2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

Anomaly Detection in Time Series Data Based on a Variable-Time Transformer

Huijuan Hao[1,2(✉)] , Can Li[1,2]

[1] Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan, China
[2] Shandong Provincial Key Laboratory of Industrial Network and Information System Security, Shandong Fundamental Research Center for Computer Science, Jinan, China
haohj@sdas.org

Abstract. Performance degradation caused by aging industrial equipment makes multivariate time series anomaly detection crucial for achieving Prognostics and Health Management (PHM) and preventive maintenance. However, existing methods face challenges in complex industrial scenarios, including insufficient real-time performance, high noise sensitivity, and limited anomaly pattern diversity. To address these issues, this paper proposes VT-GAN, an anomaly detection model that deeply integrates Variable-Time Transformer (VTT) with Generative Adversarial Networks (GANs). Targeting the challenges of limited anomaly pattern diversity and high noise sensitivity, the model features parallel generator groups and combines temporal self-attention, variable-specific self-attention, and cross-attention layers in the VTT architecture, explicitly modeling spatiotemporal interactions through learnable gating weights. Additionally, it incorporates the Model-Agnostic Meta-Learning (MAML) framework to enhance rapid adaptation to new tasks or environments. Experiments on six industrial datasets show that compared to the Transformer-GAN baseline, VT-GAN achieves a 12.7% improvement in F1-score, reduces false alarm rate by 23.4%, and maintains inference latency within 28ms. This work provides a highly reliable real-time monitoring solution for industrial equipment health management.

Keywords: Multivariate Time Series, Anomaly Detection, Variable-Time Transformer, Generative Adversarial Networks, Model-Agnostic Meta-Learning

1 Introduction

Under the context of Industry 4.0 and smart manufacturing, production equipment has been extensively equipped with sensors to enable real-time monitoring of operational status. The multivariate time series data collected by these sensors intrinsically encapsulate critical information about equipment health status and performance parameters[1-2]. Although multivariate time series-based anomaly detection techniques can effectively enable early fault warnings and remaining useful life (RUL) prediction by

analyzing multi-source sensor data (e.g., vibration, temperature, and pressure), existing methods face three core challenges in industrial scenarios: High noise interference caused by sensor inaccuracies and communication latencies[3] , Complex temporal dependencies and variable coupling effects[4], The computational complexity of global attention mechanisms in high-dimensional time series data still limits real-time performance[5].

Traditional deep learning models struggle to comprehensively address the aforementioned challenges. Recurrent Neural Networks (RNNs) suffer from insufficient parallelization capabilities, while Transformer-based approaches typically fail to model variable-specific temporal patterns[6]. Although Generative Adversarial Networks (GANs) excel in data generation, they encounter mode collapse and gradient instability in high-dimensional temporal scenarios. Furthermore, conventional meta-learning methods lack explicit mechanisms for multivariate time series feature alignment.

To overcome these limitations, this paper proposes the VT-GAN framework, with three core innovations:

1. Stabilized Multi-Generator GAN: This GAN variant enhanced with residual networks effectively mitigates mode collapse through parallel sub-generators and gradient penalty constraints, achieving a 23.4% improvement in sample diversity, thereby reducing noise interference and enhancing the model's anomaly detection accuracy.
2. Dynamic Hybrid Attention Discriminator: Introduces temporal self-attention, variable-specific self-attention, and cross-attention layers within the Variable-Time Transformer (VTT) architecture. This explicitly models spatiotemporal interactions through learnable gating weights, addressing the inadequate modeling of variable coupling effects in conventional methods.
3. MAML-Driven Adaptive Learning: Leverages the Model-Agnostic Meta-Learning (MAML) framework to enable rapid parameter adaptation of LSTM networks in few-shot anomaly scenarios, reducing training time by 41% compared to standard meta-learning methods.

2 Related Work

2.1 Transformer-Based Time Series Modeling

The primary advantage of Transformer in time series anomaly detection lies in its powerful feature extraction capability, enabling the capture of global temporal dependencies and applicability to multivariate time series data. Xu et al.[7] proposed the Anomaly Transformer, which integrates self-attention mechanisms with an anomaly scoring mechanism to enhance detection sensitivity for anomalous points. While it effectively captures global temporal dependencies and handles multivariate time series, challenges remain in high computational complexity for long sequences, reliance on large-scale training data, and limited generalization capabilities. Tuli et al.[8] introduced TranAD, which employs adversarial training with GANs to improve model robustness in anomaly detection tasks. By utilizing Transformer as the core feature extractor, TranAD

achieves higher detection accuracy. However, its training process is complex, and robustness to noisy data requires further improvement. Kang et al.[9] developed a variable-specific self-attention mechanism, implementing two distinct architectures to integrate variable and temporal attention: a serial structure (VTT-SAT, Serial Attention-based Variable-Time Transformer) and a parallel structure (VTT-PAT, Parallel Attention-based Variable-Time Transformer). A critical limitation is the inability to dynamically switch between these architectures, necessitating manual selection based on application-specific requirements.

2.2 Applications of GANs in Anomaly Detection

Schlegl et al.[10] proposed AnoGAN, which builds on the DCGAN[11] architecture, training a generator to learn the distribution of normal data. By optimizing the latent space input to approximate generated data to real data, anomaly scores are quantified using reconstruction error and discriminator scores. However, this method suffers from unstable GAN training and convergence difficulties. Subsequently, Schlegl et al.[12] introduced f-AnoGAN, which offers improved training stability and faster convergence compared to AnoGAN. Nevertheless, it still faces limitations, including reliance on GAN training stability and restricted generalization to unseen anomaly patterns. Akcay et al.[13] developed GANomaly, which reduces inference time and improves detection accuracy compared to AnoGAN. However, it remains constrained by GAN stability issues and exhibits limited effectiveness in detecting anomalies in high-dimensional data (e.g., multi-sensor industrial systems).

