This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

Generative Models for Time Series Anomaly Detection: A Survey

Jie Cao, *Member, IEEE*, Jiawei Miao, Haicheng Tao, Youquan Wang, Jia Wu, *Senior Member, IEEE*, Zidong Wang, *Fellow, IEEE*, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Time series anomaly detection (TSAD) is a fundamental practice in information management, aimed at identifying unusual patterns in temporal datasets. This process is critical to maintaining the integrity and reliability of systems. Recently, generative models have significantly advanced the capabilities of artificial general intelligence, presenting novel methodologies to understand and interpret complex data structures. In this review, we examine the latest advancements in applying generative models to TSAD and highlight how these models present a paradigm shift in detecting and analyzing anomalies within sequential data. In particular, we first present the background information, including definitions of key concepts, a taxonomy of anomaly types, and the distinction between generative and discriminative models in time series data. Then, we investigate a range of generative models, offering mathematical summaries of the predominant techniques in TSAD. Furthermore, we provide a summary of the datasets and propose recommendations for appropriate generative methods tailored to various application domains. Finally, we address the significant challenges in current research and propose potential directions for future study.**

***Impact Statement*—Generative approaches have shown exceptional performance in TSAD. Various emerging generative methods have expanded in this field, signaling a shift from traditional to deep generative techniques. Although some studies have reviewed the use of generative models like GANs and Transformers in time series, a comprehensive synthesis of these methods for anomaly detection is still lacking. This paper reviews existing work on mainstream generative approaches for this purpose. We summarize datasets and analyze methods suited to different dataset characteristics, providing tailored recommendations for various application domains. The goal of this paper is to offer researchers a reliable review and valuable guidance for future work.**

***Index Terms*—Deep learning, generative models, time series anomaly detection, survey**

I. Introduction

TIME series data, owing to its sequential structure, has broad applications across various areas, including healthcare [149], finance [29, 73], and energy [122]. Time series analysis is of significant importance in the field of information management systems [43, 71]. It provides valuable insights into patterns, trends, and deviations that have a significant impact on decision-making processes.

A new focus in the research of temporal data is Time Series Anomaly Detection (TSAD), which helps to identify anomalies in temporal data streams, thereby mitigating potential risks inherent in real-world systems [109, 128, 146]. The challenges encountered in the task of TSAD fall into three primary dimensions. First, the dynamic and evolving nature of temporal data presents a significant challenge. Second, the presence of noise and outliers obfuscates meaningful patterns and complicates the identification of true anomalies. Finally, the scarcity of labeled anomaly data makes general supervised algorithms unusable.

Traditional statistical-based methods perform anomaly detection by identifying the boundary or difference between anomalies and normal points. The K-Nearest Neighbors (KNN) method [23, 140] determines anomalies by comparing the distance between the target points and their nearest neighbors.This means points with significantly greater distances to their adjacent data points, indicating potential anomalies. The Local Outlier Factor (LOF) method [65, 117] detects anomalies in temporal data by assessing the density differences between a data point and its neighbors within a specified locality. It flags points with substantially lower densities as potential anomalies, suggesting significant density differences from their neighbors. OC-SVM (One-Class Support Vector Machine) [83] is an unsupervised learning algorithm. The core concept of OC-SVM is to construct a decision boundary that closely encapsulates the normal patterns of the data, with any points significantly deviating from this boundary being identified as outliers. In time series analysis, OC-SVM employs a suitable kernel function to model the temporal features, which helps in distinguishing between normal and anomalous patterns.

In recent years, the striking rise of generative models [1, 17, 20, 98, 143] has provided a powerful avenue for TSAD. Unlike discriminative models [5] that focus on delineating decision boundaries, generative models work on understanding and learning the underlying data distribution. The capacity to generate samples mirroring the distribution of the training set gives generative models a distinct advantage in TSAD. Bayesian networks [40] are graphical models that represent the conditional dependencies among variables to model the joint distribution of multivariate data. By constructing models of dependency relationships among key variables in time series data, Bayesian networks facilitate the inference and

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

J. Cao, J. Miao, and X. Wu are with the Hefei University of Technology, Hefei, 230009, China (e-mail: cao_jie@hfut.edu.cn; jiaweimiao@mail.hfut.edu.cn; xwu@hfut.edu.cn).

H. Tao and Y. Wang are with the Nanjing University of Finance and Economics, Nanjing, 210023, China (e-mail: {haicheng.tao, youq.wang}@gmail.com).

J. Wu is with the Department of Computing, Macquarie University, Sydney, Australia (e-mail: jia.wu@mq.edu.au).

Z. Wang is with the Department of Computer Science, Brunel University London, Uxbridge, UK (e-mail: Zidong.Wang@brunel.ac.uk).

Corresponding author: J. Miao (e-mail: jiaweimiao@mail.hfut.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

TABLE I: A Comparison Between Existing Surveys on Time Series.

Surveys	TS	TSAD	TSDL	GTSAD	Uni	Mul	Anomaly Detection			Source Code	Dataset	
							MR	AS	thresholds		Real	Synthetic
Our survey	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Wen et al. [135]	✓	-	✓	-	-	✓	-	-	-	-	-	-
Lin et al. [94]	✓	-	✓	-	-	✓	-	-	-	-	-	-
Blázquez et al. [11]	✓	✓	-	-	✓	✓	-	-	-	✓	-	-
Cook et al. [32]	✓	✓	-	-	✓	✓	-	-	-	-	-	-
Zhang et al. [146]	✓	✓	-	-	-	-	-	-	-	-	-	-
Shaukat et al. [119]	✓	✓	-	-	-	-	-	-	-	-	-	-
Braei et al. [13]	✓	✓	✓	-	✓	-	-	-	-	-	✓	✓
Chen et al. [24]	✓	✓	✓	-	-	✓	-	-	-	-	-	-
Darban et al. [37]	✓	✓	✓	-	✓	✓	-	-	-	-	✓	✓
Freeman et al. [46]	✓	✓	✓	-	✓	-	-	-	-	-	-	-
Ho et al. [63]	✓	✓	✓	-	-	✓	-	-	-	-	-	-
Li et al. [82]	✓	✓	✓	-	-	✓	-	-	-	✓	-	-
Wang et al. [132]	✓	✓	✓	-	✓	✓	✓	✓	-	✓	✓	-

* TS: Time Series, TSAD: Time Series Anomaly Detection, TSDL: Time Series with Deep Learning, GTSAD: Generative Time Series Anomaly Detection.

* Uni: Univariate, Mul: Multivariate, MR: Mathematical Representation, AS: Anomaly Score.

* ✓: included, -: not included.

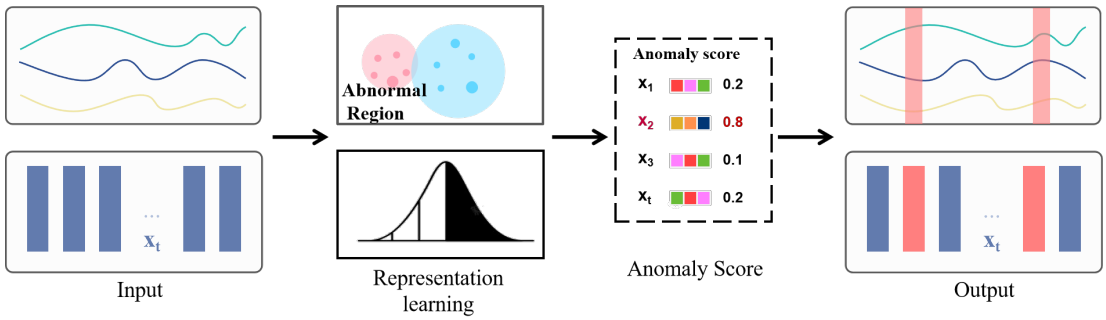

Fig. 1: Total Process of the Generative MTS anomaly detection.

prediction of normal behavior at each time point. In real-time monitoring, significant deviations of observed values from the probability distributions predicted by the model indicate potential anomalies, thereby enabling the discovery of anomalies within temporal data. Hidden Markov Models (HMMs) [55] can be used to model the state transitions and observation probabilities of the data and to identify outliers or anomalous patterns that differ significantly from normal patterns. The effectiveness of HMMs is rooted in their capability to discern underlying structures and temporal correlations in the data, which has led to their excellent success in TSAD. However, for data with high dimensionality, complex patterns and large data volumes, traditional statistical models may face great difficulties. The proposal of novel approaches is imminent.

The advent of deep neural network technology [15, 127] has marked a new era in TSAD. Deep learning models, including Variational Autoencoder (VAE) [41, 77] and Generative Adversarial Networks (GAN) [33, 54], have significantly enhanced the accuracy and performance of TSAD. For example, LSTM-VAE [107] employs LSTMs as both encoder and decoder, effectively representing and reconstructing time series data; BeatGAN [149] utilizes autoencoders as the generator within a GAN framework, providing stability and regularization to the reconstruction process. Additionally, methods based on normalizing flows [39, 106] and diffusion models [34, 99, 141] are continuously emerging.

As for TSAD, many research teams have explored and summarized this field. As shown in Table I, Blázquez et al. [11], Cook et al. [32], Zhang et al. [146], and Shaukat et al. [119] reviewed traditional TSAD approaches, focusing mainly on statistical methods and machine learning methods. Meanwhile, Braei et al. [13], Chen et al. [24], Darban et al. [37] and Freeman et al. [46] delved into deep learning-based TSAD approaches and provided a comprehensive categorization of these approaches. In addition, Wen et al. [135] and Blázquez et al. extensively summarize specific deep learning models such as GANs, Transformers, and Diffusion models in time series. Other studies have focused on specific applications of TSAD in areas such as smart grid [146], Internet of Things (IoT) [32], and TSAD [63, 82]. These studies provide important insights into the roles and challenges of TSAD techniques in real-world applications. However, there is a lack of systematic reviews on TSAD generative methods. It suggests that this is an unexplored area of research. This paper aims to offer researchers a comprehensive overview of generative approaches to TSAD, offering valuable advice and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

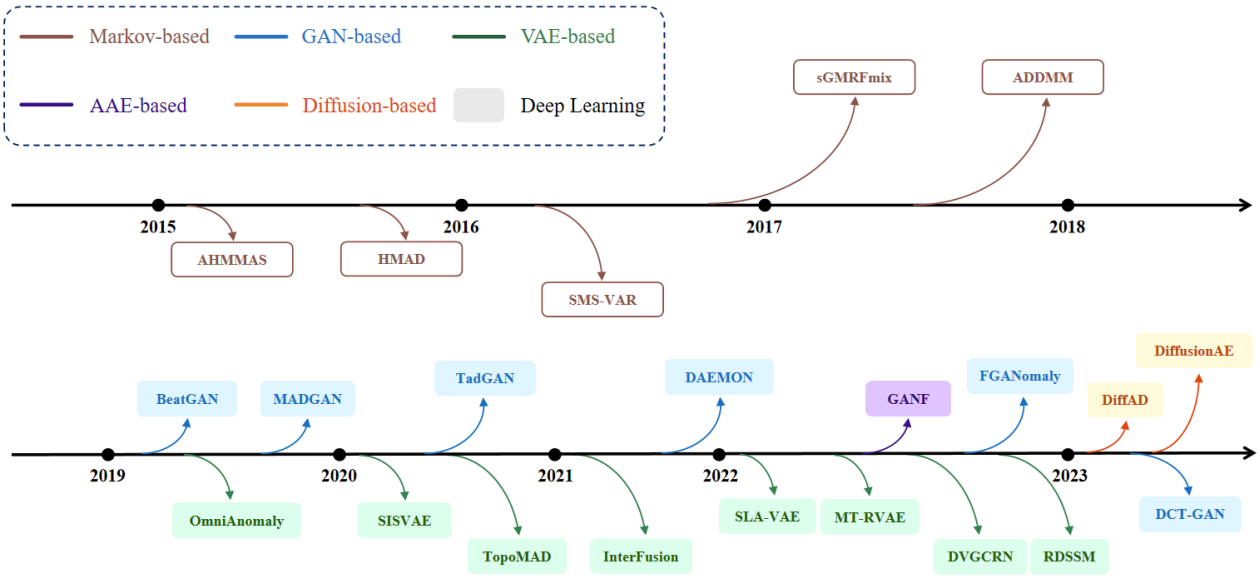

Fig. 2: An overview of the development of representative generative TSAD models. Those with corresponding background colors are deep learning models.

guidelines for future study in this area.

