timestamp between the original sequence $\mathcal{X}'$ and the reconstructed sequence $\hat{X}'$ using MSE.

Calculate the anomaly scores based on SPOT algorithm to complete the anomaly annotation $\hat{\mathcal{Y}}'$ for each timestamp.

IV. Experiments

A. Experimental Settings

Datasets: To explore the potential of TRAdetector for time series anomaly detection, we conduct comprehensive evaluations on benchmark real-world IoT datasets: 1) pool server metrics (PSM) [46], which consists of data collected from multiple application server nodes of eBay, including 13 weeks of training data and 8 weeks of testing data; 2) server machine dataset (SMD) [34] is server data from an Internet company, including data from 28 server machines and 38 sensors over a span of 5 weeks; and 3) SWaT (Secure Water Treatment) [47] is collected from a scaled water treatment testing platform with 51 sensors, over a period of 11 days.

Baselines: We conduct extensive comparisons of TRAdetector with 15 benchmark methods, including probability-based methods: Sampling, COPOD [22], ECOD [23]; linear transformation-based methods: OCSVM [24], PCA [25]; proximity-based methods: kNN [26], CBLOF [27], HBOS [28]; outlier-based methods: IForest [29], LODA [30]; and deep neural network-based methods: VAE [31], Lstm-AE [32], MTAD-GAT [33], TFAD [16], Anomaly Transformer [39], DCdetector [8], and D^{3}R [40].

Evaluation Criteria: We adopt three widely used evaluation metrics, such as Precision, Recall, and F1 score. In particular, due to the imbalance of the labels, the F1 score serves as a more convincing metric. Here, we emphasize F1 score without point adjustment, because the traditional point adjustment method (used in DCdetector [8]) suggests that if one point within a continuous segment of anomalies is correctly detected, then all anomalous points in the same segment are also considered to be correctly detected. However, extensive researches (e.g., D^3R [40]) have demonstrated that point adjustment is unreasonable and creates an illusion of model progress, unfairly assigning greater significance to longer anomalous events. Therefore, we adhere to the same setting without point adjustment, i.e., each point in a continuous segment should be judged, which is required in widespread applications, e.g., industrial monitoring. More specifically, following D^{3}R, we run the SPOT algorithm to label each timestamp.

TABLE II
RESULTS IN THE THREE REAL-WORLD DATASETS. THE HIGHER VALUES FOR ALL METRICS REPRESENT THE BETTER PERFORMANCE, AND THE BEST F1 SCORES ARE HIGHLIGHTED IN BOLD

Method	PSM			SMD			SWaT			Average
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	F1
Sampling	0.8439	0.5165	0.6408	0.7453	0.3144	0.4422	0.6062	0.8466	0.7065	0.5965
COPOD [22]	0.7602	0.3175	0.4479	0.6676	0.1366	0.2268	0.9876	0.1180	0.2108	0.2951
ECOD [23]	0.7460	0.3384	0.4656	0.7398	0.1615	0.2651	0.9761	0.1151	0.2059	0.3122
OCSVM [24]	0.8761	0.4744	0.6155	0.0000	0.0000	0.0000	0.6196	0.7558	0.6810	0.4321
PCA [25]	0.9220	0.3771	0.5353	0.8388	0.4019	0.5434	0.6358	0.7218	0.6761	0.5849
kNN [26]	0.5317	1.0000	0.6943	0.6988	0.3368	0.4546	0.0000	0.0000	0.0000	0.3830
CBLOF [27]	0.5990	0.9845	0.7449	0.8667	0.3352	0.4834	0.6308	0.7091	0.6677	0.6320
HBOS [28]	1.0000	0.0654	0.1228	0.5628	0.8007	0.6610	0.5771	0.8049	0.6722	0.4853
IForest [29]	1.0000	0.0335	0.0648	1.0000	0.0937	0.1713	0.6127	0.6280	0.6203	0.2855
LODA [30]	0.9266	0.4017	0.5605	0.5902	0.6618	0.6240	0.6117	0.7014	0.6535	0.6126
VAE [31]	0.6221	0.8772	0.7280	0.8209	0.4349	0.5686	0.6355	0.7218	0.6759	0.6575
LSTM-AE [32]	0.7511	0.7586	0.7548	0.8496	0.4349	0.5753	0.6018	0.7219	0.6564	0.6622
MTAD-GAT [33]	0.7990	0.6014	0.6863	0.8590	0.6769	0.7571	0.6590	0.7751	0.7123	0.7186
TFAD [16]	0.7914	0.7163	0.7520	0.5632	0.9783	0.7149	0.6038	0.8196	0.6953	0.7207
Anomaly Transformer [39]	0.5201	0.8504	0.6455	1.0000	0.0319	0.0619	0.5541	0.5994	0.5759	0.4278
DCdetector [8]	0.6395	0.4682	0.4823	0.7113	0.4463	0.5485	0.6123	0.5050	0.5271	0.5193
D^3R [40]	0.6294	0.9619	0.7609	0.7715	0.9926	0.8682	0.7206	0.8529	**0.7812**	0.8034
TRAdetector	0.7262	0.9708	**0.8309**	0.8032	0.9914	**0.8749**	0.7032	0.9554	0.7634	**0.8231**