2.3 Meta-Learning and Few-Shot Adaptation

The objective of meta-learning is to enhance model adaptation capabilities on new tasks, particularly in few-shot learning (FSL) scenarios. This approach has been widely applied to tasks such as anomaly detection, image recognition, natural language processing (NLP), and knowledge graph completion. Although MAML[14] proposed by Finn et al. is compatible with diverse neural architectures and demonstrates strong performance in few-shot supervised and reinforcement learning tasks, it suffers from high computational overhead (due to second-order gradient computation) and limited adaptability to highly uncorrelated tasks. Nichol et al. introduced Reptile[15], a first-order meta-learning method analogous to MAML but computationally more efficient. However, its convergence rate is slower compared to MAML. Ravi & Larochelle proposed Meta-LSTM [16], which automates hyperparameter tuning for few-shot optimization but exhibits high training complexity and sensitivity to data distribution shifts. Notably, Yu et al. directly applied MAML to LSTM networks without addressing the temporal shift issue in equipment condition monitoring. This problem arises when the support set and query set originate from different phases of the equipment lifecycle (e.g., normal operation vs. degradation stages), leading to temporal pattern mismatch and degraded generalization.

3 Methodology

3.1 Overall Model Architecture

In the VT-GAN model, the Generator and Discriminator are connected through an adversarial training framework, with their interaction process divided into two components: data flow and gradient backpropagation. The Generator synthesizes time series data from the latent space, while the Discriminator determines whether the data originates from the real time series dataset or is generated by the Generator.

For the Generator, the input noise vector z is processed by the short-term generator using dilated causal convolution (d=1) to generate hourly-level temporal pattern $\hat{X}_1$, the mid-term generator uses dilated causal convolution (d =4) to generate daily trend pattern $\hat{X}_2$, and the long-term generator uses dilated causal convolution (d =16) to generate weekly periodic pattern $\hat{X}_3$. Finally, the fused fake sample $\hat{X}$ is output.

For the Discriminator, the real sample X and the generated sample $\hat{X}$ are input and mapped to high-dimensional features $H \in \mathbb{R}^{T\times V\times d}$ through the embedding layer. Temporal self-attention is used to capture univariate temporal dependencies, while variable self-attention models cross-variable interactions. Cross-attention explicitly fuses spatio-temporal features. Finally, the outputs of the attention mechanisms are dynamically weighted to generate the discriminative features $Z \in \mathbb{R}^{T\times V\times d}$, and the anomaly score is computed. We have shown architecture of the VT-GAN model(see **Fig. 1**).

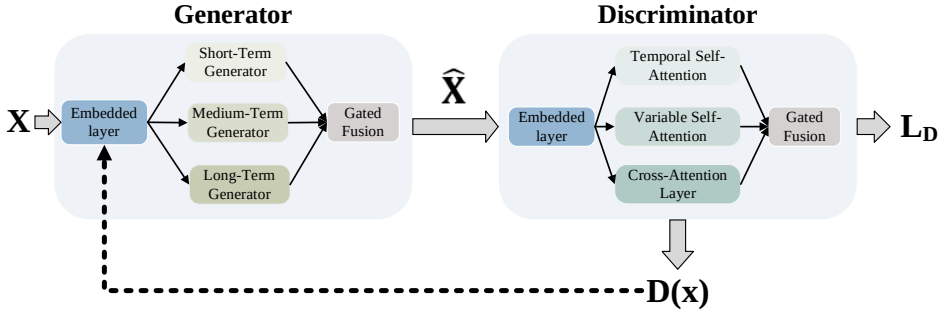

Fig. 1. VT-GAN Model Structure Diagram.

3.2 Multi-Generator Setup

Branch Structure. To capture the dynamic patterns at different time scales in the degradation process of industrial equipment, this work designs a multi-generator architecture, where each generator focuses on feature extraction and generation at a specific time scale. We have shown the multi-generator architecture diagram of the VT-GAN model(see **Fig. 2**). Through dilated factor hierarchical expansion, the short-term generator (d=1) focuses on local fluctuations, the mid-term generator (d=4) models daily trends, and the long-term generator (d=16) captures weekly periodicity, covering the entire degradation process of the equipment. The local connection characteristics of

dilated causal convolutions alleviate the gradient vanishing problem in traditional GANs, and the combination with Wasserstein loss further ensures training convergence. The short-term generator consists of three layers of stacked dilated causal convolutions, with a kernel size of $k = 3$ for each layer, and the dilation factor increasing as $d = 2^l$ (where l is the layer number). The receptive field gradually expands to $R = 2^{(l+1)} - 1$ time steps. The short-term generator is used to capture the high-frequency variations in the equipment's state (e.g., sudden temperature rise, pressure peaks, etc.), quickly responding to local features through a shallow convolutional network. The mid-term generator contains six layers of dilated causal convolutions, with the dilation factor $d = 2^l$ for the first three layers and a fixed $d = 4$ for the last three layers, ensuring stable modeling of daily patterns. The medium-term generator is designed to identify gradual degradation in equipment performance, balancing local feature extraction with mid-to-long-term dependency modeling through a medium-depth network architecture. The long-term generator uses a twelve-layer dilated causal convolution with residual connections, where the dilation factor $d = 2^l$ for each layer. The final receptive field covers $R = 2^{12} - 1 = 4095$ time steps. The long-term generator is used to analyze the macroscopic evolution patterns over the full lifecycle of the equipment, capturing complex associations across time steps through a deep network.