The organization of the subsequent chapters in this paper is outlined as follows:

- **Basics:** Section 2 gives some definitions of temporal data and anomaly detection, generalizes the types of anomalies, and summarizes the common generative and discriminative methods in this area.
- **Mainstream Generative Methods:** Section 3 presents six types of mainstream TSAD methods and analytically reviews the specific work of these methods.
- **Libraries:** Section 4 discusses the novel metrics for TSAD and summarizes the frequently utilized datasets. In addition, we expand to recommend effective methods for different datasets. We also summarize the applications of TSAD in various fields.
- **Future Directions:** In Section 5, we analyze the challenges encountered in TSAD and propose some targeted directions for future research.

II. Definitions and Background Concepts

A. Time Series Data

Time series: Let $\mathbf{X} \in \mathbb{R}^{N\times d}$ represent a time series with N sequential observations $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and d is the dimension of the time series. When $d = 1$, the time series is univariate; otherwise, it's multivariate.

Univariate time series(UTS): A sequence of data points, $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, where each data point x_i represents the observation of a single variable at the i^{th} equally spaced time interval. The set $T = \{1, 2, \ldots, N\}$ denotes the discrete time steps at which these observations occur.

Multivariate time series(MTS): A set of multiple univariate time series. Formally, an MTS can be represented as a matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^\top$, where $\mathbf{x}_t = (x_t^1, x_t^2, \ldots, x_t^d)$ is a snapshot of d variables at time t, and each x_t^j is the j^{th} variable at the time step t.

B. Anomaly detection

Time series anomaly detection: The goal of anomaly detection is to provide an anomaly vector $\mathbf{S} \in \{0, 1\}^N$ when given a time series $\mathbf{X}$. $\mathbf{S}[i] = 1$ if the time point $\mathbf{x}_i$ is judged as an anomaly, otherwise $\mathbf{S}[i] = 0$.

Generative and Discriminative time series anomaly detection methods: Generative methods primarily aim to study the distribution characteristics of normal time series to generate new samples with the same distribution as the training data. Discriminative methods focus on directly learning the decision boundary to distinguish between normal and anomalous time series. The overall architecture of deep generative models for TSAD is depicted in Figure 1.

C. Types of Anomalies

1) Time-Dependent Anomaly Patterns: These anomalies are primarily related to the temporal dependencies inherent in the time series data and include:

Spike Anomalies: Spike anomalies refer to sudden extreme values or fluctuations in a time series that significantly deviate from surrounding data points. For example, in stock trading data, a sudden surge or drop in the price of a stock within a short period may indicate a spike anomaly.

Collective Anomalies: Collective anomalies indicate long periods of anomalous states in a time series, which may be caused by system failures, persistent anomalous events, etc. For instance, if the response time of a network server consistently exceeds the average response time for an extended period, it could signal a collective anomaly due to server malfunctions or network congestion.

Seasonal Anomalies: Seasonal anomalies occur when data points within certain cycles in a time series significantly deviate from the expected pattern. For instance, in yearly sales data, a seasonal product suddenly experiencing a surge in sales during off-season periods may indicate the presence of a seasonal anomaly.

Trend Anomalies: Trend anomalies refer to non-periodic long-term trend changes in a time series that either go against the expected trend or significantly deviate from it. For example, in weather data, if the temperature in an area shows a gradual upward trend over time, contrary to the expected seasonal temperature change, this may indicate a trend anomaly.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

2) Variable-Related Anomaly Patterns: These anomalies are associated with the relationships between various variables in the time series, including:

Sequence Synchronization Anomalies: Sequence synchronization anomalies occur when the synchronization or temporal alignment between related sequences in a multivariate time series is disrupted. This anomaly manifests as one or more sequences deviating from their normal synchronization patterns with others, leading to a breakdown in the overall coherence of the system. Sequence synchronization anomalies are particularly significant in time-sensitive systems, such as industrial control or network communications, where maintaining temporal synchronization is crucial.

Cross-Sequence Relationship Anomalies: Cross-sequence relationship anomalies refer to changes in the relationships between multiple sequences in a multivariate time series. Such anomalies cannot be detected by analyzing individual sequences in isolation but require a holistic analysis of the relationships between sequences. These anomalies are closely tied to the physical significance of the data, often reflecting abnormal deviations in the coupling or collaborative patterns between different variables in the system. However, these types of anomalies are often underrepresented in existing classification frameworks, making them easy to overlook.

D. Generative Models for Time series anomaly detection

TSAD remains a hot topic. Early discriminative methods focused on detecting anomalies distinct from known normal patterns. In contrast, generative approaches emphasize learning data distributions and producing new samples. The advancement of generative methods in time series is attributed to progress in deep learning and probabilistic modeling. These methods provide flexible and accurate approaches to capturing data distributions, thereby pushing the frontier of time series forward.

After investigating generative methods for TSAD, we categorize them as follows:

Markov Methods [31, 113, 118], initially for speech recognition, model the probabilistic relationship between observation and state sequences in time series. They offer a probabilistic framework for generative anomaly detection methods.

Variational Autoencoders (VAEs) [27, 93, 107, 111] applied successfully in TSAD, aim to train the probabilistic encoder $q_\phi(\mathbf{z}|\mathbf{x})$ (parameterized by ϕ) to map temporal data into a low-dimensional embedding $\mathbf{z}$. Concurrently, the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ (parameterized by θ) is tasked with reconstructing sequences from $\mathbf{z}$.

Generative Adversarial Networks (GANs) [85, 116, 130] leverage adversarial learning for implicit time series data distribution modeling, employing a generator (G) and discriminator (D). Specifically, the generator learns the time series distribution to produce new sequences, while the discriminator distinguishes real from generated ones.

Normalizing Flows (NFs) map time series into latent variables $\mathbf{z}$ using invertible functions $f(\mathbf{x})$. They model distributions by tracking density changes through Jacobian matrices. Function $f^{-1}(\mathbf{z})$ produces new samples from latent variables, with training maximizing log-likelihood via gradient descent. [36] enhances Normalizing Flows with Bayesian networks to more accurately estimate joint densities, achieving unsupervised anomaly detection across multiple sequences.

Diffusion Models, also known as diffusion probabilistic models, gradually add Gaussian noise to transform a time series into pure Gaussian noise $\mathbf{z}$. They then generate a new time series by gradually denoising $\mathbf{z}$ until it approximates the true data distribution. Some scholars [138] proposed a diffusion-based anomaly detection method that uses weighted incremental diffusion, effectively mitigating the impact of anomalies by capturing long-range dependencies from selected normal points.

Adversarial Autoencoder (AAE) [101] is an improved neural network architecture of autoencoders(AE), which skillfully combines AE with GAN. It aims to learn efficient representations of time series in an unsupervised manner. In AAEs, the encoder-decoder architecture is augmented with a discriminator network, similar to those used in generative adversarial networks (GANs). The encoder maps input data into latent variables $\mathbf{z}$, the decoder reconstructs the data from the latent variables $\mathbf{z}$, and the discriminator distinguishes between temporal data from the true sequence and those generated by the decoder.

Other Methods, such as Denoising Autoencoder [72, 145], Masked Autoencoder [51, 125], and Boltzmann Machine [62, 80], have found application in other time series tasks like forecasting [38, 91] and classification [70, 148]. However, the discussion and utilization of them in anomaly detection tasks are relatively limited.

III. Mainstream Methods for Time Series Anomaly Detection

A. Markov Models for time series anomaly detection

Markov Model is a classic statistical model widely used in natural language processing (NLP) tasks such as speech recognition, part-of-speech tagging, phoneme-to-grapheme conversion, and probabilistic grammars. Over years of development, particularly its successful application in speech recognition, the Markov Model has become a general and effective statistical tool. Currently, it is still considered one of the most successful methods for implementing fast and accurate speech recognition systems.

The challenges of TSAD include the temporal dependence and dynamics of the data, for which Markov methods tend to capture potential patterns and transitions in temporal data through state transfer modeling. The model is trained as a predictor to simulate normal behavior and can be used to generate predictions for points or subsequences of the nearest window. This is particularly useful in real industrial scenarios where normal behavior is abundant but anomalous behavior is scarce. Anomalies can be detected by comparing the differences or residuals between the actual observations and the model predictions. This can be expressed as:

$$\begin{aligned} \hat{\mathbf{x}}_t &= \text{Predictor}(\mathbf{x}_{t-1}, \mathbf{z}_{t-1}) \\ \|\mathbf{x}_t - \hat{\mathbf{x}}_t\| &> \text{threshold} \end{aligned} \tag{1}$$

Here, $\hat{\mathbf{x}}_t$ and $\mathbf{x}_t$ are the predicted and true values at time t, respectively. $\mathbf{x}_{t-1}$ and $\mathbf{z}_{t-1}$ are the value and potential representations of the previous moment $t-1$. Function Predictor is specifically designed for each method separately.

Markov models are widely recognized for their computational efficiency and strong theoretical foundations, making them particularly effective for real-time detection tasks and offering high interpretability that is critical for problem diagnosis in industrial applications. However, their reliance on the assumption that future states depend solely on the current state inherently limits their ability to capture long-term dependencies, while the simplification of state transition dynamics may reduce their adaptability to complex patterns. Additionally, the assumption of stationary state transitions often conflicts with the non-stationary nature of real-world data, leading to potential performance degradation in dynamic environments.

HMAD [55] combines Hidden Markov Anomaly Detectors and One-class Support Vector Machines to deal with time series with latent dependency structure. Additionally, A DC (Difference of Convex Functions) algorithm is introduced to optimize the non-convex methods, improving the generalization ability and performance of the anomaly detection model. Similarly, Cao et al. [16] proposed the AHMMAS model, which integrates the adaptive hidden Markov model with the wavelet transform to provide an enhanced signal decomposition technique for time series. In addition, the introduction of an adaptive mechanism also compensates for the non-stationarity of the time series. However, the primary objective of these methods is to generate a single scalar representation of the outlier degree in a sample, often lacking direct variable-level information. For multivariate time series, sGMRFmix [69] combined Gaussian Markov Random Fields and Bayesian inference, which were able to filter out uncorrelated variables and effectively identify outliers. To cope with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

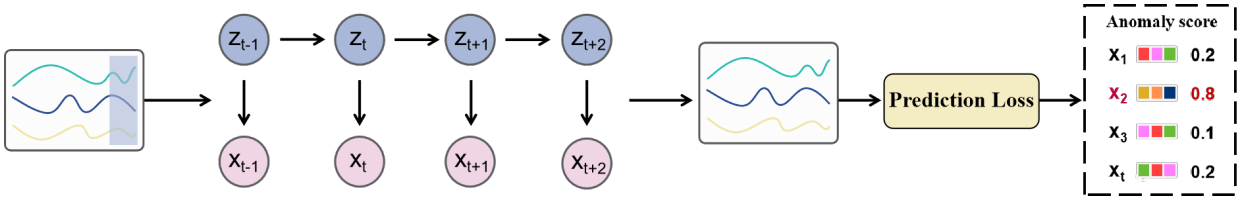

Fig. 3: Structure of the Markov-based TSAD.

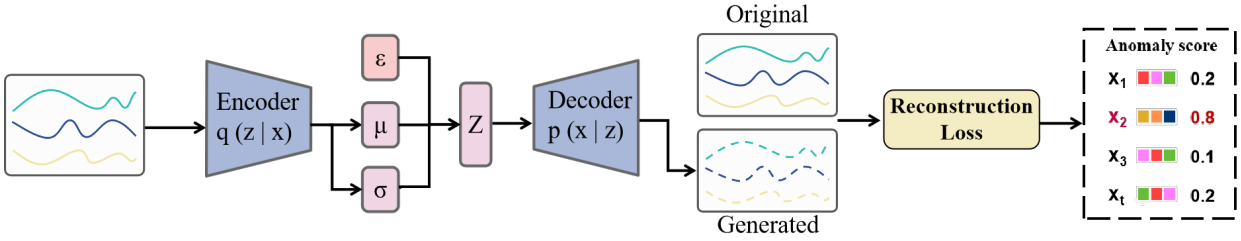

Fig. 4: Structure of the VAE-based TSAD.