TABLE III
ABLATION STUDY OF EACH MODULE OF TRADETECTOR ON THE PSM, SMD, AND SWAT DATASETS

Method	PSM			SMD			SWaT			Average
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	F1
$\text{TRAdetector}_{(VAE)}$	0.7144	0.9428	0.8023	0.7782	0.9882	0.8520	0.6901	0.9310	0.7495	0.8013
$\text{TRAdetector}_{Sparse\text{-}VAE-}$	0.6539	0.8872	0.7512	0.7233	0.8879	0.8117	0.6225	0.8847	0.7396	0.7675
$\text{TRAdetector}_{(Trend)R\text{-}VAE-}$	0.7032	0.9577	0.7992	0.7638	0.9726	0.8412	0.6912	0.9120	0.7392	0.7932
$\text{TRAdetector}_{(Seasonal)R\text{-}VAE-}$	0.6840	0.9246	0.7821	0.7434	0.9640	0.8330	0.6722	0.9130	0.7422	0.7858
$\text{TRAdetector}_{cross\text{-}attention-}$	0.6452	0.8943	0.7639	0.7496	0.9012	0.8293	0.6187	0.8716	0.7371	0.7768
TRAdetector	**0.7262**	**0.9708**	**0.8309**	**0.8032**	**0.9914**	**0.8749**	**0.7032**	**0.9554**	**0.7634**	**0.8231**

Implementation Details: We optimize our proposed approach using the Adam optimizer, with a batch size of 32, a learning rate of 0.001, and a feature vector dimension of 1024. For the frequency domain decomposition module, the parameter β is set to 3, meaning that the three lowest frequency components are extracted to represent the trend component. Meanwhile, for the decomposition of seasonal and residual components, a top-K value of 10 is chosen, indicating that the ten frequencies with the highest amplitudes are selected as seasonal components. Additionally, the parameter *s* is set to 100. To ensure a fair comparison, we adopt the same dataset partitioning strategy as D^3R. All models are trained and evaluated on an NVIDIA Tesla V100 32 GB GPU, ensuring computational efficiency and consistency across experiments.

B. Overall Performance

The experimental results are summarized in Table II, where TRAdetector consistently outperforms existing anomaly detection algorithms in terms of both robustness and accuracy. By disentangling time series into three distinct components—seasonality, trend, and residual—and modeling the underlying distribution characteristics of anomalous signals, TRAdetector enhances its ability to distinguish anomalies from normal patterns. In particular, when compared to the current state-of-the-art algorithm D^3R, TRAdetector achieves notable improvements in the F1 score, with a significant increase of 9.2% on the PSM dataset and a 0.8% improvement on the SMD dataset. Furthermore, for each dataset, we conducted experiments using five different random seeds. Table II reports the mean performance across these five runs. The variance of the F1 score is 0.021 on the PSM dataset, 0.015 on the SMD dataset, and 0.019 on the SWaT dataset. These results highlight the effectiveness of TRAdetector in capturing temporal anomalies and reducing false positives, making it a reliable solution for IoT anomaly detection across real-world applications.

C. Ablation Studies

To assess the contribution of each component within TRAdetector, we conducted ablation experiments on three benchmark datasets, such as PSM, SMD, and SWaT. The detailed experimental results are presented in Table III.

We assessed the impact of the Sparse-VAE module via two ablation settings: 1) its complete removal ($\text{TRAdetector}_{\text{Sparse}-\text{VAE}-}$) and 2) replacement with a standard

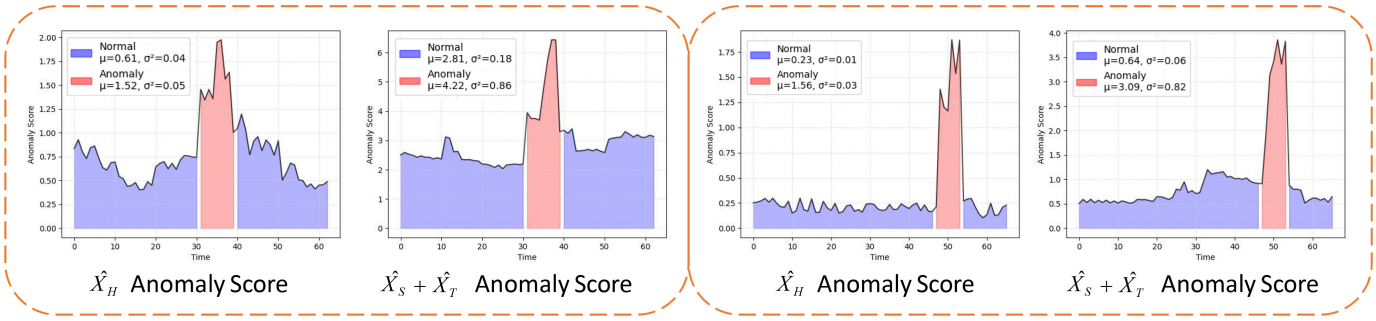

Fig. 3. Visualization of reconstruction errors. The dashed box displays the anomaly scores of the same sample, comparing $\hat{X}_H$ with $\hat{X}_S + \hat{X}_T$. The red area indicates anomalous signals, while the blue area represents normal signals. The figure legend depicts the mean and variance of the normal and anomalous signals.

VAE (TRAdetector$_{(\text{VAE})}$). Removing Sparse-VAE led to an average F1 score drop of 7.2%, highlighting its effectiveness in capturing irregular residual patterns.

Next, we evaluated the role of regular temporal patterns by independently removing the trend and seasonal components from R-VAE, resulting in the variants TRAdetector$_{(\text{Trend})R-\text{VAE}-}$ and TRAdetector$_{(\text{Seasonal})R-\text{VAE}-}$, respectively. Removing the trend component led to a 3.7% drop in F1 score, while excluding the seasonal component caused a 4.7% decline. These results underscore the importance of explicitly modeling regular temporal dynamics, with both trend and seasonal representations proving essential for optimal anomaly detection performance.