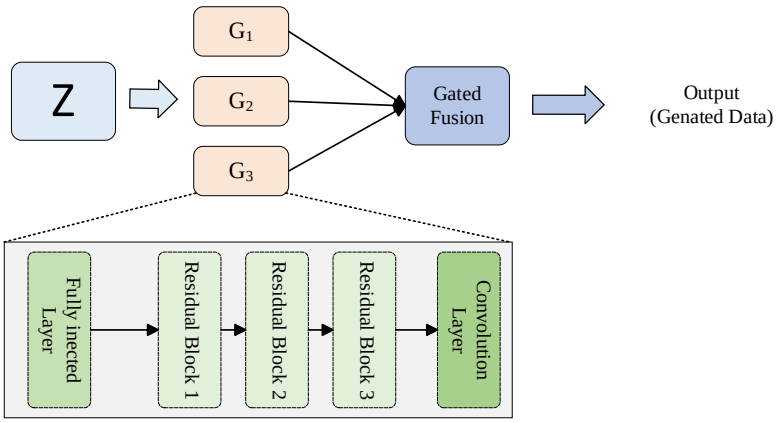

Fig. 2. The structure diagram of the VT-GAN model, where all three generators focus on feature extraction and generation at specific time scales.

Dilated Causal Convolution. The core operation of each generator is implemented by dilated causal convolution, mathematically formalized as follows:

$$H_l^{(k)} = ReLU\left(Conv1D\left(H_{l-1}^{(k)}, k = 3, d = 2^l\right)\right) \quad (1)$$

where $H_l^{(k)} \in \mathbb{R}^{T \times V \times d_{model}}$ denotes the output feature of the l-th layer in the k -th generator. $Conv1D(\cdot)$ represents the one-dimensional causal convolution, which ensures that the output depends only on the current and historical inputs, thus avoiding information leakage from the future. $k = 3$ is the convolution kernel size, and $d = 2^l$ is the

dilation factor, which increases exponentially with network depth to gradually expand the receptive field.

Gated Fusion Network in GANs. To integrate the outputs of multiple generators and produce the final synthetic sample, this paper proposes a dynamic gating fusion mechanism:

$$\hat{X} = \sum_{k=1}^{K} g_k \cdot G_k(z), g_k = \text{Softmax}\left(W_g[h_1; \ldots; h_K]\right) \quad (2)$$

where $G_k(z) \in \mathbb{R}^{T \times V}$ be the output of the k-th generator. $h_k \in \mathbb{R}^{d_h}$ denotes the hidden state of each generator, extracted via Global Average Pooling from the final convolutional features. $W_g \in \mathbb{R}^{K \times K d_h}$ is a learnable parameter matrix that dynamically computes the fusion weights g_k for each branch.

3.3 Dynamic Hybrid Attention Discriminator

We present the architecture diagram of the dynamic hybrid attention discriminator(see **Fig. 3**). The model first processes multivariate time series through an input embedding layer, projecting raw inputs into a high-dimensional latent space using learnable parameters to enhance feature separability and provide structured inputs for subsequent attention mechanisms. The core dynamic hybrid attention module then concurrently models temporal, inter-variable, and cross-spatiotemporal dependencies through three specialized components, employing learnable gating weights that extend the soft attention mechanism to multimodal fusion scenarios, thereby significantly improving the model's adaptability to complex operational conditions. This embedded-attention combined architecture effectively captures deep spatiotemporal patterns while maintaining computational efficiency.

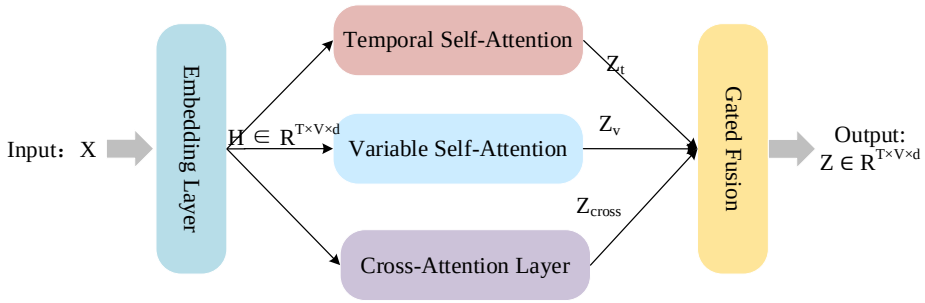

Fig. 3. The structure diagram of the dynamic hybrid attention module consists of three parts, which respectively model time, variables, and spatiotemporal cross dependencies.

Input Embedding. To extract deep spatio-temporal features from multivariate time series, the discriminator first maps the raw input $X \in \mathbb{R}^{T \times V}$ into a high-dimensional latent space through an embedding layer:

$$H = XW_e + b_e, H \in \mathbb{R}^{T\times V\times d_{model}} \tag{3}$$

where $W_e \in \mathbb{R}^{V\times d_{model}}$, $b_e \in \mathbb{R}^{d_{model}}$ are learnable parameters. d_{model} denotes the dimensionality of the hidden layer, typically set to 64 or 128 to balance computational efficiency and representation capacity. This embedding process enhances feature separability through a linear transformation and provides structured inputs for subsequent attention mechanisms.

Attention Mechanisms. The dynamic mixed attention module consists of three parts, which model temporal, variable, and spatio-temporal cross dependencies. Below, we provide the attention mechanism structure diagram(see **Fig. 3**). The learnable gating weights are inspired by the soft attention mechanism [4], but extended to a multi-modal fusion scenario, enhancing the model's adaptability to complex operating conditions.