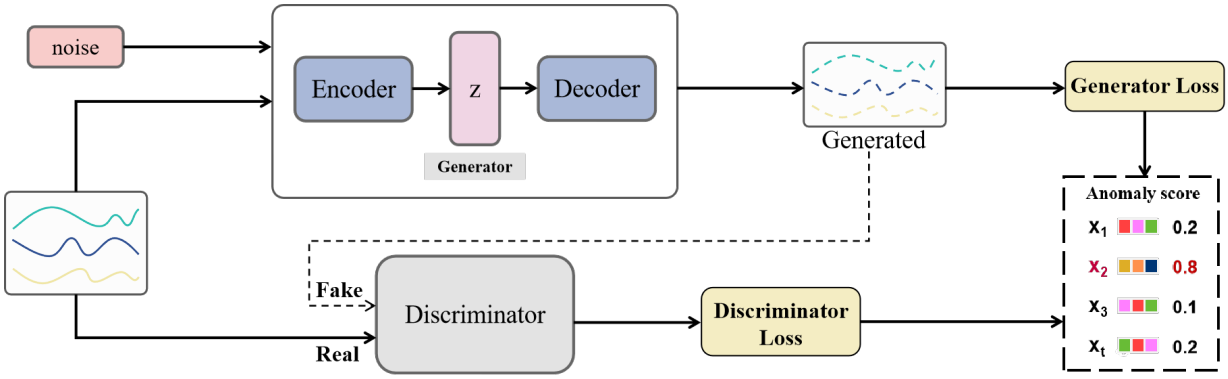

Fig. 5: Structure of the GAN-based TSAD.

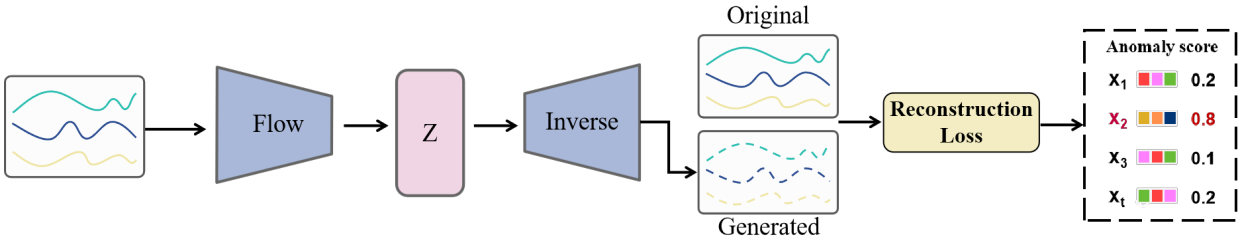

Fig. 6: Structure of the NF-based TSAD.

multivariate, variable-length datasets, SMS-VAR [102] uses a semi-Markov switching vector autoregressive model for heterogeneous time series. Owing to the lightweight and parallelizable nature of the model, the method can be used for online anomaly detection. ADDMM [115] introduces a dynamic Markov TSAD approach for sequence data, addressing limitations of traditional Markov chain techniques. Utilizing a sliding window and higher order Markov models, it balances memory length with sequence trends. An anomaly substitution strategy ensures continuous detection without compromising the integrity of the model.

B. Variational Autoencoders for time series anomaly detection

The Variational Autoencoder (VAE) is a classic generative model proposed by Kingma and Welling in 2013. It combines variational inference and deep learning, approximating complex probability distributions through neural networks. Due to its simplicity, stability, and clear theoretical foundation, VAE has received widespread attention.

In time series anomaly detection, VAE can learn normal patterns of time series and detect anomalies using reconstruction errors. VAE is suitable for various types of time series data, such as financial data and industrial sensor data.

Inference Network

The working principle of VAE is similar to an autoencoder, but instead of encoding inputs into a single point, it employs an inference network $q_\phi(\mathbf{z} \mid \mathbf{x})$ to encode them into a distribution, where ϕ represents its parameters. It maps a d-dimensional series $\mathbf{x}$ to a latent representation $\mathbf{z}$ with a lower dimension $k < d$.

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{x}), \log(\sigma^2_\phi(\mathbf{x}))) \tag{2}$$

Here, $\mathcal{N}$ denotes the Gaussian distribution with mean μ and variance σ^2. The functions $\mu_\phi(\mathbf{x})$ and $\sigma^2_\phi(\mathbf{x})$ are the outputs of the inference

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

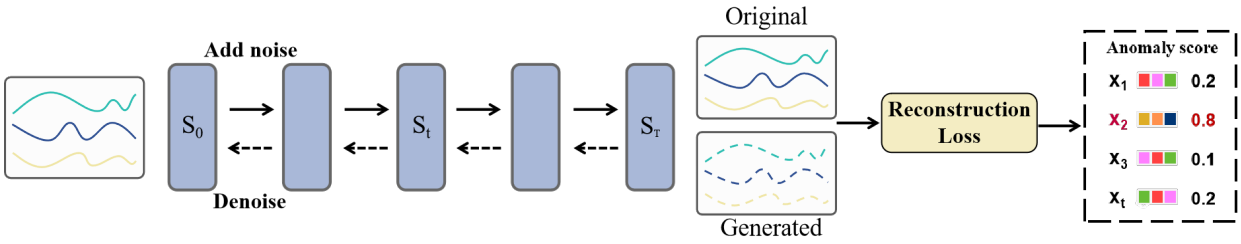

Fig. 7: Structure of the Diffusion Models for TSAD.

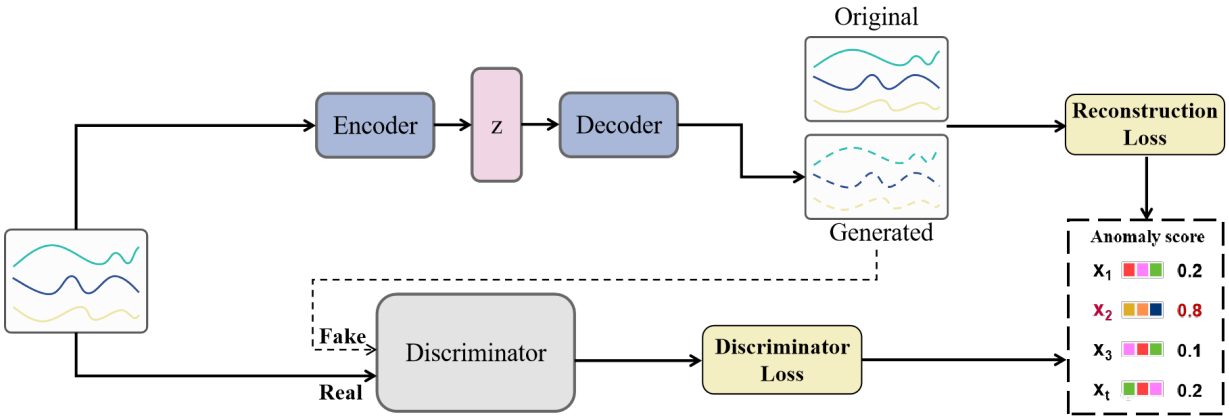

Fig. 8: Structure of the AAE-based TSAD.

network (encoder) parameterized by ϕ.

Generative Network

The sampling layer extracts a sample from the latent distribution and feeds it into the generative network $p_\theta(\mathbf{x} \mid \mathbf{z})$, where θ is its parameter, and the output is $Decoder(\mathbf{Z})$.

$$p_\theta(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}(\mu_\theta(\mathbf{z}), \sigma_\theta^2(\mathbf{z})) \tag{3}$$

ELBO (Evidence Lower Bound)

A commonly used variational inference method in VAEs is SGVB (Stochastic Gradient Variational Bayes). It optimizes the parameters ϕ and θ by maximizing the Evidence Lower Bound (ELBO), denoted as $\text{ELBO}(\theta, \phi; \mathbf{x})$:

$$\text{ELBO}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] - \text{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}) || p(\mathbf{z})\right) \tag{4}$$

Here,$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}$ represents the expectation over the distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$, KL is the KL divergence, and $p(\mathbf{z})$ is the prior distribution for latent variables.

Optimization Objective

$$\max_{\theta,\phi} \frac{1}{N} \sum_{i=1}^{N} \left[\mathbb{E}_{q_\phi \mathbf{z}|\mathbf{x}_i)}\left[\log p_\theta(\mathbf{x}_i \mid \mathbf{z})\right] - \text{KL}\left(q_\phi(\mathbf{z} \mid \mathbf{x}_i) || p(\mathbf{z})\right)\right] \tag{5}$$

Here, N represents the size of the training set, and $\mathbf{x}_i$ is the i-th sample in the training set.

To provide a comprehensive analysis of VAE-based methodologies, we summarize the three pivotal modules of the process: **1)Representation Learning:** Given a training set, models learn the representation of the normal sequences by means of the respective neural network modules. **2)Anomaly Score:** For the sequences in the test set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the anomaly score $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ corresponding to each point (subsequence) in the sequence is computed using the respective scoring method. **3)Thresholds:** Appropriate thresholds are selected, and points (sub-sequences) with anomaly scores greater than the thresholds are determined to be anomalous.

Representation Learning

Many state-of-the-art methodologies, including OmniAnomaly[123], SISVAE [87], and RDSSM [90], employ GRU (RNN) architectures as the primary encoding and decoding structures to capture temporal characteristics inherent in the sequence. In particular, approaches such as InterFusion [89] enhance the basic GRU structure by incorporating 1D convolutional layers and leveraging Two-view embeddings. This augmentation facilitates a more nuanced understanding of intermetric embeddings, allowing for the discernment of acquired temporal insights while preserving temporal consistency within these embeddings. TopoMAD [61] and DVGCRN [26] amalgamate RNNs with GCNs to emulate the spatial and temporal granularity correlations present in MTS. MT-RVAE [129] introduces the self-attention mechanism into the VAE framework and devises a comprehensive temporal encoding scheme. This innovation aims to capture latent correlations between sequences and encapsulate multiscale temporal information effectively.

Anomaly Score

The majority of methodologies[61, 87, 89, 90, 123] adopt a consistent anomaly scoring strategy, namely the reconstruction probability. DVGCRN [26] innovatively combines the reconstruction probability with prediction error, establishing a novel scoring criterion. SLA-VAE [67] employs the plain absolute reconstruction error. MT-RVAE [129] uses the absolute reconstruction error and integrates an EWMA method to facilitate the smoothing of reconstruction errors.

Thresholds

Anomaly scoring measures the extent of deviation observed in data instances. Nevertheless, in practical applications, a threshold remains essential to delineate between normal and anomalous patterns. Unlike conventional static thresholding methods, VAE-based approaches typically incorporate dynamic thresholds that account for data variability.

Several techniques, such as OmniAnomaly [123] and DVGCRN [26], utilize the POT(Peaks-Over-Threshold) [121] method to determine the threshold. InterFusion [89] adopts a threshold selection based on maximizing the global F1-score. SISVAE [87] conducts comprehensive experiments, exploring both globally optimal F1-score thresholds and those determined by F1-score rankings. Similarly,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

TABLE II: Summary of representative VAE methods.