Finally, to assess the Multihead Cross Attention module, we replaced it with a simple summation of seasonal, trend, and residual features followed by a linear reconstruction layer (TRAdetector$_{\text{cross}-\text{attention}-}$). This simplification led to an average 6.0% drop in F1 score, highlighting the importance of effective feature interaction for accurate anomaly detection.

The results demonstrate that all components are indispensable—particularly, removing the cross-attention module leads to a sharp decline in performance. This highlights each component's individual contribution and the synergistic effects among the model's components. We argue that unlike simple concatenation followed by a fully connected layer, cross-attention enables the model to adaptively learn which component is most important at each time step, thus facilitating fine-grained, conditional interactions among components.

D. Reconstruction Error Analysis

TRAdetector employs a gating unit to effectively fuse the high-frequency residual component $\hat{X}_H$ with the sum of the seasonal and trend components $(\hat{X}_S + \hat{X}_T)$, producing the final reconstructed sequence $\hat{X}$. By computing the reconstruction error between $\hat{X}$ and the original input sequence $\mathcal{X}$, the model identifies and filters out anomalous signals. Fig. 3 provides a visualization of the reconstruction errors, measured by mean squared error (MSE), across different components on the SWaT dataset. The results clearly indicate that abnormal data points exhibit significantly higher reconstruction errors compared to normal points. This discrepancy underscores the model's ability to effectively differentiate anomalous patterns by leveraging disentangled component-wise reconstruction, thereby enhancing anomaly detection accuracy.

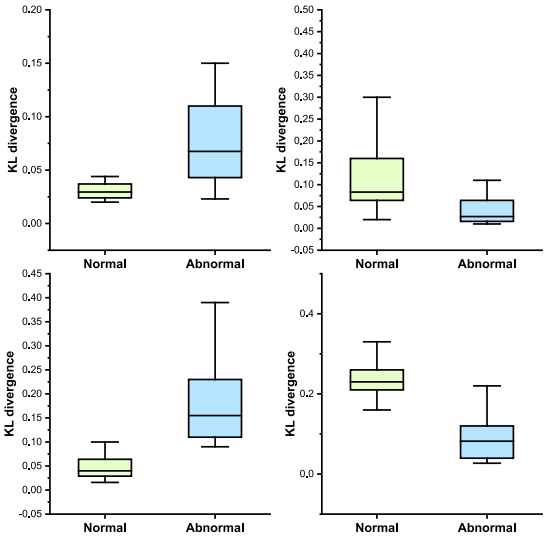

Fig. 4. Disparities in the KL divergence of the representation between normal and anomalous signals.

E. Distribution Difference Comparison

To evaluate the discriminative power of residual representations in identifying anomaly signals, we compute the KL divergence between a reference uniform distribution (centered at a mean of 0.5) and the residual representations corresponding to both normal and anomaly signals. As illustrated in Fig. 4, the KL divergence values exhibit a significant disparity between the residual representations of normal and anomaly signals when compared to the uniform distribution. This observation suggests that the residual representations effectively capture distinct patterns associated with anomalies, reinforcing their utility in differentiating anomalous behavior from normal temporal dynamics.

F. Hyperparameter Experiments

Anomaly detection in the IoT domain is highly complex due to the entanglement of multidimensional information and pervasive noise. In practice, the choice of hyperparameters not only affects the decomposition stability of frequency-domain models but also directly determines overall

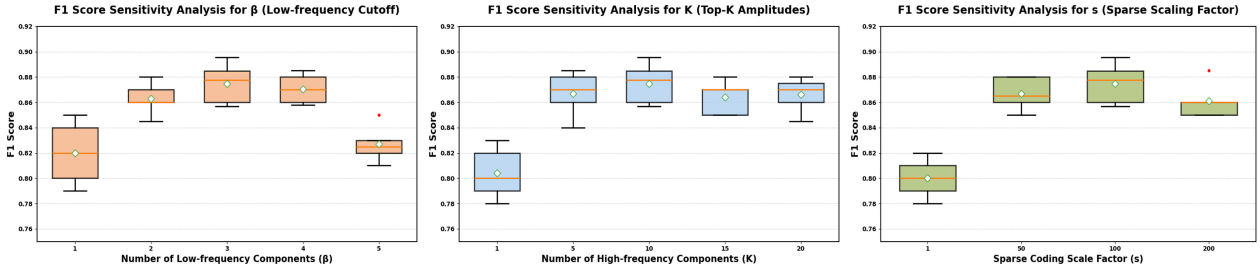

Fig. 5. Impact of hyperparameter settings on decomposition stability and anomaly detection performance. (a) Relationship between low-frequency cutoff threshold (β) and decomposition stability as well as anomaly detection effectiveness (measured by F1-score). (b) Effect of varying the number of high-frequency amplitudes (K) retained during seasonal signal reconstruction on the overall detection performance (F1-score). (c) Impact of different sparse coding scaling factors (s) on the sparsity of residual components and anomaly detection capability.

Fig. 6. Visualization of anomaly detection capabilities by different reconstruction components. The red area indicates anomalous time points. The green area represents the anomaly scores of the reconstructed component $\hat{X}_H$; the yellow area represents the anomaly scores of the seasonal reconstructed component $\hat{X}_S$; and the purple area represents the anomaly scores of the trend reconstructed component $\hat{X}_T$.