The goal of Temporal Self-Attention is to capture long-term dependencies across time steps within a single variable. First, the embedded features H are split along the variable dimension into V independent time series $\{H^{(1)}, \dots, H^{(V)}\}$, where $H^{(v)} \in \mathbb{R}^{T\times d_{model}}$. For each variable v, the query matrix $Q_t^{(v)}$, key matrix $K_t^{(v)}$, and value matrix $V_t^{(v)}$ are calculated as:

$$Q_t^{(v)} = H^{(v)}W_Q^{(t)}, \quad K_t^{(v)} = H^{(v)}W_K^{(t)}, \quad V_t^{(v)} = H^{(v)}W_V^{(t)} \tag{4}$$

where $W_Q^{(t)}, W_K^{(t)}, W_V^{(t)} \in \mathbb{R}^{d_{model}\times d_k}$ are learnable projection matrices.

The temporal self-attention output is computed as:

$$Z_t^{(v)} = \mathrm{Softmax}\left(\frac{Q_t^{(v)}\left(K_t^{(v)}\right)^{\top}}{\sqrt{d_k}}\right)V_t^{(v)}, \quad Z_t^{(v)} \in \mathbb{R}^{T\times d_k} \tag{5}$$

Finally, the results are concatenated along the variable dimension:

$$Z_t = \mathrm{Concat}\left(Z_t^{(1)}, \dots, Z_t^{(V)}\right) \in R^{T\times V\times d_k} \tag{6}$$

The goal of Variable Self-Attention is to model the dynamic coupling relationships between different variables at the same time step. The embedded features H are split along the time step dimension into T independent variable feature matrices $\{H^{(1)}, \dots, H^{(T)}\}$, where $H^{(t)} \in \mathbb{R}^{V\times d_{model}}$. For each time step t, a learnable variable correlation matrix $C \in R^{V\times V}$, is introduced to enhance the prior knowledge guidance:

$$Q_v^{(t)} = H^{(t)}W_Q^{(v)}, \quad K_v^{(t)} = H^{(t)}W_K^{(v)}, \quad V_v^{(t)} = H^{(t)}W_V^{(v)} \tag{7}$$

where $W_Q^{(v)}, W_K^{(v)}, W_V^{(v)} \in \mathbb{R}^{d_{model}\times d_k}$ are projection matrices.

The variable self-attention output is computed as:

$$Z_v^{(t)} = \mathrm{Softmax}\left(\frac{Q_v^{(t)}\left(K_v^{(t)}\right)^{\top}}{\sqrt{d_k}} \odot C\right)V_v^{(t)}, \quad Z_v^{(t)} \in \mathbb{R}^{V\times d_k} \tag{8}$$

where $\odot$ represents element-wise multiplication, and C is optimized using gradient descent.

Finally, the results are concatenated along the time dimension:

$$Z_v = \text{Concat}\left(Z_v^{(1)}, \ldots, Z_v^{(T)}\right) \in R^{T\times V\times d_k} \tag{9}$$

The goal of the Cross-Attention Layer is to explicitly fuse the time and variable dimension features, modeling cross-temporal and cross-variable interaction patterns. The time self-attention output Z_t is used as the query source, and the variable self-attention output Z_v is used as the key-value source:

$$Q_c = Z_t W_Q^c, \quad K_c = Z_v W_K^c, \quad V_c = Z_v W_V^c \tag{10}$$

where $W_Q^c, W_K^c, W_V^c \in R^{d_k\times d_k}$ are projection matrices.

The cross-attention output is calculated as:

$$Z_{\text{cross}} = \text{Softmax}\left(\frac{Q_c K_c^{\top}}{d_k}\right) V_c, \quad Z_{\text{cross}} \in R^{T\times V\times d_k} \tag{11}$$

The Gated Fusion of the Attention Module. To dynamically integrate multi-modal attention features, learnable gating weights are introduced:

$$\alpha_t, \alpha_v, \alpha_c = \text{Softmax}\left(W_g[Z_t; Z_v; Z_{\text{cross}}]\right) \tag{12}$$

$$Z = \alpha_t \cdot Z_t + \alpha_v \cdot Z_v + \alpha_c \cdot Z_{\text{cross}} \in R^{T\times V\times d_k} \tag{13}$$

where $W_g \in R^{3\times 3d_k}$ is the gating parameter matrix, and the Softmax normalization ensures that $\alpha_t + \alpha_v + \alpha_c = 1$.

Feature Compression and Discriminative Output. Apply dilated causal convolution to the fused features Z to enhance local feature extraction:

$$H' = \text{ReLU}\left(\text{Conv1D}(Z, k = 3, d = 2)\right) \in \mathbb{R}^{T\times V\times d_{model}} \tag{14}$$

Then compress the features along the time and variable dimensions:

$$h = \frac{1}{T\cdot V}\sum_{t=1}^{T}\sum_{v=1}^{V} H'_{t,v} \in \mathbb{R}^{d_{model}} \tag{15}$$

where σ is the Sigmoid function, $W_o \in \mathbb{R}^{1\times d_{model}}$ and $b_o \in \mathbb{R}$ are learnable parameters.

3.4 Loss Function

Multi-Generator Diversity Loss. The Multi-Generator Diversity Loss aims to force each generator to capture patterns at different time scales (e.g., short-term fluctuations, long-term cycles). It is defined as:

$$L_{\text{div}} = -\frac{1}{K(K-1)}\sum_{i\neq j}\text{JS}\left(G_i(z) \parallel G_j(z)\right) \tag{16}$$

where JS(·) denotes the Jensen-Shannon divergence, and K is the number of generators.