Models	Representation Learning	Anomaly Score	Thresholds
OmniAnomaly [123]	$\begin{cases} Enc(\cdot) = f_\phi(\phi_{GRU}, \phi_{MLP}) \\ \mu_{\mathbf{z}}, \sigma_{\mathbf{z}} = Enc(\mathbf{x}) \\ \mathbf{z} = PlanarNF(\mu_{\mathbf{z}}, \sigma_{\mathbf{z}}) \\ q_\phi(\mathbf{z}\|\mathbf{x}) \sim N(\mu_{\mathbf{z}}, diag(\sigma_{\mathbf{z}}^2)) \\ Dec(\cdot) = f_\theta(\theta_{GRU}, \theta_{MLP}) \\ \mu_{\mathbf{x}}, \sigma_{\mathbf{x}} = Dec(\mathbf{z}) \\ p_\theta(\mathbf{x}\|\mathbf{z}) \sim N(\mu_{\mathbf{x}}, diag(\sigma_{\mathbf{x}}^2)) \end{cases}$	$\mathbf{S}_t = -\log(p_\theta(\mathbf{x}_t\|\mathbf{z}_{t-T:t}))$	$\begin{cases} th_0 = Q(p) \\ th^* \approx th_0 - \frac{\beta}{\gamma}\left(q\frac{N'}{N'_{th}}\right)^{-\gamma} - 1 \end{cases}$
SISVAE [87]	$\begin{cases} Enc(\cdot) = f_\phi(\phi_{GRU}, \phi_{MLP}) \\ Dec(\cdot) = f_\theta(\theta_{GRU}, \theta_{MLP}) \end{cases}$	$\mathbf{S}_t = -\log(p_\theta(\mathbf{x}_t\|\mathbf{z}_{t-T:t}))$	$th^* = \begin{cases} \text{argmax}_{th} F1(th) \\ \text{TOP}_k(S) \end{cases}$
TopoMAD [61]	$\begin{cases} GraphLSTM(\cdot) = LSTM(GCN(\cdot)) \\ Enc(\cdot) = f_\phi(\phi_{GraphLSTM}, \phi_{MLP}) \\ \mu_{\mathbf{z}}, \sigma_{\mathbf{z}} = Enc(\mathbf{x}, E) \\ q_\phi(\mathbf{z}\|\mathbf{x}) \sim N(\mu_{\mathbf{z}}, diag(\sigma_{\mathbf{z}}^2)) \\ Dec(\cdot) = f_\theta(\theta_{GraphLSTM}, \theta_{MLP}) \\ \mu_{\mathbf{x}}, \sigma_{\mathbf{x}} = Dec(\mathbf{z}, E) \\ p_\theta(\mathbf{x}\|\mathbf{z}) \sim N(\mu_{\mathbf{x}}, diag(\sigma_{\mathbf{x}}^2)) \end{cases}$	$\mathbf{S}_t = -\log(p_\theta(\mathbf{x}_t\|\mathbf{z}_{t-T:t}))$	$\begin{cases} Gap(th) = \min(S_{>th}) - \max(S_{<th}) \\ Sum(th) = \min(S_{>th}) + \max(S_{<th}) \\ \quad - 2 * \min(S_{<th}) \\ d(S_{<th}, S_{>th}) = \frac{Gap(th)}{Sum(th)} \\ th^* = \text{argmax}_{th} d(S_{<th}, S_{>th}) \end{cases}$
InterFusion [89]	$\begin{cases} Enc1 : \mathbf{z}2 = Conv1D(\mathbf{x}) \\ Enc2 : \mathbf{z}1 = GRU(DeConv1D(\mathbf{z}2)) \\ q(\mathbf{z}1, \mathbf{z}2\|\mathbf{x}) = q(\mathbf{z}1\|\mathbf{z}2, x)q(\mathbf{z}2\|\mathbf{x}) \\ Dec1 : \mathbf{z}1 = GRU(DeConv1D(\mathbf{z}2)) \\ Dec2 : \mathbf{x}' = MLP(\mathbf{z}1, \mathbf{z}2) \end{cases}$	$\mathbf{S}_t = -\log(p_\theta(\mathbf{x}_t\|\mathbf{z}_{t-T:t}))$	$th^* = \text{argmax}_{th} F1(th)$
RDSSM [90]	$\begin{cases} Enc(\cdot) = f_\phi(\phi_{BiGRU}, \phi_{MLP}) \\ Dec(\cdot) = f_\theta(\theta_{GRU}, \theta_{MLP}) \end{cases}$	$\mathbf{S}_t = -\log(p_\theta(\mathbf{x}_t\|\mathbf{z}_{t-T:t}))$	-
DVGCRN [26]	$\begin{cases} Enc(\cdot) = f_\phi(\phi_{LSTM}, \phi_{GCN}) \\ Dec(\cdot) = f_\theta(\theta_{LSTM}, \theta_{GCN}) \end{cases}$	$\begin{cases} \mathbf{R}_t = \log(p_\theta(\mathbf{x}_t\|\mathbf{z}_t)) \\ \mathbf{P}_t = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_2 \\ \mathbf{S}_t = \eta(-\mathbf{R}_t) + (1-\eta)\mathbf{P}_t \end{cases}$	$\begin{cases} th_0 = Q(p) \\ th^* \approx th_0 - \frac{\beta}{\gamma}\left(q\frac{N'}{N'_{th}}\right)^{-\gamma} - 1 \end{cases}$
SLA-VAE [67]	$\begin{cases} \mu_{\mathbf{z}}, \sigma_{\mathbf{z}} = Enc(\mathbf{x}) \\ q_\phi(\mathbf{z}\|\mathbf{x}) \sim N(\mu_{\mathbf{z}}, diag(\sigma_{\mathbf{z}}^2)) \\ \mu_{\mathbf{x}}, \sigma_{\mathbf{x}} = Dec(\mathbf{z}) \\ p_\theta(\mathbf{x}\|\mathbf{z}) \sim N(\mu_{\mathbf{x}}, diag(\sigma_x^2)) \end{cases}$	$S_t = \|\mathbf{x}_t - \mathbf{x}'_t\|_2$	$\begin{cases} th_0 = Q(p) \\ th \approx th_0 - \frac{\beta}{\gamma}\left(q\frac{N'}{N'_{th}}\right)^{-\gamma} - 1 \\ th^* = \text{argmax}_{th} F1(th) \end{cases}$
MT-RVAE [129]	$\begin{cases} Enc(\cdot) = f_\phi(\phi_{Transformer}) \\ Dec(\cdot) = f_\theta(\theta_{Transformer}) \end{cases}$	$\begin{cases} \mathbf{R}_t = \|\mathbf{x}_t - \mathbf{x}'_t\|_2 \\ \mathbf{S}_t = \eta\mathbf{S}_{t-1} + (1-\eta)\mathbf{R}_t \end{cases}$	-

Notations: $\mathbf{x} \in \mathbb{R}^{n\times d}$ denotes the input time series data, $\mathbf{z} \in \mathbb{R}^{n\times k}$ represents the latent variable representation of $\mathbf{x}$ in a lower-dimensional space ($k < d$). $\mu_{\mathbf{z}}$ and $\sigma_{\mathbf{z}}$ are the mean vector and standard deviation vector of $\mathbf{z}$ output by the encoder. $q_\phi(\mathbf{z}|\mathbf{x})$ indicates the approximate posterior distribution defined by its parameters ($\mu_{\mathbf{z}}$, $\sigma_{\mathbf{z}}$). $p_\theta(\mathbf{x}|\mathbf{z})$ represents the generative distribution (likelihood). $\hat{\mathbf{x}} \in \mathbb{R}^{n\times d}$ denotes the reconstructed time series generated by the decoder. f_ϕ and f_θ refer to the encoder and decoder neural networks respectively. ϕ_{NN} and θ_{NN} specify the neural network architectures employed (e.g., MLP, 1D-CNN, LSTM, GRU, Transformer Encoder, or combinations).

* $Enc(\cdot)$: Encoder, $Dec(\cdot)$: Decoder.

* S_t: the anomaly score of time t.

* th_0: the initial threshold, th^*: the final threshold, $Q(p)$: the quantile function, β: the scale parameter of the Generalized Pareto Distribution (GPD), γ: the GPD shape parameter (tail index), q: A tuning factor adjusting threshold selection sensitivity, N': the observed number of exceedances over the initial threshold, N'_{th}: theoretical or expected excess quantity.

* $d(S_{<th}, S_{>th})$ define the distance of two anomaly scores sets $S_{<th}$ and $S_{>th}$ separated by a threshold th, max(S) denotes the maximal element in S, and min(S) denotes the minimal element.

SLA-VAE [67] initiates with a threshold derived from the POT method and subsequently refines it based on the principle of optimizing the F1-score. Furthermore, TopoMAD [61] posits a hypothesis wherein normal data anomalous scores reside within high-density regions, while anomalous data scores occupy low-density regions. Guided by this principle, threshold selection endeavors to maximize the distance between these two distinct regions.

The core assumption of the Variational Autoencoder (VAE) model is that time series data can be generated from low-dimensional latent variables. By leveraging a learnable approximate posterior distribution to replace the true posterior, VAE integrates the probabilistic generative framework with deep representation learning, offering a flexible solution for anomaly detection in time series. However, there are several limitations associated with VAEs: the ability to model temporal dependencies and long-term correlations is restricted, and their performance in detecting abrupt changes or changepoints is suboptimal. Additionally, the training process of VAEs suffers from instability due to the non-convex nature of Evidence Lower Bound (ELBO) optimization and the high variance of gradient estimation, resulting in fluctuating reconstruction errors. Furthermore, as the dimensionality of the latent space increases, the variance of reparameterization gradients grows exponentially, which imposes significant constraints on the application of VAEs in low-latency, real-time detection tasks. Despite these challenges, VAEs

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

demonstrate significant advantages in handling complex temporal patterns and high-dimensional multivariate time series. In unsupervised anomaly detection scenarios where large amounts of labeled data are unavailable and real-time performance is not a priority, VAEs exhibit substantial potential.

OmniAnomaly [123] uses a stochastic RNN for MTSAD. The method learns robust representations of normal time series patterns through techniques such as stochastic variable connection and planar normalizing flow. The model reconstructs the input data using these representations and determines anomalies based on reconstruction probabilities. Moreover, OmniAnnaly offers interpretability for detected anomalies by quantifying the contribution of each individual univariate series within the entity based on their respective reconstruction probabilities. SISVAE [87] combines a recurrent neural network and a variational autoencoder to parameterize the mean and variance at each time point with a flexible neural network to obtain a non-stationary model. In addition, a novel variational smoothing regularizer is proposed which provides robustness by penalizing the non-smooth output of the generated model. This work also discusses two anomaly detection criteria based on reconstruction probability and reconstruction error. As the complexity of a system increases, it becomes a challenge to effectively model the data collected from the various components of the system and to model the spatio-temporal dependencies among them. TopoMAD [61] leverages a combination of graph neural networks, LSTM, and VAE to form a novel neural network architecture that effectively models complex spatiotemporal dependencies in contaminated data. DVGCRN [26] combines EPN with Graph Convolutional Recurrent Network (GCRN) into a unified framework, thus learning the robust representations of MTS by considering both temporal, interrelationship and stochasticity characteristics. In addition, they combine reconstruction and prediction optimization objectives for inference to increase the stability of anomaly detection. The development of graph neural networks has provided a boost to inter-relational modeling of time series, but most of the methods are less effective in dealing with data with fewer dimensions or sparse inter-relationships between sequences. MT-RVAE [129] employs a self-attention mechanism to model interdependencies among time series, thereby mitigating the influence of feature dimensionality and relationship strength on algorithmic performance. InterFusion [89] employs two stochastic latent variables to jointly capture both inter-metric and temporal dependencies in multivariate time series. Furthermore, an MCMC-based method is proposed to derive plausible embeddings and reconstructions even at anomalous segments, enabling improved interpretation of anomalies. Unlike previous unsupervised models, SLA-VAE [67] adopts a semi-supervised VAE to detect anomalies. It further utilizes an active learning strategy to refine the online model using a limited number of uncertain samples.

Critically, empirical implementations reveal that subtle architectural adjustments (e.g., LSTM-VAE's hidden-to-input dimension ratio) and numerical stabilization techniques (e.g., Monte Carlo sampling for VAE reconstruction scoring) prove decisive to detection efficacy [27, 59, 103, 120, 126] .

C. Generative Adversarial Networks for time series anomaly detection

The Generative Adversarial Network (GAN) was initially proposed by Goodfellow et al. in 2014 and was originally used for image generation tasks. Due to its outstanding ability in sample generation, GAN-based anomaly detection methods have rapidly developed. GAN has achieved great success in tasks such as image generation, image translation, and video prediction, and researchers have also demonstrated its effectiveness in anomaly detection. However, at that time, the application of GAN to time series data was relatively rare, mainly because the complex temporal dependencies of such data posed significant challenges to generative modeling. It was not until 2020 that Bashar et al. proposed the reconstruction-based TAnoGAN [8] model, marking an early seminal application of GANs for anomaly detection in time series data.

The training process of a GAN follows an adversarial game principle. The ability of G and D is continuously improved by alternating training.