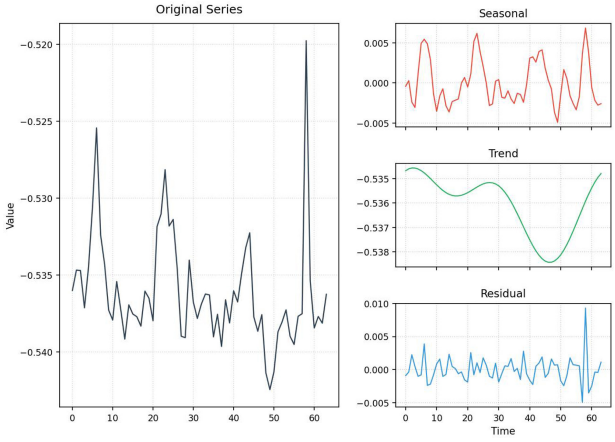

Fig. 7. Visualization of the frequency-domain decomposition of a time series. The left panel shows the original input time series; the right panels, from top to bottom, display the decomposed seasonal component, trend component, and residual (noise) component.

detection performance. To systematically analyze how the key hyperparameters influence TARdetector's effectiveness, we conducted a series of controlled experiments on the SMD dataset, focusing on three parameters: 1) the low-frequency cutoff threshold β; 2) the high-frequency amplitude cutoff K; and 3) the sparse-coding scaling factor s.

We first examined how β impacts decomposition stability and anomaly detection performance. To isolate its effect, we fixed all other hyperparameters and varied β over the set $1, 2, 3, 4, 5$. To evaluate robustness, we ran each setting five times with different random seeds. As shown in Fig. 5(a), when $\beta = 1$, the cutoff is too conservative, causing the trend component to retain excessive long-term variation and leaving much trend information within the seasonal component. As β increases, the trend component gradually incorporates high-frequency noise, destabilizing the decomposition, which distorts the residuals and degrades the F1 score. At $\beta = 3$, we observe an ideal balance between trend smoothness and detection sensitivity, yielding the highest F1 performance.

Next, we analyzed the influence of K on seasonal reconstruction and overall detection ability. Holding other parameters constant, we tested K in $\{1, 5, 10, 15, 20\}$. As illustrated in Fig. 5(b), setting $K = 1$ retains only the single most prominent frequency, discarding important seasonal patterns and severely weakening detection performance. Conversely, overly large K values capture excessive weak noise frequencies, causing minor performance fluctuations. We found that $K = 10$ strikes the best tradeoff—effectively capturing key seasonal features while filtering out noise—thereby maximizing F1.

Finally, we evaluated how the scaling factor s affects residual sparsity and detection outcomes. With all other hyperparameters held constant, we varied s over the set $1, 50, 100, 200$. Fig. 5(c) shows that at $s = 1$, the prior reduces to a Gaussian distribution, preventing the residual from distinguishing anomalies effectively. When s is too large, the residual becomes over-smoothed, increasing false positives. The results indicate that $s = 100$ provides the best compromise: maintains a high recall for true anomalies while controlling noise-induced false alarms, achieving optimal overall performance. Further experiments on SWaT and WADI reproduce similar optimal ranges, demonstrating that these settings are robust across datasets rather than tailored to a single data distribution.

Through this comprehensive analysis, we identify the optimal configuration of β, K, and s for real-world scenarios and demonstrate that, under these settings, TARdetector consistently delivers superior anomaly-detection performance.

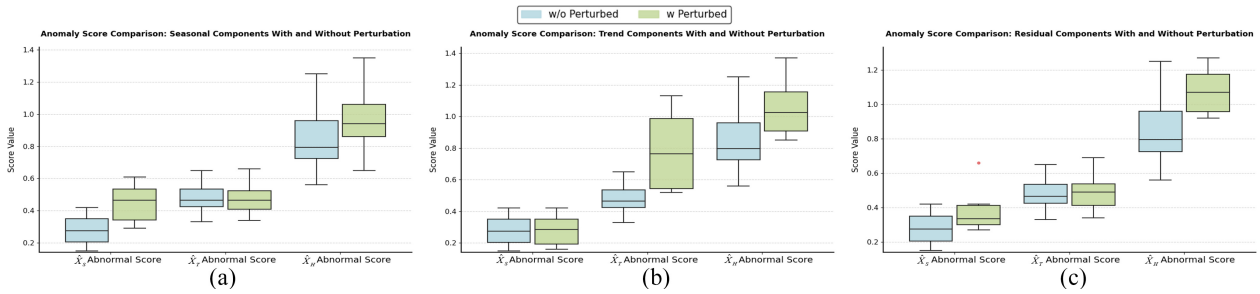

Fig. 8. Comparison of results from the anomalous signal perturbation experiment. (a)–(c) illustrate how the distributions of anomaly scores change after injecting noise into the seasonal component, trend component, and residual component, respectively.

These findings offer practical guidance for the deployment and establish a solid foundation for future model refinements.

G. Contribution of Different Components

We further investigate the contributions of different reconstruction components in enhancing anomaly detection performance. Specifically, we analyze how seasonal, trend, and residual components respond to various types of anomalies. As illustrated in Fig. 6, during the first anomalous sequence, the seasonal reconstruction component $\hat{X}_S$ exhibits a stronger ability to detect anomalies, suggesting its heightened sensitivity to periodic fluctuations and deviations from expected seasonal patterns. In contrast, during the second anomalous sequence, the trend reconstruction component $\hat{X}_T$ plays a more prominent role in detecting anomalies, particularly in the latter part of the sequence. This observation underscores its effectiveness in identifying long-term trend shifts, which may not be immediately evident in shorter time spans. These findings highlight the importance of disentangling different temporal components in anomaly detection, as each component captures distinct types of irregularities. By leveraging their complementary strengths, TRAdetector enhances its ability to detect diverse anomalies, improving robustness across various temporal patterns.