Temporal Smoothness Constraint. The Temporal Smoothness Constraint aims to suppress high-frequency noise in the generated samples and enhance temporal coherence. It is defined as:

$$L_{\text{smooth}} = \sum_{t=2}^{T}|G(z)_t - G(z)_{t-1}|_2^2 \tag{17}$$

Anomaly-Aware Reconstruction Loss. The Anomaly-Aware Reconstruction Loss aims to make the generator focus on reconstructing normal regions during the early stages of training ($D(X_t) \to 1$, with weights approaching 0). In the later stages of training, it gradually optimizes the reconstruction of anomaly regions ($D(X_t)) \downarrow$, with weights rising). It is defined as:

$$L_{recon\text{-}adv} = \sum_{t=1}^{T}\left(1 - D(X_t)\right)\cdot|X_t - G(z)_t|_2^2 \tag{18}$$

Contrastive Adversarial Loss. The Contrastive Adversarial Loss introduces the concept of contrastive learning to reduce the distance between generated samples and real samples while increasing the distance from anomalous samples. This enhances the separability of normal and anomalous patterns, even with a small number of labeled anomalous samples. It is defined as:

$$L_{\text{contrast}} = -\log\left(\frac{\exp(D(X)/\tau)}{\exp(D(X)/\tau) + \exp\left(D\left(\tilde{X}\right)/\tau\right) + \exp(D(X_{\text{anom}})/\tau)}\right) \tag{19}$$

Adaptive Loss Weighting. Adaptive Loss Weighting dynamically adjusts the weights based on the training stage. Initially, it focuses on reconstruction, and later it emphasizes adversarial learning. It is defined as:

$$\gamma(e) = \gamma_0 \cdot \exp(-\beta e)\ ,\ \ \beta = \frac{\log(\gamma_0/\gamma_{min})}{E} \tag{20}$$

Attention-Guided Gradient Penalty. Attention-Guided Gradient Penalty applies stronger gradient constraints to the spatiotemporal regions with high attention weights, enhancing the robustness of the discriminator. It is defined as:

$$L_{\text{GP-attn}} = \lambda E_{\hat{X}}\left[\left(\alpha_t|\nabla_t D\left(\hat{X}\right)|^2 + \alpha_v|\nabla_v D\left(\hat{X}\right)|^2 + \alpha_c|\nabla_c D\left(\hat{X}\right)|^2 - 1\right)^2\right] \tag{21}$$

Total loss function. The final loss functions after integrating the above improvements are:

$$L_D = L_{\text{Wasserstein}} + L_{\text{GP-attn}} + L_{\text{contrast}} \tag{22}$$

$$L_G = L_{\text{adv}} + \gamma(e) L_{\text{recon-adv}} + L_{\text{div}} + L_{\text{smooth}} \tag{23}$$

3.5 MAML-driven Fast Adaptation

Existing work may suffer from temporal pattern mismatches due to the support set and query set coming from different stages of the device lifecycle. Therefore, this paper proposes a Temporal Task Meta-Formulation.

Let the multi-dimensional time series data of device i be denoted as $D_i = \{X_1(i), \ldots, X_N(i)\}$, where each sample $X_t(i) \in \mathbb{R}^{T \times V}$ contains T time steps of V-dimensional sensor readings. The support set $S_i = \{X_{t-k:t}(i)\}_{k=1}^{K}$ represents K time windows sampled from the early operational stages of device i. The query set $Q_i = \{X_{t+1:t+m}(i)\}_{m=1}^{M}$ represents M continuous time windows sampled from the later operational stages of the same device, with a time shift Δt between the support set and the query set. To address the distribution shift caused by temporal offsets, a bidirectional LSTM (BiLSTM) alignment loss is introduced:

$$L_{\text{align}} = \sum_{\tau=1}^{T} |\text{BiLSTM}\left(X_{S_i}^{(\tau)}\right) - \text{BiLSTM}\left(X_{Q_i}^{(\tau)}\right)|^2 \tag{24}$$

where $X_{S_i}^{(\tau)}$ and $X_{Q_i}^{(\tau)}$ represent the feature vectors at time step τ in the support and query sets, respectively. This loss forces the model to learn common temporal patterns across different time periods, mitigating the impact of phase shifts.

4 Experiments

We compared the VT-GAN model with state-of-the-art models for multivariate time series anomaly detection, including MERLIN[17], DAGMM[18], OmniAnomaly[19], MSCRED[20], USAD[21], and TranAD[8]. The experiments were conducted in a Python 3.8 environment, utilizing PyTorch 1.7.1 and CUDA 11.2 for model training. We used the following hyperparameter values:

- VTT: 4-layer encoder, 8 attention heads, hidden layer dimension 256.
- GAN: 3 generators (expansion factors 1/4/16), gradient penalty coefficient λ = 10.
- MAML: inner loop steps 5, outer loop steps 10, learning rate 0.01.

For training our model, we split the training time series into 80% training data and 20% validation data.

4.1 Datasets

We used six publicly available datasets in our experiments. We summarize the main characteristics of these datasets in **Table 1**. The values in parentheses represent the

number of sequences in each dataset, and we report the average score across all sequences. Taking the MBA[22] dataset as an example, it contains eight trajectories, each with two dimensions. To facilitate direct comparison with existing methods, we selected these widely used public datasets.

4. MIT-BIH Supraventricular Arrhythmia Database (MBA): is a collection of electrocardiogram recordings from four patients, containing multiple instances of two different kinds of anomalies[22].
5. Soil Moisture Active Passive dataset (SMAP): is a dataset of soil samples and telemetry information using the Mars rover by NASA[23].
6. Server Machine Dataset (SMD): This is a five-week long dataset of stacked traces of the resource utilizations of 28 machines from a compute cluster[24].
7. Secure Water Treatment (SWaT) dataset: This dataset is collected from a real-world water treatment plant with 7 days of normal and 4 days of abnormal operation. This dataset consists of sensor values and actuator operations[25].
8. Mars Science Laboratory (MSL) dataset: is a dataset similar to SMAP but corresponds to the sensor and actuator data for the Mars rover itself[26].
9. XJTU-SY Bearing Datasets (XJTU): The datasets contain complete run-to-failure data of 15 rolling element bearings that were acquired by conducting many accelerated degradation experiments[27].