Here are the objective functions for GAN training, along with the basic formulas achieved through min-max optimization:

Generator's Objective Function (Minimization)

The objective of the generator is to generate sequences that are close to the distribution of true samples, making it challenging for the discriminator to distinguish. The loss function of G is usually defined as the probability of misleading the discriminator.

$$\min_G V(G,D) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log(1 - D(\mathbf{x}))] \tag{6}$$

Discriminator's Objective Function (Maximization)

The discriminator aims to differentiate between the original series and the generated ones. The discriminator's loss function is usually formulated as the summation of the probabilities of accurately categorizing the true sample and the generated sample.

$$\begin{aligned}\max_D V(G,D) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log D(\mathbf{x})] \\ +\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]\end{aligned} \tag{7}$$

Final GAN Objective Function

The overall GAN objective function is a combination of the generator and discriminator loss functions, which are usually trained by alternating optimization.

$$\begin{aligned}\min_G \max_D V(G,D) = \mathbb{E}_{\mathbf{x}\sim p_(\mathbf{x})}[\log D(\mathbf{x})] \\ +\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))]\end{aligned} \tag{8}$$

In practical training, SGD or its variants are commonly used to minimize and maximize the respective loss functions.

For GAN-based methods, we propose a generalized framework with three modules: **1)Data processing:** GAN-based methods often require clean data, and some methods are used to cull out anomalies in the training set. **2)Representation Learning.:** For for a given training set, the model learns patterns of normal sequences via the respective neural network modules to learn the patterns of normal sequences. **3)Anomaly Detection:** For the sequences in the test set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, the anomaly score $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ corresponding to each point (subsequence) in the sequence is computed using the respective scoring method. Points (sub-sequences) with anomaly scores greater than a threshold are determined to be anomalous.

Data processing

Most methods [25, 30, 64, 66] prefer clean data as the purity of data significantly impacts GAN performance. Given the sparsity of anomalies, some approaches [47, 116] default to training on normal data to avoid influencing the model's learning of normal data distribution. FGANomaly [44] introduces a pseudo-label generation method based on reconstruction error, enabling GANs to filter potential anomaly samples. Additionally, DAEMON [22] employs the spectral residual algorithm to clean potential outliers in the training dataset, enhancing VAE's accuracy in learning the normal distribution of time series.

Representation Learning

Several approaches [53] employ an RNN generator and discriminator as the foundational model for the GAN framework. Besides, methods like MAD-GAN[86] and TAnoGAN [8] use LSTM to capture temporal correlations in time series distribution. Some methods [30, 88, 124] utilize CNN as the foundational model, employing convolution operations to capture local patterns in the data for anomaly identification. Furthermore, some approaches [88] introduce attention mechanisms to capture relationships between variables, further enhancing anomaly detection performance. In contrast to the aforementioned methods, BeatGAN [149] and FGANomaly [44] use autoencoders as the generator for GAN, focusing on data reconstruction. Some methods deviate from the original GAN architecture, making subtle modifications. TadGAN [50] adopts a dual-discriminator architecture to assess the quality of both time series

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

TABLE III: Summary of representative GAN methods.

Models	Data processing	Representation Learning	Anomaly Detection
BeatGAN [149]	clean	$\begin{cases} G_D(\cdot) = CNN() \\ G_E(\cdot) = CNN() \\ \mathbf{z} = G_E(\mathbf{x}), \mathbf{x}' = G_D(\mathbf{z}) \\ D_{cnn} : \mathbf{x} \rightarrow [0,1] \end{cases}$	$\mathbf{S}_t = \|\mathbf{x}_t - \mathbf{x}'_t\|_2$
MADGAN [86]	clean	$\begin{cases} \mathbf{z} \leftarrow \text{Random Latent Space} \\ \mathbf{x}' = G_{rnn}(\mathbf{z}) \\ D_{rnn} : \mathbf{x} \rightarrow [0,1] \end{cases}$	$\begin{cases} Res_t = \|\mathbf{x}_t - G_{rnn}(\mathbf{z}_t)\| \\ Dis_t = -log(D_{rnn}(\mathbf{x}_t)) \\ \mathbf{S}_t = \eta Res_t + (1-\eta) Dis_t \end{cases}$
TadGAN [50]	clean	$\begin{cases} G_E(\cdot) = BiLSTM() \\ G_D(\cdot) = BiLSTM(BiLSTM()) \\ \mathbf{z} = G_E(\mathbf{x}), \mathbf{x}' = G_D(\mathbf{z}) \\ C_{\mathbf{x}}(\mathbf{x}) = Conv1D(BiLSTM(\mathbf{x})) \\ C_{\mathbf{z}}(\mathbf{z}) = Conv1D(BiLSTM(\mathbf{z})) \\ C_{\mathbf{x}} : \mathbf{x} \rightarrow (0,1) \\ C_{\mathbf{z}} : \mathbf{z} \rightarrow (0,1) \end{cases}$	$\begin{cases} Res_t = \|\mathbf{x}_t - G_D(G_E(\mathbf{x}_t))\| \\ Dis_t = -log(C_{\mathbf{x}}(\mathbf{x}_t)) \\ \mathbf{S}_t = \eta Res_t + (1-\eta) Dis_t \end{cases}$
DAEMON [22]	$\begin{cases} contaminated \\ spectral residual \end{cases}$	$\begin{cases} G_E(\cdot) = Conv1D() \\ G_D(\cdot) = Deconv1D() \\ \mathbf{z} = G_E(\mathbf{x}), \mathbf{x}' = G_D(\mathbf{z}) \\ D_D(\mathbf{x}) = Sigmoid(Conv1D(\mathbf{x})) \\ D_E(\mathbf{z}) = Sigmoid(Conv1D(\mathbf{z})) \\ D_D : \mathbf{x} \rightarrow (0,1) \\ D_E : \mathbf{z} \rightarrow (0,1) \end{cases}$	$\mathbf{S}_t = \|\mathbf{x}_t - G_D(G_E(\mathbf{x}_t))\|_1$
FGANomaly [44]	$\begin{cases} contaminated \\ pseudo-label \end{cases}$	$\begin{cases} G_E(\cdot) = Linear(BiLSTM()) \\ G_D(\cdot) = BiLSTM(Linear()) \\ \mathbf{z} = G_E(\mathbf{x}), \mathbf{x}' = G_D(\mathbf{z}) \\ D(\cdot) = Feedforward() \\ D : \mathbf{x} \rightarrow [0,1] \end{cases}$	$\mathbf{S}_t = \|\mathbf{x}_t - \mathbf{x}'_t\|_2$
DCT-GAN [88]	contaminated	$\begin{cases} \mathbf{z} \leftarrow \text{Random Latent Space} \\ G_E(\cdot) = Attention(CNN()) \\ \mathbf{x}' = G_E(\mathbf{z}) \\ D : \mathbf{x} \rightarrow [0,1] \end{cases}$	$\mathbf{S}_t = \|\mathbf{x}_t - \mathbf{x}'_t\|_2$

Notations: $\mathbf{x} \in \mathbb{R}^{n \times d}$ represents the input time series data, $\mathbf{z} \in \mathbb{R}^{n \times k}$ denotes the latent representation of $\mathbf{x}$ sampled from a simple distribution, and $\hat{\mathbf{x}} \in \mathbb{R}^{n \times d}$ indicates the time series reconstructed by the generator.

* $G(\cdot)$: Generator, $D(\cdot)$: Discriminator.

* $G_E(\cdot)$: Encoder of Generator which using an autoencoder structure, $G_D(\cdot)$: Decoder of Generator which using an autoencoder structure.

* $C_{\mathbf{x}}$ and $C_{\mathbf{z}}$ evaluate the quality of the original data $\mathbf{x}$ and the latent encoding $\mathbf{z}$ respectively, and similarly for D_D and D_E.

and latent space, addressing gradient instability and mode collapse issues. DAEMON [22] includes two GAN sets, with two generators acting as encoder and decoder, forming a shared VAE. The remaining two discriminators serve as independent modules, making the training of the variational autoencoder structure more robust and reducing overfitting.

Anomaly Detection

Many GAN-based TSAD methods[44, 88, 149] use the L2 norm reconstruction error. Some methods [22] employ the L1 norm, potentially because of its greater robustness. Additionally, TAnoGAN [8] attempts to consider a combination of sequence reconstruction error and latent variable reconstruction error. In contrast, DEGAN [58] uses discriminator results (discrimination probability) as the criterion for anomaly scoring. Building on this, some methods, such as TadGAN [50] and MAD-GAN [86], consider combining reconstruction error with discriminator results for anomaly detection.

Compared to probabilistic models such as Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs) do not require explicit assumptions about the data distribution of time series. This provides GANs with greater flexibility in handling time series with unknown or complex distributions. The architecture of GANs can capture the distribution of time series data and provide global regularization information during training, thereby alleviating overfitting problems. However, this is only effective when the training set is not contaminated by anomalous samples. If the training set contains anomalies, the model may capture the distribution of anomalous data, leading to degraded detection performance. Furthermore, GAN training is difficult to balance due to the need for alternating training of the generator and discriminator, making it prone to mode collapse and poor diversity in generated samples. Many approaches have been proposed to improve GAN-based anomaly detection methods. It is also worth noting that GANs were originally designed for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

independent and identically distributed (i.i.d.) data. When directly applied to time series, they may fail to adequately capture the dynamic characteristics inherent in sequential data. To address these limitations, numerous methods have been proposed to improve GAN-based anomaly detection approaches.

BeatGAN [149] enhances the robustness of the model by regularizing the reconstruction error and applying time series warping for data augmentation. MAD-GAN [86] uses a novel anomaly score called DR-score to detect anomalies by discrimination and reconstruction. In addition, MAD-GAN was the first to explore the issues of determining the optimal subsequence length as well as the potential model instability of the GAN-based approaches. Similarly, TadGAN [50] also explores new anomaly scoring methods. The combination of reconstruction error and critic output provides more robust anomaly scores, which helps to reduce the number of false positives and increase the number of true positives. Meanwhile, trained by cycle consistency loss, TadGAN allows robust reconstruction of time series data. DAEMON [22] innovates on the structure of a GAN by using two discriminators to inversely train a self-encoder to learn the normal patterns of a multivariate time series. To address extremely imbalanced and contaminated training datasets, FGANomaly [44] filters out potential anomalous samples using pseudo-labels before training the discriminator, thereby capturing the distribution of normal data as accurately as possible. A new training objective is designed for the generator, which encourages it to focus more on reliable normal data while ignoring anomalies.

D. Normalizing Flows for time series anomaly detection

Normalizing Flows provide an efficient and flexible way to fit arbitrary distributions by mapping simple distributions to complex ones through a series of optimizable mapping functions. In recent years, Normalizing Flows have achieved state-of-the-art (SOTA) performance in tasks such as speech generation. One advantage of Normalizing Flows over other methods is the convenience of data generation, such as using the simplest Linear Flows. However, Linear Flows are inefficient and slow to train because the computation of determinants has an $O(n^3)$ complexity. Methods such as Real NVP Flows have been designed to reduce computational complexity by making the Jacobian matrix of the transformation triangular.

Normalizing Flows provide precise likelihood estimation, facilitating the generation of high-quality data, but they require significant computational resources and involve complex model design. Additionally, the model's output is a probability distribution at each time point, which may affect the continuity and smoothness of the generated time series.

Normalizing Flows (NFs) map time series data into latent variables $\mathbf{z}$ using invertible functions $f(\mathbf{x})$. They model distributions by tracking density changes through Jacobian matrices. The inverse function $f^{-1}(\mathbf{z})$ produces new samples from latent variables, with training maximizing log-likelihood via gradient descent. [36] enhances Normalizing Flows with Bayesian networks to more accurately estimate joint densities, achieving unsupervised anomaly detection across multiple sequences.

In the NF framework, given a time series dataset $X = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$, in which $\mathbf{x}_i$ is the data sample at time i, the goal is to map this data to a latent variable space $\mathbf{Z} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_N\}$, where $\mathbf{z}_i$ corresponds to the latent variable associated with $\mathbf{x}_i$. NF accomplishes this mapping through an invertible function $f(\mathbf{x})$, such that $\mathbf{z} = f(\mathbf{x})$. The inverse function $f^{-1}(\mathbf{z})$ can then map latent representation back to the original series.