H. Anomalous Signal Perturbation

To further evaluate TRAdetector's sensitivity to perturbations in different signal components during anomaly detection, we designed a series of targeted experiments. First, we visualized the frequency-domain decomposition of time series in Fig. 7 to confirm the model's ability to disentangle their constituent components. As shown, the left panel displays the original time series, while the right panel vertically arranges the corresponding seasonal, trend, and residual components. Of particular note is the residual component's spike-and-slab sparse distribution, which clearly contains extreme anomalous events—further evidence of the model's effectiveness in distinguishing anomalous patterns from normal behavior.

Next, to quantify the contribution of each component to the overall detection sensitivity, we independently injected noise into the seasonal, trend, and residual components of the same time series (Fig. 8) and compared the resulting anomaly-score trajectories. When perturbations were applied solely to the seasonal or trend component, the anomaly score of the perturbed component changed significantly and induced marked fluctuations in the global score, while the unperturbed components remained largely stable. This indicates that the seasonal and trend components each contribute to anomaly detection with a high degree of independence and specificity. In contrast, perturbing only the residual component produced the most pronounced change in the global anomaly score; the seasonal-component score exhibited a minor response, and the trend-component score remained essentially unchanged. These observations confirm the critical role of the residual component in capturing sudden or irregular anomalies.

Taken together, these results underscore the effectiveness of frequency-domain decomposition in anomaly detection tasks and demonstrate that TRAdetector can accurately identify and sensitively respond to perturbations across different signal types. This sensitivity analysis provides empirical support for the model's anomaly-detection and localization capabilities in real-world IoT scenarios, thereby enhancing its credibility and interpretability.

I. Runtime Analysis

To assess TRAdetector's suitability for production environments, we conducted a systematic comparison of training time, inference time, and model size across various neural network-based models using the SMD dataset. The results are summarized in Table IV.

Both the training and test sets of SMD encompass approximately 16 days of data. All experiments were performed on an NVIDIA Tesla V100 GPU server with 32 GB of memory. Our findings show that every model completed training in under 10 min, satisfying the rapid-deployment and efficient-maintenance requirements of real-world applications. Further analysis reveals that, relative to the Anomaly Transformer and D^3R models, TRAdetector achieves significant gains in both training and inference throughput. This advantage stems primarily from limiting the use of attention mechanisms to the feature-fusion stage, thereby substantially reducing computational complexity.

In terms of model size, TRAdetector occupies 105.45 MB, which remains within acceptable bounds given the increasing availability of computational resources. Notably, despite having a parameter count comparable to D^3R, TRAdetector accelerates training by approximately 52.3% and inference by approximately 46.9%.

TABLE IV
RUN TIMES IN THE SMD. THE LOWER VALUES REPRESENT THE BETTER PERFORMANCE. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Method	Training Time (s)	Inference Time (s)	Model Size (MB)
Anomaly Transformer	297.89	79.08	**28.15**
D^3R	276.32	86.77	109.35
TRAdetector	**181.36**	**59.06**	105.45

Overall, these results demonstrate that TRAdetector achieves an exceptional balance among speed, model footprint, and detection performance. This underscores its potential for rapid deployment and efficient operation in industrial IoT settings, providing robust empirical support for its use in large-scale anomaly-detection tasks.

V. CONCLUSION

In this article, we propose TRAdetector, an unsupervised framework for time series anomaly detection based on disentangled representation learning. By decomposing time series into seasonal, trend, and residual components and applying a gated reconstruction strategy, TRAdetector effectively highlights anomalies. Variational inference with Gaussian and spike-and-slab priors models both consistent and irregular patterns, enhancing detection accuracy. Experiments on three benchmarks show TRAdetector outperforms or matches state-of-the-art methods. However, the aforementioned datasets' anomaly patterns can be constrained by domain-specific characteristics, and they may not fully capture the diversity of long-term drifts and nonstationarities. In addition, TRAdetector currently lacks mechanisms for generalization, adversarial robustness and privacy guarantees—critical for real-world deployment in sensitive IoT applications. Future work will focus on integrating privacy-preserving inference, adversarial defense, and trustworthy uncertainty quantification, moving toward more secure and reliable anomaly detection systems.

APPENDIX

DERIVATION OF LIKELIHOOD FUNCTION

The detailed derivation of the likelihood function is presented as follows:

$$
\begin{aligned}
&\log p(\mathcal{X}) = \log p(X_S) + \log p(X_T) + \log p(X_N) \\
&= \log \int_{Z_S}\int_{Z_T}\int_{Z_N} p_{\theta^D}(\mathcal{X}, Z_S, Z_T, Z_N) dZ_S dZ_T dZ_N \\
&+ \log \int_{Z_S} p_{\theta_S^D}(X_S, Z_S) dZ_S + \log \int_{Z_T} p_{\theta_T^D}(X_T, Z_T) dZ_T \\
&+ \log \int_{Z_N} p_{\theta_N^D}(X_N, Z_N) dZ_N \\
&= \log \int_{Z_S}\int_{Z_T}\int_{Z_N} \\
&\frac{q_{\theta_S^E}(Z_S|\mathcal{X}) q_{\theta_T^E}(Z_T|\mathcal{X}) q_{\theta_N^E}(Z_N|\mathcal{X}) p_{\theta^D}(\mathcal{X}, Z_S, Z_T, Z_N)}{q_{\theta_S^E}(Z_S|\mathcal{X}) q_{\theta_T^E}(Z_T|\mathcal{X}) q_{\theta_N^E}(Z_N|\mathcal{X})} dZ_S dZ_T dZ_N \\
&+ \log \int_{Z_S} \frac{q_{\theta_S^E}(Z_S|\mathcal{X}) p_{\theta_S^D}(X_S, Z_S)}{q_{\theta_S^E}(Z_S|\mathcal{X})} dZ_S \\
&+ \log \int_{Z_T} \frac{q_{\theta_T^E}(Z_T|\mathcal{X}) p_{\theta_T^D}(X_T, Z_T)}{q_{\theta_T^E}(Z_T|\mathcal{X})} dZ_T \\
&+ \log \int_{Z_N} \frac{q_{\theta_N^E}(Z_N|\mathcal{X}) p_{\theta_N^D}(X_N, Z_N)}{q_{\theta_N^E}(Z_N|\mathcal{X})} dZ_N \\
&= \frac{1}{L}\sum_{t=1}^{L}\Bigg(\log E_{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)} \\
&\left[\frac{p_{\theta^D}(x_t, z_t^S, z_t^T, z_t^N)}{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)} \right] + \log E_{q_{\theta_S^E}(z_t^S|x_{0:t})}\left[\frac{p_{\theta_S^D}(x_t^S, z_t^S)}{q_{\theta_S^E}(z_t^S|x_{0:t})} \right] \\
&+ \log E_{q_{\theta_T^E}(z_t^T|x_{0:t})}\left[\frac{p_{\theta_T^D}(x_t^T, z_t^T)}{q_{\theta_T^E}(z_t^T|x_{0:t})} \right] \Bigg) + \log E_{q_{\theta_N^E}(Z_N|\mathcal{X})}\left[\frac{p_{\theta_N^D}(X_N, Z_N)}{q_{\theta_N^E}(Z_N|\mathcal{X})} \right] \\
&\geq \frac{1}{L}\sum_{t=1}^{L}\Bigg(E_{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)} \\
&\left[\log \frac{p_{\theta^D}(x_t, z_t^S, z_t^T, z_t^N)}{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)} \right] \\
&+ E_{q_{\theta_S^E}(z_t^S|x_{0:t})}\left[\log \frac{p_{\theta_S^D}(x_t^S, z_t^S)}{q_{\theta_S^E}(z_t^S|x_{0:t})} \right] + E_{q_{\theta_T^E}(z_t^T|x_{0:t})}\left[\log \frac{p_{\theta_T^D}(x_t^T, z_t^T)}{q_{\theta_T^E}(z_t^T|x_{0:t})} \right] \Bigg) \\
&+ E_{q_{\theta_N^E}(Z_N|\mathcal{X})}\left[\log \frac{p_{\theta_N^D}(X_N, Z_N)}{q_{\theta_N^E}(Z_N|\mathcal{X})} \right] \qquad (16)
\end{aligned}
$$

$$
\begin{aligned}
&\mathcal{L}_{\text{ELBO}} \\
&= \frac{1}{L}\sum_{t=1}^{L}\Bigg(E_{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)} \\
&\left[\log \frac{p_{\theta^D}(x_t, z_t^S, z_t^T, z_t^N)}{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)} \right] \\
&+ E_{q_{\theta_S^E}(z_t^S|x_{0:t})}\left[\frac{p_{\theta_S^D}(x_t^S, z_t^S)}{q_{\theta_S^E}(z_t^S|x_{0:t})} \right] + E_{q_{\theta_T^E}(z_t^T|x_{0:t})}\left[\frac{p_{\theta_T^D}(x_t^T, z_t^T)}{q_{\theta_T^E}(z_t^T|x_{0:t})} \right] \Bigg) \\
&+ E_{q_{\theta_N^E}(Z_N|\mathcal{X})}\left[\log \frac{p_{\theta_N^D}(X_N, Z_N)}{q_{\theta_N^E}(Z_N|\mathcal{X})} \right] \\
&= \frac{1}{L}\sum_{t=1}^{L}\Bigg(E_{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)} \\
&\left[\log \frac{p_{\theta^D}(z_t^S) p_{\theta^D}(z_t^T) p_{\theta^D}(z_t^N) p_{\theta^D}(x_t|z_t^S, z_t^T, z_t^N)}{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)} \right] \\
&+ E_{q_{\theta_S^E}(z_t^S|x_{0:t})}\left[\log \frac{p_{\theta_S^D}(z_t^S) p_{\theta_S^D}(x_t^S|z_t^S)}{q_{\theta_S^E}(z_t^S|x_{0:t})} \right] \\
&+ E_{q_{\theta_T^E}(z_t^T|x_{0:t})}\left[\log \frac{p_{\theta_T^D}(z_t^T) p_{\theta_T^D}(x_t^T|z_t^T)}{q_{\theta_T^E}(z_t^T|x_{0:t})} \right] \Bigg) \\
&+ E_{q_{\theta_N^E}(Z_N|\mathcal{X})}\left[\log \frac{p_{\theta_N^D}(Z_N) p_{\theta_N^D}(X_N|Z_N)}{q_{\theta_N^E}(Z_N|\mathcal{X})} \right] \\
&= \frac{1}{L}\sum_{t=1}^{L}\Bigg(E_{q_{\theta_S^E}(z_t^S|x_{0:t}) q_{\theta_T^E}(z_t^T|x_{0:t}) q_{\theta_N^E}(z_t^N|x_t)}
\end{aligned}
$$

$$\Big[\log p_{\theta^D}(x_t|z_t^S, z_t^T, z_t^N)\Big] + E_{q_{\theta_S^E}(z_t^S|x_{0:t})}\Big[\log p_{\theta_S^D}(x_t^S|z_t^S)\Big] + E_{q_{\theta_T^E}(z_t^T|x_{0:t})}\Big[\log p_{\theta_T^D}(x_t^T|z_t^T)\Big]\Big) + E_{q_{\theta_N^E}(Z_N|\mathcal{X})}\Big[\log p_{\theta_N^D}(X_N|Z_N)\Big] - KL\Big(q_{\theta_S^E}(Z_S|\mathcal{X})||p(Z_S)\Big) - KL\Big(q_{\theta_T^E}(Z_T|\mathcal{X})||p(Z_T)\Big) - KL\Big(q_{\theta_N^E}(Z_N|\mathcal{X})||p(Z_N)\Big). \quad (17)$$

REFERENCES

[1] Y. Yang, Y. Li, and H. Zhang, "Pipeline safety early warning method for distributed signal using bilinear CNN and LightGBM," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2021, pp. 4110–4114.