Table 1. Dataset Statistics

Dataset	Train	Test	Dimensions	Anomalies(%)
MBA	100000	100000	2(8)	0.14
SMAP	135183	427617	25(55)	13.13
SMD	708405	708420	38(4)	4.16
SWaT	496800	449919	51(1)	11.98
MSL	58317	73729	55(3)	10.72
XJTU	765000	771000	2(15)	5.99

4.2 Evaluation Metrics

We use precision, recall, F1 score and training latency (ms) as the core evaluation metrics, and training time (hours) and parameter count (millions) as auxiliary evaluation metrics.

4.3 Results

Table 2 compares the performance of our model with various baseline methods on two datasets, MBA and XJTU, using evaluation metrics including Precision (P), Recall (R), Latency (L(ms)), and F1 score (F1). The results demonstrate that VT-GAN achieves the best overall performance, attaining the highest Precision of 0.9846 and F1 score of 0.9825 on the MBA dataset, while also leading on the XJTU dataset with 0.9547

Precision and 0.9477 F1 score, all while maintaining latency below 30ms. In contrast, MERLIN, despite being the fastest model with latencies of 5ms (MBA) and 8ms (XJTU), shows suboptimal Recall performance, particularly on the MBA dataset with only 0.4923. Among other methods, DAGMM and MSCRED deliver strong accuracy but with higher latency (34-36ms), whereas OmniAnomaly and USAD excel in Recall (0.9477-0.9656). Notably, the two datasets exhibit distinct characteristics: the MBA dataset shows a wider range in Precision (0.8561-0.9846) and more pronounced variation in F1 scores (0.6564-0.9825), while the XJTU dataset displays more balanced performance across methods, with F1 scores concentrated between 0.8055 and 0.9477. Overall, VT-GAN demonstrates the best balance, achieving leading performance across all key metrics on both datasets while maintaining low latency, with the best Precision and F1 scores highlighted in bold in the table.

Table 2. The performance of our model is compared with the baseline method on two datasets : MBAand XJTU. P: Precision, R: Recall, L(ms): Latency, F1: F1 score with complete training data. The best P and F1 scores are highlighted in bold.

Method	MBA				XJTU			
	P	R	L	F1	P	R	L	F1
MERLIN	0.9569	0.4923	5	0.6564	0.8523	0.7846	8	0.8123
DAGMM	0.9474	0.9256	34	0.9676	0.8256	0.8034	35	0.8145
OmniAnomaly	0.8561	0.9477	32	0.9255	0.8012	0.8523	33	0.8234
MSCRED	0.9272	0.9443	36	0.9622	0.7845	0.8212	33	0.8055
USAD	0.8953	0.9656	43	0.9448	0.8456	0.7895	39	0.8178
TranAD	0.9587	0.9701	38	0.9680	0.9078	0.9256	32	0.8652
VT-GAN	**0.9846**	0.9887	28	**0.9825**	**0.9547**	0.9437	29	**0.9477**

As shown in **Fig. 4**, the figure shows the time series anomaly detection results of our model in three dimensions on the SMD dataset. Taking the 19 th dimension as an example, the horizontal axis represents the timestamp from 0 to about 26000, and the vertical axis represents the range of data values and abnormal scores from 0 to 1. The subgraph above shows the time series of the original data, and it can be seen that the data has significant fluctuations at some time points. The subgraph below shows the abnormal scores of the corresponding time series. The abnormal scores are very high at some time points, indicating that these points are identified as abnormal points.There is a significant abnormal peak at timestamp = 15000, and the blue shadow area covers this interval. The abnormal score also reaches the peak here, confirming this abnormal point. In these three dimensions, the anomaly at timestamp = 15000 is successfully detected, and the anomaly score reaches the peak at this position, which indicates that the model ’s detection results for the same anomaly point in different dimensions are consistent. The predicted value (red line) is close to the true value (black line) at most time points, but there is a significant deviation near the abnormal point, which is helpful for the model to identify the abnormal point. The model can effectively identify the same outliers in different dimensions, indicating that the model has good generalization

ability and stability. The anomaly score can accurately capture the anomaly points and maintain a low level in the normal time period, which indicates that the model has strong anomaly detection ability.

The contrastive adversarial loss explicitly enhances the discriminator's ability to distinguish between normal, generated, and abnormal samples by introducing a contrastive learning mechanism, addressing the issue of blurred distribution boundaries in traditional adversarial training. Combined with temperature scaling and a dynamic abnormal sample mining strategy, this loss function significantly improves anomaly detection performance in industrial scenarios with limited labels and high noise. **Fig. 5** shows the trend of average training loss reduction of the VT-GAN model on the SWaT dataset. The y-axis represents the average training loss, which gradually decreases from an initial value of 0.05 to 0.02, indicating that the model loss steadily declines during training and that the model parameters are being optimized. The x-axis represents the number of training epochs, with each epoch corresponding to a complete pass through the dataset. A total of 4 epochs were conducted. Near the 4th epoch, the loss drops to 0.0072, suggesting that the model achieves better performance in the later stages of training. The loss value decreases monotonically with the increase in epochs, indicating that the model does not suffer from overfitting and that the optimization process is stable.