Density estimation in NF is achieved by tracking the transformations $f(\mathbf{x})$ using Jacobian matrices. For a sample $\mathbf{z}$ in the latent variable space $\mathbf{z}$, its corresponding probability density is given by:

$$p_{\mathbf{Z}}(\mathbf{z}) = p_X(f^{-1}(\mathbf{z})) \cdot \left|\det(J_f(f^{-1}(\mathbf{z})))\right| \tag{9}$$

Here, $p_X(\mathbf{x})$ is the probability density function of the original time series data $\mathbf{x}$, and $J_f(f^{-1}(\mathbf{z}))$ is the Jacobian matrix representing the mapping from $\mathbf{z}$ to $\mathbf{x}$ in the latent variable space.

Detection of anomalous points can be performed by comparing the probability density of data points in the latent variable space. Specifically, for a given time series data point $\mathbf{x}_i$, we first map it to the latent variable space to obtain $\mathbf{z}_i = f(\mathbf{x}_i)$, and then compute its probability density $p_{\mathbf{Z}}(\mathbf{z}_i)$. If this density is below a certain threshold, we can consider the data point as an anomaly.

In summary, NFs excel in high-dimensional, complexly distributed and noisily controllable time series scenarios (e.g., server monitoring, precision device sensing), but its computational overhead and temporal modeling shortcomings make it difficult to be directly applied to ultra-long sequences, strong real-time requirements, or highly data-contaminated tasks. The core conflict lies in the trade-off between reversibility and computational efficiency, and the conflict between global accuracy and local timing dependence of density estimation. Future breakthroughs may rely on the deep integration of NF with Neural ODEs or structured probabilistic models such as GraphNF.

E. Diffusion Models for time series anomaly detection

Diffusion models have rapidly emerged in various fields, significantly impacting computer vision (CV), natural language processing (NLP), and audio processing. Due to the availability of large and diverse datasets in these fields, diffusion models are often combined with large language models (LLM) or other foundational models, driving rapid progress in these areas. Inspired by non-equilibrium thermodynamics, diffusion models define a Markov chain with diffusion steps, gradually adding random noise to the data and then learning to reverse the diffusion process to generate the desired data samples from noise.

In recent years, diffusion models exhibit unique advantages and challenges in time series analysis. Its core advantage lies in its fine-grained generation capability: through a multi-step denoising process, the model is able to incrementally learn complex dynamic characteristics (e.g., nonlinear relationships, long-term dependencies), and compared with the adversarial training mechanism of GAN, the optimization objective of diffusion model based on maximum likelihood estimation ensures a more stable generation quality, which can effectively circumvent the problem of pattern collapse.

However, its application to time series faces two challenges. First, the native model undermodels the time series structure: the standard diffusion process assumes that data points are generated independently, ignoring causal dependencies in the time dimension. Although subsequent work (e.g., TimeGrad's introduction of RNNs to encode historical information) has partially addressed this issue, the underlying framework still requires targeted improvements. Second, high computational costs limit real-time applications: typical diffusion models require hundreds of iterations to generate samples, resulting in significantly higher inference latencies than single-step generation models such as GAN. Although knowledge distillation (e.g., Progressive Distillation) can compress the number of steps to less than 10, it is still difficult to meet the demand for millisecond real-time detection. We analyze the process of diffusion models in TSAD using Denoising Diffusion Probabilistic Models (DDPMs) as a representative.

Let $\mathbf{X}_0 \in \mathbb{R}^{N \times d}$ be the input MTS, where N is the sequence length and d is the number of features. In the forward process, we keep adding Gaussian noise to the input at the previous time:

$$q(\mathbf{X}_t|\mathbf{X}_{t-1}) = \mathcal{N}(\mathbf{X}_t; \sqrt{1-\beta_t}\mathbf{X}_{t-1}, \beta_t\mathbf{I}) \tag{10}$$

where $\beta_t \in (0, 1)$ is the fixed variance that increases linearly with t.

During the reverse process, the gradual elimination of noise from the impaired time series is characterized by the following formula:

$$p_\theta(\mathbf{X}_{t-1}|\mathbf{X}_t) = \mathcal{N}(\mathbf{X}_{t-1}; \mu_\theta(\mathbf{X}_t, t), \tilde{\beta}_t\mathbf{I}) \tag{11}$$

here, $\tilde{\beta}t = \frac{1-\bar{\alpha}t-1}{1-\bar{\alpha}_t}\beta_t$ is acquired through neural network training, where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}t = \prod s = 1^t \alpha_s$.

Instead of learning $\mu_\theta(X_t, t)$, the network ϵ_θ is trained to predict the noise $\epsilon \sim \mathcal{N}(0, I)$ given X_t. The loss function is:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

$$L = \|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\|_2^2 \tag{12}$$

At test time, Gaussian noise is added to the input $\mathbf{X}_0$ and then denoised back: $\mathbf{X}_0 \rightarrow \mathbf{X}_{\text{noisy}} \rightarrow \tilde{\mathbf{X}}_0$. Specifically, $\mathbf{X}_0$ is corrupted with M steps of Gaussian noise, then iteratively denoised for M steps to obtain the reconstruction $\tilde{\mathbf{X}}_0$. The distance between the initial and denoised data is used as the anomaly score.

DiffusionAE [110] applies the diffusion process to the reconstructed time series of the autoencoder (instead of the original series), making the model robust to different training anomaly ratios. At the same time, the diffusion process smooths the anomalous segments, leading to higher reconstruction errors and improved performance. Unlike previous prediction-based and reconstruction-based methods that use partial or complete data as observations for estimation, DiffAD [138] employs a density-ratio-based strategy with a flexible selection of normal observations that can be easily adapted to anomalous concentration scenarios.

F. Adversarial Autoencoder for time series anomaly detection

The Adversarial Autoencoder (AAE) was developed based on the VAE, with the unique introduction of adversarial training. Unlike the VAE, which uses the KL divergence to measure the distance between a predefined prior distribution $p(\mathbf{z})$ and the variational posterior $q(\mathbf{z}|\mathbf{x})$, AAE employs deterministic encoding. Both VAE and AAE attempt to explicitly find the probability density of real samples and find the optimal solution by minimizing the log-likelihood function. The latent encoding $p(\mathbf{z}|\mathbf{x})$ of VAE is usually a Gaussian distribution, and the VAE Encoder fits the mean and variance of this Gaussian distribution. In contrast, AAE's latent encoding is deterministic and can be directly obtained through the Encoder. In AAE, the autoencoder and adversarial network play roles in the reconstruction and regularization stages, respectively. In the reconstruction stage, the autoencoder minimizes the reconstruction error of input data by updating the encoder and decoder. In the regularization stage, the adversarial network first updates the discriminator network to distinguish between real and generated samples, then updates the encoder of the autoencoder to fool the discriminator network. Although AAE performs well in time series, it also inherits the instability issues of GAN training.

Adversarial Autoencoders (AAEs) are applied in TSAD due to their capability to learn complex data distributions and generate samples similar to normal data but distinct from them. AAEs combine autoencoder architecture with adversarial training. Autoencoders learn low-dimensional representations of data and reconstruct it, enabling unsupervised learning.

$$\begin{aligned} \mathbf{z} &= E(\mathbf{x}), \\ \hat{\mathbf{x}} &= D(\mathbf{z}), \\ G_{AE}(\mathbf{x}) &= D(E(\mathbf{x})), \end{aligned} \tag{13}$$

where $\mathbf{z}$ represents the latent representation and $\hat{\mathbf{x}}$ is the reconstructed output. The encoder $E(\mathbf{x})$ and decoder $D(\mathbf{x})$ work together to realize the reconstruction of the time series.

Adversarial training involves adversarial networks, making generated samples harder to distinguish between real and reconstructed samples. In adversarial training, an additional adversarial network Dis, called the discriminator, is introduced. It aims to discriminate between real data samples and generated (reconstructed) samples. The generator (encoder-decoder) tries to fool the discriminator by generating samples that are close to the true data distribution. This procedure can be formulated as follows:

$$\begin{aligned} \min_{E,D}\max_{Dis} V(Dis, E, D) = \mathbb{E}_{\mathbf{x}\sim P_{\text{data}}(\mathbf{x})}[\log Dis(\mathbf{x})] \\ + \mathbb{E}_{\mathbf{z}\sim P_{\text{enc}}(\mathbf{z})}[\log(1 - Dis(D(\mathbf{z})))], \end{aligned} \tag{14}$$

where $P_{\text{data}}(\mathbf{x})$ denotes the distribution of real data, and $P_{\text{enc}}(\mathbf{z})$ denotes the distribution of latent representations.

Following training, TSAD process is performed by comparing the reconstruction error between normal and abnormal series. The underlying assumption is that anomalies will yield a significantly higher error than normal samples. The reconstruction error can be calculated using MSE:

$$L(\mathbf{x}, \hat{\mathbf{x}}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2 \tag{15}$$

SaVAE-SR [97] uses the encoder as a discriminator in the framework of AAE. The encoder is not only trained to model the approximate posterior of latent variables, but also trained to distinguishes between real samples in the training data and fake samples generated by the generator in the latent space. To alleviate the problem of anomaly data contamination encountered in many previous unsupervised anomaly detection techniques, SaVAE-SR employs a spectral residual technique to find the most significant anomalies and provide pseudo-labels for unlabeled training data.

Adversarial Autoencoders (AAEs) enhance the framework of Variational Autoencoders (VAEs) by employing adversarial training to enforce the alignment of latent variables with a target prior distribution. In certain implementations, the discriminator replaces the KL divergence term in variational inference, effectively mitigating the posterior collapse issue inherent in VAEs. Additionally, adversarial training alleviates the non-convexity challenges associated with the Evidence Lower Bound (ELBO), leading to accelerated convergence. However, the mode collapse problem prevalent in Generative Adversarial Networks (GANs) remains unresolved, and training instability frequently occurs, ultimately resulting in degraded anomaly detection performance.

IV. Libraries for Time Series Anomaly Detection

A. Applications in various fields

TSAD has broad and deep applications across various fields, including finance [18, 21], healthcare [75], real-world systems [32, 146], server monitoring [92], distributed networks, and even social platforms [14, 137]. It is crucial in pinpointing potential problems, optimizing system performance, and ultimately achieving safer, more reliable, and more efficient business operations.

Finance

Risk management remains a critical priority for banks, portfolio managers and firms involved in trading on stock exchanges. With the increase in trading volume and frequency, illegal methods such as price manipulation may cause great damage to the proper functioning and integrity of financial markets. TSAD is an important tool for protecting investors' interests and optimizing asset allocation. By monitoring trading patterns and abnormal fluctuations in stock prices, it can detect market anomalies and fraudulent transactions in a timely manner and help investors make better decisions.

Healthcare

In the healthcare sector, TSAD is employed to monitor patients' physiological parameters, facilitating the timely detection of changes in health status. Real-time monitoring of metrics including blood pressure and heart rate enables the prompt identification of acute conditions like heart attacks and hypertension, allowing healthcare professionals to intervene and provide timely treatment. Additionally, it is utilized for monitoring the operational status of medical equipment to promptly detect faults or anomalies, ensuring the smooth functioning of medical devices and patient safety.

Real-world Systems

In practical systems such as water treatment, transportation, and power systems, TSAD is widely used for equipment condition monitoring and fault detection. By analyzing time series data of equipment operating states, faults or anomalies can be promptly detected, facilitating predictive maintenance and enhancing system reliability and efficiency. Faults identified by industrial monitoring systems can be simplified as anomalous points detected from temporal data. It is essential for ensuring system security and preventing financial losses.

Server Monitoring and Distributed Networks

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

In the domain of server monitoring and distributed networks, TSAD is applied to monitor system performance, detect intrusion behaviors, and ensure system security. By analyzing time series data of server and network performance indicators, this method can promptly detect anomalies like system overload and latency. This facilitates proactive resource allocation adjustments and ensures system stability. In addition, it can detect network intrusions, such as network attacks and unauthorized access, and detect and stop malicious activities in time to ensure system security.

Social Platforms

In the field of social platforms, TSAD is used to analyze user behavior and content flow and identify anomalies such as fake accounts and malicious comments. By analyzing user interaction behavior patterns and content flow fluctuations on social platforms, it can detect anomalous behaviors and take timely measures to protect the rights and interests of users and maintain the integrity of social platforms.