[2] Q. Wen, L. Yang, T. Zhou, and L. Sun, "Robust time series analysis and applications: An industrial perspective," in *Proc. 28th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2022, pp. 4836–4837.

[3] K. Leng and C.-F. Wu, "Traffic management optimization via IoT-enhanced cooperative vehicle-infrastructure systems," *IEEE Internet Things J.*, vol. 12, no. 4, pp. 3493–3501, Feb. 2025.

[4] A. Salam et al., "Securing smart manufacturing by integrating anomaly detection with zero-knowledge proofs," *IEEE Access*, vol. 12, pp. 36346–36360, 2024.

[5] F. Ullah et al., "Deep trust: A novel framework for dynamic trust and reputation management in the Internet of Things (IoT)-based networks," *IEEE Access*, vol. 12, pp. 87407–87419, 2024.

[6] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[7] Y. Liu, Y. Zhou, K. Yang, and X. Wang, "Unsupervised deep learning for IoT time series," *IEEE Internet Things J.*, vol. 10, no. 16, pp. 14285–14306, Aug. 2023.

[8] Y. Yang, C. Zhang, T. Zhou, Q. Wen, and L. Sun, "DCdetector: Dual attention contrastive representation learning for time series anomaly detection," in *Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2023, pp. 3033–3045.

[9] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 22419–22430, Dec. 2021.

[10] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *Proc. 39th Int. Conf. Mach. Learn.*, 2022, pp. 27268–27286.

[11] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "CoST: Contrastive learning of disentangled seasonal-trend representations for time series forecasting," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–18.

[12] Z. Wang, X. Xu, W. Zhang, G. Trajcevski, T. Zhong, and F. Zhou, "Learning latent seasonal-trend representations for time series forecasting," in *Proc. 36th Adv. Neural Inf. Process. Syst.*, 2022, pp. 38775–38787.

[13] 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. Knowl. Discov. Data Min.*, 2018, pp. 387–395.

[14] Q. Wen, J. Gao, X. Song, L. Sun, H. Xu, and S. Zhu, "RobustSTL: A robust seasonal-trend decomposition algorithm for long time series," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5409–5416.

[15] Y. Zhao, X. Wang, H. Yang, L. Song, and J. Tang, "Large scale evolving graphs with burst detection," in *Proc. IJCAI*, 2019, pp. 4412–4418.

[16] C. Zhang, T. Zhou, Q. Wen, and L. Sun, "TFAD: A decomposition time series anomaly detection architecture with time-frequency analysis," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manag.*, 2022, pp. 2497–2507.

[17] Z. Li et al., "Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2021, pp. 3220–3230.

[18] 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. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2020, pp. 4322–4326.

[19] F. Tonolini, B. S. Jensen, and R. Murray-Smith, "Variational sparse coding," in *Proc. 35th Uncertain. Artif. Intell.*, 2020, pp. 690–700.

[20] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Stat. Assoc.*, vol. 65, pp. 1509–1526, Dec. 1970.

[21] J. Knoblauch and T. Damoulas, "Spatio-temporal Bayesian on-line changepoint detection with model selection," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 2718–2727.

[22] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "COPOD: Copula-based outlier detection," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2020, pp. 1118–1123.

[23] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "ECOD: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12181–12193, Dec. 2023.

[24] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[25] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proc. IEEE Found. New Dir. Data Min. Workshop*, 2003, pp. 172–179.

[26] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2000, pp. 427–438.

[27] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1641–1650, 2003.

[28] M. Goldstein and A. Dengel, "Histogram-based outlier score (HBOS): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster Demo Track*, vol. 1, pp. 59–63, Sep. 2012.

[29] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Min.*, 2008, pp. 413–422.

[30] T. Pevnỳ, "Loda: Lightweight on-line detector of anomalies," *Mach. Learn.*, vol. 102, pp. 275–304, Feb. 2016.

[31] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022, *arXiv:1312.6114*.

[32] T. Kieu, B. Yang, and C. S. Jensen, "Outlier detection for multidimensional time series using deep neural networks," in *Proc. 19th IEEE Int. Conf. Mobile Data Manage. (MDM)*, 2018, pp. 125–134.

[33] H. Zhao et al., "Multivariate time-series anomaly detection via graph attention network," in *Proc. IEEE Int. Conf. Data Min. (ICDM)*, 2020, pp. 841–850.

[34] 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. Knowl. Discov. Data Min.*, 2019, pp. 2828–2837.

[35] 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, Feb. 2023.

[36] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4027–4035.

[37] H. Qin, X. Zhan, and Y. Zheng, "CSCAD: Correlation structure-based collective anomaly detection in complex system," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 5, pp. 4634–4645, May 2023.

[38] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "BeatGAN: Anomalous rhythm detection using adversarially generated time series," in *Proc. IJCAI*, 2019, pp. 4433–4439.

[39] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," in *Proc. Int. Conf. Learn. Represent.*, 2022, pp. 1–20.

[40] C. Wang et al., "Drift doesn't matter: Dynamic decomposition with diffusion reconstruction for unstable multivariate time series anomaly detection," in *Proc. 37th Conf. Neural Inf. Process. Syst.*, 2023, pp. 10758–10774.