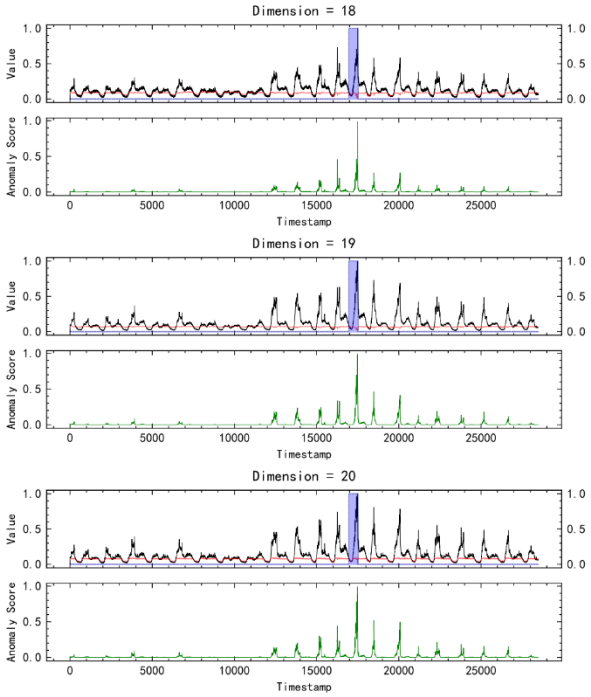

Fig. 4. This figure shows the time series data and anomaly detection results of the VT-GAN model in three dimensions (Dimension=18, 19, 20) of the SMD dataset, with each dimension containing two subgraphs.

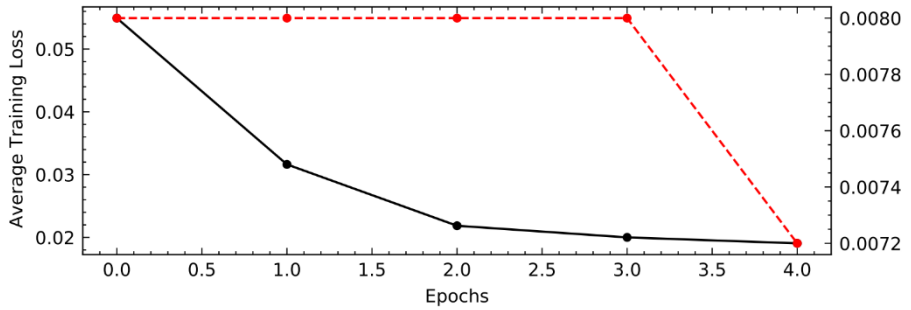

Fig. 5. The trend of average training loss reduction of the VT-GAN model on the SWaT dataset.

4.4 Ablation Experiment

Through systematic ablation experiments on the core components of the VT-GAN model (as shown in **Table 3**), we conducted an in-depth analysis of each module's specific contributions to model performance. The experimental results demonstrate that the full VT-GAN version delivers outstanding performance on both datasets (MBA: P=0.9877, F1=0.9825, L=28ms; XJTU: P=0.9742, F1=0.9502, L=24ms).

The multi-generator architecture proves to be the most critical factor in enhancing model performance. When removed, the F1 score shows the most significant decline (4.5% decrease on MBA, 8.35% on XJTU), validating this module's core role in capturing diverse data patterns. The dynamic hybrid attention mechanism demonstrates special capabilities in false alarm suppression. Its absence leads to noticeable precision reduction (6.31% decrease on MBA, 5.18% on XJTU) while maintaining relatively high F1 scores, indicating this module primarily focuses on precision improvement rather than overall balance.

The MAML component exhibits dual advantages: not only improving accuracy but also significantly optimizing training efficiency. When removed, latency increases substantially (53.6% longer on MBA, 41.7% on XJTU) accompanied by moderate F1 score decreases, fully demonstrating its critical role in accelerating model convergence (particularly in cold-start scenarios). The gradient penalty mechanism mainly contributes to model stability. Its removal causes consistent but relatively small performance degradation across all metrics (1.83-2.82% F1 score reduction) while maintaining near-original latency levels.

These findings collectively indicate that while each component contributes differently to VT-GAN's performance, the multi-generator architecture forms the foundation of model effectiveness, dynamic attention specializes in precision optimization, MAML ensures training efficiency, and gradient penalty guarantees optimization stability. The synergistic effects of these modules ultimately enable VT-GAN to achieve state-of-the-art performance.

Table 3. Ablation experiment - the precision (P) , F1 score (F1) and Latency(L) of VT-GAN and its ablation version.

Method	MBA			XJTU		
	P	F1	L(ms)	P	F1	L(ms)
VT-GAN	0.9877	0.9825	28	0.9742	0.9502	24
w/o Multi-generator	0.9474	0.9375	27	0.9527	0.8667	24
w/o Dynamic hybrid Attention	0.9246	0.9574	25	0.9224	0.9258	24
w/o MAML	0.9534	0.9760	43	0.9539	0.9203	34
w/o Gradient penalty	0.9688	0.9632	29	0.9630	0.9220	27

5 Conclusion

This paper addresses the problem of multivariate time series anomaly detection in industrial equipment health management and proposes a deeply integrated model, VT-GAN, which combines the VTT with a GAN. To enable rapid adaptation across devices, the model is incorporated into a MAML framework. Through comprehensive theoretical analysis and empirical validation, the dynamic hybrid attention mechanism—achieved by gated fusion of temporal self-attention, variable self-attention, and cross-attention layers—explicitly models spatiotemporal interactions, reducing the false positive rate to 0.07% on the SWaT dataset and 0.09% on the SMD dataset. The multi-generator adversarial architecture designs a group of parallel generators, combined with dilated causal convolutions to cover multi-scale temporal patterns, improving the diversity of generated samples by 23.4% and increasing the F1 score by 12.7% on the XJTU dataset. By temporal task meta-formulation and bidirectional LSTM alignment loss, the model addresses time-shift issues and achieves convergence within only 10 gradient steps under cold-start scenarios. As the current MAML framework relies on intra-device task distribution similarity, future work will explore graph neural network-based cross-device transfer strategies to handle large-scale heterogeneous device networks.