Overall, TSAD, as an important data analysis technique, plays a vital role in various fields. It not only helps to identify potential problems and optimize system operation, but also improves system security, reliability and efficiency. With the continuous development of data technology and the expansion of application scenarios, TSAD will be more and more widely used in various fields, bringing more convenience and safety to people's life and work.

TABLE IV: Summary of Datasets.

Datasets	Fields	‖Training‖	‖Test‖	Dimension	Anomalys(%)
NASA					
MSL	Space	58,317	73,729	55	10.7
SMAP	Space	135,183	427,617	25	13.1
Real World System					
SWaT	Water System	224,959	224,960	51	19.1
WADI	Water System	86,401	86,401	123	3.9
μPMU	Power System	864,000	864,000	36	0.6
PMU-B	Power System	10 months	1 months	38	-
PMU-C	Power System	10 months	1 months	132	-
METR-LA	Traffic	-	-	207	-
Numenta Anomaly Benchmark					
NAB Art	Artificial	24,192	6,048	1	1.0
NAB AdEx	Transactions	7,965	1,992	1	1.0
NAB AWS	Cloud	67,644	16,911	1	0.9
NAB Traf	Traffic	15,662	3,916	1	1.0
NAB Tweets	Twitter	158,511	39,628	1	1.0
Server Monitoring					
SMD	Server	708,405	708,420	38	4.2
MBD	Server	1,000	1,000	130	0.2
MMS	Server	1,000	1,000	310	0.1
ASD	Server	302,400	216,000	19	0.3
Distributed Networks					
KDDCUP99	Networks	562,387	494,021	34	80.3
DND	Networks	8 days	2 days	38	0.5
MSDS	Networks	146430	146430	10	5.4
Yahoo					
Yahoo S5 A1	Server	94,866	23,717	1	0.2
Yahoo S5 A2	Synthetic	142,100	35,525	1	0.6
Yahoo S5 A3	Synthetic	168,000	42,000	1	0.6
Yahoo S5 A4	Synthetic	168,000	42,000	1	0.5
Electrocardiogram (ECG)					
MIT-BIH	ECG	31,221,760	-	2	10.5
Motion					
CMU	Motion	10,309	-	4	35.8

B. Datasets

Table IV summarizes common datasets for time series anomaly detection tasks, presenting information on dataset scale, dimensions, and anomaly proportions. Categorizing these datasets based on their respective domains yields four classifications: **1) NASA:** Various sensor data from NASA's Mars probes [68], widely utilized. Such datasets are real and validated, and often have a high degree of complexity and diversity. **2) Real World System:** Temporal data generated in diverse real-world systems such as water treatment [52], power systems[122], and traffic systems[84]. These datasets may vary widely in terms of data sources and uses, but what they have in common is that they reflect the operation of different systems in the real world. These datasets may be characterized by their dynamism and need for real-time availability. **3) Server Monitoring and Distributed Networks:** Real-time data from server monitoring [61, 89, 123] and distributed networks[26, 86]. This type of data tends to be high-dimensional, high-frequency and requires fast response times for anomaly detection algorithms. **4) Other Datasets:** Includes datasets provided by benchmarks [81], data from domains like healthcare [104] and social platforms, and synthetic data [50, 69, 102, 115]. The quality of such datasets tends to vary, posing additional challenges to anomaly detection algorithms.

C. Dataset Types and Challenges in Datasets

Additionally, we reviewed recent research, analyzing the intrinsic reasons for variations in performance across different datasets and application scenarios. Based on this analysis, we recommend suitable methods for different types of datasets and specific scenario challenges:

1) Dataset Types

Univariate dataset: Comprising time series data with only one variable (or dimension), these datasets are often simpler in nature [50, 81]. Univariate time series data represents the most basic form of time series. Characteristically, it often exhibits periodic patterns, seasonal variations, trends, and residual components. General generative methods can effectively perform single-dimensional TSAD tasks. Considering the low complexity of the task, computationally efficient methods are often preferred. Effective methods on such datasets include ADDMM [115], RDSSM [90], DCT-GAN [88].

Multivariate dataset: Involving time series data with multiple dimensions, such as data from multiple sensors, servers, or graph-structured data. This type of data often possesses the ability to represent both time and space. Treating them as multiple single-dimensional time series often fails to achieve satisfactory results, as considering only temporal dependencies is far from sufficient. High-dimensional multivariate time series data necessitates our attention to the correlations between variables, which is crucial for understanding the complex interactions and relationships between variables as they evolve over time. Multi-dimensional time series encompass numerous periodic and seasonal time features, coupled with the interplay between multiple variables, resulting in complex dynamic patterns that pose challenges for TSAD. The emergence of deep learning techniques such as Graph Neural Networks (GNNs) and Transformers has, to some extent, addressed this issue. GNNs excel in handling graph-structured data, enabling them to capture intricate relationships between variables. Through interactions between nodes and edges, GNNs can uncover the temporal evolution patterns of node attributes, understanding the spatial semantics of time series. Transformers employ bidirectional self-attention mechanisms to capture long-range dependencies and variable correlations respectively, making them suitable for processing multi-dimensional time series data. Furthermore, certain generative deep learning architectures, such as Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs), have distinguished themselves in variable correlation modeling due to their powerful learning capabilities. Prominent methods for these datasets include InterFusion [89], FGANomaly [44], TopoMAD [61], DVGCRN [26], MT-RVAE [129], GANF [36].

2) Challenges in Datasets

Contaminated dataset: Typically, datasets are divided into training and testing sets. The training set generally contains no anomalies or only a negligible number of anomalies, while the test set includes both normal and anomalous data. However, in the case of a "Contaminated Dataset," the training set includes a significant number of anomalies, which poses challenges for standard training

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

TABLE V: Summary of representative generative methods.

Model	Su/Un	Dataset	Venue	Code
Markov				
HMAD [55]	Su	-	ICML'2015	-
AHMMAS [16]	Un	Google, Microsoft, Intel, Apple, ARM, BARCLAYS, Vodafone	TNNLS'2015	-
SMS-VAR [102]	Un	Synthetic data and Real data	KDD'2016	Link
sGMRFmix [69]	Un	Synthetic data and Real data from the oil industry	ICDE'2016	Link
ADDMM [115]	Un	Synthetic data, ECG	IS'2017	-
VAE				
OmniAnomaly [123]	Un	SMD, MSL, SMAP	KDD'2019	Link
SISVAE [87]	Un	Yahoo S5 A1, Yahoo S5 A2, Yahoo S5 A3, Yahoo S5 A4, μPMU	TNNLS'2020	Link
TopoMAD [61]	Un	MBD, MMS	TNNLS'2020	Link
InterFusion [89]	Un	SWaT, WADI, SMD, ASD	KDD'2021	Link
RDSSM [90]	Un	MSL, SMAP, Yahoo S5 A1, Yahoo S5 A2, Yahoo S5 A3, Yahoo S5 A4	TKDE'2022	-
DVGCRN [26]	Un	DND, SMD, MSL, SMAP	ICML'2022	Link
SLA-VAE [67]	Semi	Cloud Server A, Cloud Server B	WWW'2022	Link
MT-RVAE [129]	Un	SAT, SKAB, NAB	Measurement'2022	-
GAN				
BeatGAN [149]	Un	MIT-BIH, CMU	IJCAI'2019	Link
MADGAN [86]	Un	SWaT, WADI, KDDCUP99	ICANN'2019	Link
TadGAN [50]	Un	MSL, SMAP, Yahoo S5 A1, Yahoo S5 A2, Yahoo S5 A3, Yahoo S5 A4, NAB Art, NAB AdEx, NAB AWS, NAB Traf, NAB Tweets	BigData'2020	Link
DAEMON [22]	Un	SMD, MSL, SMAP, SWaT	ICDE'2021	-
FGANomaly [44]	Un	MSL, SMAP, SWaT, WADI	TKDE'2022	Link
DCT-GAN [88]	Un	NAB, SWAT, WADI	TKDE'2023	-
Normalizing Flow				
GANF [36]	Un	PMU-B, PMU-C, SWaT, METR-LA	ICLR'2022	Link
Diffusion Model				
DiffusionAE [110]	Un	NeurIPS-TS, SWaT, WADI	ICDM(W)'2023	Link
DiffAD [138]	Un	MSL, SMAP, SWaT, PSM, SMD	KDD'2023	Link
Adversarial Autoencoder				
SaVAE-SR [97]	Un	KPI, Yahoo S5 A1, Yahoo S5 A2, Yahoo S5 A3, Yahoo S5 A4	Neurocomputing'2021	Link

methods. This contamination can hinder the model's ability to learn the distribution of normal data, thereby affecting its performance. For example, on SMD and ASD, there are anomalies in the training data. It is often difficult to achieve good results without using strategies such as pre-filtering, which overfits the anomalous patterns in the training data with learned flexible inter-embeddings of the metrics. In the Secure Water Treatment datasets SWaT and WADI, the original dataset encompasses 7 days of normal operation and 4 days under various attack scenarios. By partitioning the dataset according to a specific principle (e.g., a time-based split), we can ensure that the training subset contains only normal patterns and is free from anomalies. Deep generative models such as GANs and VAEs can capture the distribution of the training set, learn the latent representation of normal data, and thereby generate samples that follow this distribution. Since they have never encountered any anomalous samples, it is difficult for them to produce anomalous samples. In this scenario, the reconstruction of normal samples will achieve relatively high accuracy, while anomalous samples cannot be reconstructed well, ultimately making them easier to detect. To address these challenges, some methods employ pseudo-labels to filter potential anomalous samples before training the detector, thereby capturing the distribution of normal data as accurately as possible. Additionally, other methods design novel training objectives that focus the loss function more on credible normal data while neglecting anomalies. Effective methods for such datasets include DAEMON [22], FGANomaly [44], DCT-GAN [88], InterFusion [89] and MT-RVAE [129].

Online-offline industrial dataset: In industrial scenarios, models need to adapt and adjust to new incoming data. Offline industrial detection refers to analyzing and identifying anomalous patterns in historical data, while online industrial detection involves real-time analysis and monitoring of the industrial system's state. In industrial system environments, as time progresses and data characteristics change, new data distributions continuously emerge, leading to frequent concept drift phenomena. However, existing algorithms typically can only capture historical data distributions, failing to meet the requirements of online detection. Furthermore, online anomaly detection algorithms need to keep inference latency within a low range. Detecting anomalies earlier minimizes the losses caused by system abnormalities, but this places higher demands on the algorithm's time complexity. To cope with them, some methods are designed with specialized Offline Training with Active Online Detection

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

strategies. Active learning methods are used to continuously optimize the online detection model with the aim of learning and updating the model from a small amount of new data. Methods demonstrating good performance on these datasets include OmniAnomaly [123], TopoMAD [61], InterFusion [89], SLA-VAE [67].

D. Threshold selection for time series anomaly detection

In the early stages of TSAD, little attention was paid to the design of anomaly thresholds, and threshold selection based on expert experience became the mainstream approach [55]. Experts establish a threshold for each observed feature based on industry reference values, and results exceeding the threshold are considered anomalous. However, this requires sufficient priori knowledge. Various methods have been proposed to improve thresholding.

1) Static Threshold Methods

AHMMAS [16] sets the threshold at the 99% cumulative distribution cutoff, flagging the top and bottom 0.5% of feature values as anomalies, considering them the most extreme and rare. [45] uses an adaptive threshold where, for each cube position, the lowest likelihood value among training samples is identified and multiplied by a factor k (typically a constant greater than 1) to generate the final threshold. BeatGAN [149] incorporates the anomaly proportion of the dataset into its thresholding strategy by proposing a naive percentile-based thresholding method for anomaly scores.

Despite these advancements, such approaches often fail to provide robust discrimination between normal and anomalous samples. To address this, statistical methods have been developed to refine threshold selection. For instance, TopoMAD [61] introduces a thresholding method based on the assumption that anomaly scores of normal data lie in high-density regions, while those of anomalous data are located in low-density regions, with a significant distance separating the two. Other approaches, such as SISVAE [87] and InterFusion [89], aim to select the threshold corresponding to the highest F1 score achievable by the model. These methods evaluate each anomaly score as a potential threshold, compute the resulting F1 score, and choose the score that maximizes it. Although effective, this approach is computationally demanding in practice. Moreover, metrics like Average Precision (AP) and the Area Under the ROC Curve (AUROC), which do not rely on threshold optimization, have shown to be robust alternatives.