[41] H. Zhang, Y.-F. Zhang, W. Liu, A. Weller, B. Schölkopf, and E. P. Xing, "Towards principled disentanglement for domain generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8024–8034.

[42] C.-Y. Lai, F.-K. Sun, Z. Gao, J. H. Lang, and D. S. Boning, "Nominality score conditioned time series anomaly detection by point/sequential reconstruction," in *Proc. 37th Adv. Neural Inf. Process. Syst.*, 2023, pp. 1–19.

[43] A. Siffer, P.-A. Fouque, A. Termier, and C. Largouet, "Anomaly detection in streams with extreme value theory," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, 2017, pp. 1067–1075.

[44] D. Gottlieb and C.-W. Shu, "On the Gibbs phenomenon and its resolution," *SIAM Rev.*, vol. 39, no. 4, pp. 644–668, 1997.

[45] R. G. Krishnan, U. Shalit, and D. Sontag, "Structured inference networks for nonlinear state space models," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 2101–2109.
[46] A. Abdulaal, Z. Liu, and T. Lancewicki, "Practical approach to asynchronous multivariate time series anomaly detection and localization," in *Proc. 27th ACM SIGKDD Conf. Knowl. Discov. Data Min.*, 2021, pp. 2485–2494.
[47] A. P. Mathur and N. O. Tippenhauer, "SWaT: A water treatment testbed for research and training on ICS security," in *Proc. Int. Workshop Cyber-Phys. Syst. Smart Water Netw. (CySWater)*, 2016, pp. 31–36.

Liang Zhang received the B.S. degree in software engineering and the M.S. degree in computer applied technology from Dalian University of Technology, Dalian, China, in 2011 and 2014, respectively, and the Ph.D. degree in applied mathematics, Stony Brook University, Stony Brook, NY, USA, in 2018.

He is an Associate Professor with the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. He has published papers in high-quality journals and conferences, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *Journal of Biomedical Informatics*, *Journal of Medical Internet Research*, IEEE INTERNATIONAL CONFERENCE ON DATA MINING, CIKM, and BIBM. His research interests include sequential data mining, graph structure learning, and time series prediction.

Jianping Zhu received the B.S. degree from Northeast Agricultural University, Harbin, China, in 2021, and the M.S. degree from Dalian University of Technology, Dalian, China, in 2024.

He is a Research Assistant with Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China. He has published papers in high-quality conferences, such as AAAI, BIBM, and PAKDD. His research interests include data mining, information systems, and intelligent computing.

Guangjie Han (Fellow, IEEE) received the Ph.D. degree from Northeastern University, Shenyang, China, in 2004.

He is currently a Professor with the Department of Internet of Things Engineering, Hohai University, Changzhou, China. In February 2008, he was a Postdoctoral Researcher with the Department of Computer Science, Chonnam National University, Gwangju, South Korea. From October 2010 to October 2011, he was a Visiting Research Scholar with Osaka University, Suita, Japan. From January 2017 to February 2017, he was a Visiting Professor with the City University of Hong Kong, Hong Kong, China. From July 2017 to July 2020, he was a Distinguished Professor with Dalian University of Technology, Dalian, China. He has over 500 peer-reviewed journal and conference papers, in addition to 160 granted and pending patents. His current H-index is 73 and i10-index is 322 in Google Citation (Google Scholar). The total citation count of his papers raises above 18 700+ times. His current research interests include Internet of Things, industrial Internet, machine learning, artificial intelligence, mobile computing, and security and privacy.

Prof. Han has been awarded the 2020 IEEE Systems Journal Annual Best Paper Award and the 2017–2019 IEEE Access Outstanding Associate Editor Award. He has served on the Editorial Boards of up to 10 international journals, including the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, and IEEE SYSTEMS. He has Guest-Edited several special issues in IEEE Journals and Magazines, including the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE COMMUNICATIONS, IEEE WIRELESS COMMUNICATIONS, and *Computer Networks*. He has also served as the Chair of organizing and technical committees in many international conferences. He is a Fellow of the U.K. Institution of Engineering and Technology..

Bo Jin (Senior Member, IEEE) received the B.S. degree in mechanical design automation and the Ph.D. degree in computer application technology from Dalian University of Technology, Dalian, China, in 2001 and 2009, respectively.

He is currently a Full Professor with the School of Innovation and Entrepreneurship, Dalian University of Technology. He has published papers in high-quality journals and conferences, such as IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE SMC, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, *Knowledge-Based Systems*, SIGKDD, AAAI, ICDM, CIKM, and SDM. His research interests include information retrieval, data mining, and intelligent computing.

Pengfei Wang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in software engineering from Northeastern University, Shenyang, China, in 2013, 2015, and 2020, respectively.

From 2016 to 2018, he was a visiting Ph.D. student with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA. He is currently an Associate Professor with the School of Computer Science and Technology, Dalian University of Technology, Dalian, China. He has authored more than 60 papers on high-quality journals and conferences, such as IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON SERVICES COMPUTING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, ACM TRANSACTIONS ON SENSOR NETWORKS, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, IEEE INFOCOM, IEEE ICNP, and IEEE ICDCS. He also holds a series of patents in U.S. and China. His research interests are distributed artificial intelligence, computer networks, and IoT.

Xiaopeng Wei received the Ph.D. degree from Dalian University of Technology, Dalian, China, in 1993.

He is currently a Full Professor with Dalian University of Technology, and Dalian University, Dalian. He has co-authored about 160 published papers. His research interests include medical and health informatics, computer animation, computer vision, and intelligent CAD.