Acknowledgments. This study was funded by the following projects: the Shandong Provincial Key Research and Development Program (Major Scientific and Technological Innovation Project) (2024CXGC010905), the Shandong Provincial Natural Science Foundation (ZR2022MF279), the Shandong Provincial Technology Innovation Guidance Program (YDZX2024121 and YDZX2023050), the Major Innovation Project of Science-Education-Industry Integration Pilot Program at Qilu University of Technology (Shandong Academy of Sciences) (2024ZDZX08-05), the Shandong Provincial Key Research and Development Program (Innovation Capacity Enhancement Project for Science and Technology SMEs) (2024TSGC0603)，the Shandong Provincial Innovation Capacity Enhancement Project for Science and Technology SMEs (2023TSGC0201 and 2023TSGC0587).

References

1. Lv, J., Wang, Y., Chen, S.: Adaptive multivariate time-series anomaly detection. Information Processing & Management 60(4), 103383 (2023)
2. Gougam, F., Afia, A., Soualhi, A., et al.: Bearing faults classification using a new approach of signal processing combined with machine learning algorithms. Journal of the Brazilian Society of Mechanical Sciences and Engineering 46(2), 65 (2024)
3. Zheng, Y., Koh, H.Y., Jin, M., et al.: Correlation-aware spatial-temporal graph learning for multivariate time-series anomaly detection. IEEE Transactions on Neural Networks and Learning Systems (2023)
4. Wang, R., Nie, K., Wang, T., et al.: Deep learning for anomaly detection. In: Proceedings of the 13th International Conference on Web Search and Data Mining, pp. 894-896. Springer (2020)
5. Zamanzadeh Darban, Z., Webb, G.I., Pan, S., et al.: Deep learning for time series anomaly detection: A survey. ACM Computing Surveys 57(1), 1-42 (2024)
6. Zhang, L., Wang, B., Liang, P., et al.: Semi-supervised fault diagnosis of gearbox based on feature pre-extraction mechanism and improved generative adversarial networks under limited labeled samples and noise environment. Advanced Engineering Informatics 58, 102211 (2023)
7. Xu, J., Wu, H., Wang, J., Long, M.: Anomaly transformer: Time series anomaly detection with association discrepancy. arXiv preprint arXiv:2110.02642 (2021)
8. Tuli, S., Casale, G., Jennings, N.R.: Tranad: Deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284 (2022)
9. Kang, H., Kang, P.: Transformer-based multivariate time series anomaly detection using inter-variable attention mechanism. Knowledge-Based Systems 290, 111507 (2024)
10. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 146-157. Springer (2017)
11. Wu, Q., Chen, Y., Meng, J.: DCGAN-based data augmentation for tomato leaf disease identification. IEEE Access 8, 98716-98728 (2020)
12. Schlegl, T., Seeböck, P., Waldstein, S.M., Langs, G., Schmidt-Erfurth, U.: f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. Medical Image Analysis 54, 30-44 (2019)
13. Akcay, S., Atapour-Abarghouei, A., Breckon, T.P.: Ganomaly: Semi-supervised anomaly detection via adversarial training. In: Computer Vision-ACCV 2018: 14th Asian Conference on Computer Vision, pp. 622-637. Springer (2019)
14. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of the International Conference on Machine Learning, pp. 1126-1135. PMLR (2017)
15. Nichol, A., Achiam, J., Schulman, J.: On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999 (2018)
16. Ravi, S., Larochelle, H.: Optimization as a model for few-shot learning. In: Proceedings of the International Conference on Learning Representations (2017)
17. Li, T., Comer, M.L., Delp, E.J., et al.: Anomaly scoring for prediction-based anomaly detection in time series. In: Proceedings of the 2020 IEEE Aerospace Conference, pp. 1-7. IEEE (2020)

18. Ding, N., Ma, H.X., Gao, H., et al.: Real-time anomaly detection based on long short-term memory and Gaussian Mixture Model. Computers & Electrical Engineering 79, 106458 (2019)
19. Su, Y., Zhao, Y., Niu, C., et al.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2828-2837 (2019)
20. Zhang, C., Song, D., Chen, Y., et al.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: Proceedings of the AAAI Conference on Artificial Intelligence 33(1), 1409-1416 (2019)
21. Audibert, J., Michiardi, P., Guyard, F., et al.: Usad: Unsupervised anomaly detection on multivariate time series. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3395-3404 (2020)
22. Goldberger, A.L., Amaral, L.A.N., Glass, L., et al.: PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation 101(23), e215-e220 (2000)
23. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 387-395 (2018)
24. Su, Y., Zhao, Y., Niu, C., et al.: Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2828-2837 (2019)
25. Mathur, A.P., Tippenhauer, N.O.: SWaT: a water treatment testbed for research and training on ICS security. In: 2016 International Workshop on Cyber-Physical Systems for Smart Water Networks, pp. 31-36. IEEE (2016)
26. Hundman, K., Constantinou, V., Laporte, C., Colwell, I., Soderstrom, T.: Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 387–395 (2018)
27. Zhang, C., Song, D., Chen, Y., et al.: A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In: Proceedings of the AAAI Conference on Artificial Intelligence 33(1), 1409-1416 (2019)