2) Dynamic Threshold Methods

While static thresholds remain fixed, they do not adapt to changes in the variance of time series data, which is often non-stationary in real-world applications. For example, in dynamic operating environments with diverse working conditions, static threshold methods struggle to generalize and apply a uniform limit to faults of varying nature. Dynamic thresholding methods have emerged to address this issue.

TadGAN employs a sliding window approach to compute thresholds dynamically, with the window size determining the number of historical anomaly scores used for threshold calculation. A common formulation for a dynamic threshold is the Non-Parametric Dynamic Threshold (NDT) [68], expressed as:

$$\begin{cases} MAX_t = \mu_i + Z \cdot \delta_i^2 \\ MIN_t = \mu_i - Z \cdot \delta_i^2 \\ i = \lfloor \frac{t}{n} \rfloor \end{cases} \tag{16}$$

where:

- Z is a manually set hyperparameter.
- n represents the size of the sliding window.
- μ_i is the mean value of the error vector V in the i-th sliding window.
- δ_i^2 is the variance of the error vector V in the i-th sliding window.
- $\lfloor \cdot \rfloor$ denotes the floor (round-down) operation.

OmniAnomaly uses an adapted Peak-Over-Threshold (POT) methodology combined with sliding window and extreme value theory for automatic threshold selection. Unlike traditional POT, which focuses on the extremely high values of the distribution, SLA-VAE identifies anomalies at the extremely low values of the distribution. SLA-VAE further refines the threshold tuning by applying POT to select an initial threshold, and then performs a grid search in a defined grid space based on the initial threshold to find the threshold that maximizes the F1 score.

E. Metrics

When evaluating the performance of TSAD models, common evaluation metrics include Precision (P), Recall (R), F1 Score, and Area Under Curve (AUC) [49, 79]. These metrics are widely used to quantify the effectiveness of anomaly detection. The metrics and their variants are listed below:

- **Precision (P)** measures the proportion of correctly identified anomalies among all samples labeled as anomalies by the model.

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

- **Recall (R)** quantifies the percentage of actual anomalies accurately detected by the model.

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

- **F1 Score (F1)** is the harmonic mean of P and R. These three metrics are often used together to comprehensively evaluate the model's performance, especially when there is a need to balance P and R.

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{19}$$

- **F1 with Point-Adjustment** [6] is a common evaluation strategy in TSAD. Under this protocol, if any point within a contiguous anomaly segment is detected, the entire segment is deemed to be correctly identified. Many researchers believe that this method tends to overestimate the effectiveness of anomaly detection to some extent. In many studies, the F1 scores obtained using the point adjustment strategy are significantly higher than those obtained without it.
- **F1 with PA%K** [76] is proposed as an improvement on point adjustment, which mitigates the effect of overestimation of F1 with Point-Adjustment and the possibility of underestimation of F1. Unlike F1 with Point-Adjustment, all anomalies in an anomalous segment are considered to be correctly detected only if the ratio of the number of correctly detected anomalies in a consecutive anomalous segment to its length exceeds the PA%K threshold.
- **Area Under the Curve (AUC)** is a common metric used to evaluate binary classification models and can also be applied to assess the performance of TSAD models. The higher the AUC value, the better the performance of the model in distinguishing between normal and abnormal samples. The AUC value 1 indicates perfect discrimination between anomalies and normal samples.

V. Future Directions

A. Data Source

Challenge. Most methods discussed in the paper operate on preprocessed data, yet real-world data is often incomplete and of low quality. Issues such as different frequencies in multivariate data and the presence of missing data and labels pose significant challenges. Varied frequencies may result in different time intervals across dimensions in time series, while missing data and labels can impact the accuracy of TSAD. Applying existing methods directly to such data might not yield desirable results, as current models often demand high-quality data.

Opportunity. Some efforts have explored standardized data preprocessing methods, including advanced interpolation techniques and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

intelligent approaches for handling missing values when dealing with different frequencies and missing data. Additionally, methods have experimented with pseudo-labels [44], data augmentation [134], and generating high-quality [130] data. However, there remains a lack of universal approaches to efficiently enhance data quality. Thus, we propose two prospective research directions: **1)** Extend existing data processing methods to formulate a rational and universal time series data processing pattern, aiming to maintain data quality and enhance model robustness. **2)** Within the realm of existing GAN-based methods, focus on strategies to generate higher-quality time series data.

Self-supervised learning (SSL) is an essential technique to explore, especially when dealing with the scarcity of labeled data. With the limited availability of annotated datasets, self-supervised pretraining has gained significant attention and has been extensively researched in natural language modeling [112], sequence recommendation [131, 142], and computer vision [60, 139]. Self-supervised learning notably reduces the dependency on high-quality labeled data across various tasks while enhancing the representational power of time series data. Notably, recent studies have begun to focus on self-supervised pretraining methods specifically tailored for time series data. The two most prominent paradigms in this context are contrastive learning and masked time series modeling. Contrastive learning involves learning representations of time series data by contrasting positive and negative samples. Some approaches [147] have introduced the use of multiple invariances to generate a diverse set of augmentations.TimesURL [95] highlights that improper construction of positive and negative pairs can introduce undesirable inductive biases, which may neither preserve temporal characteristics nor provide sufficient discriminative features. To address this, they introduced a frequency-based augmentation method that preserves time-related features and employed temporal reconstruction as a joint optimization objective in contrastive learning, aiming to capture both segment-level and instance-level information. On the other hand, masked time series modeling [42] enables the model to learn the structure and dependencies within time series data by masking parts of the input sequence and predicting the masked portions. This approach allows the model to develop a deeper understanding of temporal patterns and relationships.

B. Integration of Multimodal Data

Challenge. While significant progress has been made in single-modal TSAD tasks, real-world applications, including industrial production, IOT and healthcare, often involve multimodal time series data. Some methods have shown promising results for multimodal data from different sensors, but existing work in TSAD has yet to extensively cover richer modalities like audio, video, trajectory graphs, and text streams. Therefore, there is substantial exploration space for the detection of anomalies in multimodal temporal data.

Opportunity. For multimodal temporal data, exploring advanced fusion methods becomes crucial, with a particular focus on understanding the correlation of temporal information in the multimodal feature space. An In-depth investigation into the interactions between multimodal features aids in a more comprehensive understanding of information from different modalities. Bai et al. [7] proposed a Prompt-based Distribution Alignment method for the unsupervised Domain Adaptation problem. It uses two prompt tuning modules to realize cross-domain alignment, which enhances the model's learning capability within the target modality. Therefore, I suggest future work should focus on the following two directions: **1)** Modality alignment is a key issue in multimodal data processing. Future research may concentrate on improving data alignment across different modalities, especially considering different time scales, sampling rates, and data types. **2)** Designing more complex model structures, effective training strategies, and loss functions adaptable to multimodal data should be prioritized to address this gap.

C. Enhancing Interpretability and Trustworthiness

Challenge. While existing generative methods have demonstrated effectiveness in the task of TSAD, a prevalent challenge is the limited interpretability [151] of these methods. This becomes particularly problematic in sensitive domains such as healthcare or finance, where it is vital to understand how these methods work. Despite some progress in generative methods for TSAD, the interpretability of these approaches remains a notable challenge, especially in explaining the rationale behind model outputs.

Opportunity. Current methods often provide explanations based on reconstruction. For instance, OmniAnomaly offers anomaly explanations based on reconstruction probabilities, avoiding rule-based or expert knowledge-based interpretations to reduce biases and errors. BeatGAN [149] locates the time points of anomalies by comparing residuals between input and reconstructed heartbeat signals, offering visualizations and attention guidance. However, such explanatory results often exhibit incompleteness or uncertainty.

In time series analysis, causal inference is a crucial approach. Identifying potential lagged causal processes in sequence data is essential for understanding temporal dynamics and downstream reasoning. Based on this theory, researchers introduced a method called CaRiNG [28], which learns causal representations of non-reversible generative time series data with identifiability guarantees. This method utilizes temporal context to recover missing latent information and applies theoretical conditions to guide the training process. Additionally, time series decomposition is another promising direction. Existing decomposition methods are categorized into three types [132]: Seasonal-Trend Decomposition [35], Basis Expansion [105], and Matrix Factorization [144]. Some studies suggest that a better approach is to decouple information in the time and sample dimensions, allowing each representation to be learned more fully under its specific objectives. TimeDRL [19] proposes an unsupervised pre-training method based on decomposition learning, which decouples representations in the time and sample dimensions. This generates representation vectors that better support various time series tasks while enhancing interpretability.

D. LLMs for Time Series

Challenge. While foundational models have achieved significant success in domains such as natural language processing (NLP) and computer vision [12, 78, 114], the development of foundational models for time series prediction has lagged behind. Early efforts in this area attempted to leverage the network structures of large models to retrain a foundational model specifically for time series data using vast amounts of time series data across various domains. These methods often employed Transformer-based unified sequence modeling approaches to enable cross-domain learning, thereby aiming to achieve robust time series representations. Although some progress has been made in developing a unified time series foundational model [10, 48, 56, 74, 96, 100, 150], there are still many challenges. **1)** There are notable differences between the different domains, especially in terms of dimensionality, frequency and modalities. These differences present obstacles to joint training efforts. **2)** Pretraining a time series-specific foundational model often requires an enormous amount of high-quality time series data, which is difficult to obtain. **3)** Models trained on low-quality, weakly semantic, and hard-to-predict data tend to lack the general understanding necessary for effective time series analysis.

Opportunity. Recent efforts have addressed these challenges by adopting patch-based approaches for modeling time series. Researchers [136] concatenated multivariate time series, flattening them into a single sequence and introducing mask characters to denote positions for prediction. Structurally, they employed a multi-granularity patch modeling approach, accommodating sequences of different frequencies. [10] proposed a training methodology more aligned with conventional NLP large models. They introduced the next patch prediction task, continuously predicting the next patch with MSE used to compute loss for each patch prediction. Therefore, based

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

on their theoretical findings, it is very promising and feasible to explore a more refined approach to LLM modeling and training. Another representative work in this area is Chronos, which operates on the premise that, despite the differences between natural language and time series data, both are inherently sequential. By scaling and quantizing time series data, Chronos [4] converts continuous time series into discrete tokens, allowing the application of language models without significant architectural modifications. It then employs a cross-entropy loss to train existing Transformer-based language model architectures to handle these tokenized time series. Similarly, TIME-LLM [74] introduces the concept of text prototypes to reprogram the input time series, aligning it with the frozen LLM. To enhance the reasoning capabilities of LLMs for time series data, TIME-LLM introduces the Prompt-as-Prefix (PaP) approach, which enriches the input context and guides the reprogramming of input patches. This method has demonstrated strong performance in both few-shot and zero-shot learning scenarios. In the context of anomaly detection, two primary LLM-based approaches have emerged for time series: **1)** PROMPTER: This approach converts the anomaly detection task into a prompt that is fed into the LLM, which then directly provides the answer. **2)** DETECTOR: In this approach, the LLM first predicts the time series, and anomalies are identified by comparing the predicted values to the actual ones. The SIGLLM [3] framework, for example, employs GPT-3.5-turbo to address TSAD, providing initial validation of the effectiveness of LLMs in this domain. To adapt time series data for LLM input, SIGLLM converts the series into numerical values, with a focus on retaining as much of the original time series information as possible using the shortest input length. Additionally, exploring alternatives to the Transformer architecture is another direction worth noting. Mamba [57, 133] is one of the most discussed models in this context and is even considered by some in the industry as a potential replacement for Transformers. Mamba is a State Space Model-based architecture, reminiscent of RNNs. Compared to Transformers, Mamba exhibits linear time complexity concerning sequence length during both training and inference stages, leading to significantly higher computational efficiency. Some studies [2, 9, 108] have explored the application of the Mamba model in time series prediction tasks, demonstrating its effectiveness.

E. Online Learning and Adaptive Approaches

Challenge. In practical applications, the demand for models to dynamically adapt to changing data distributions and patterns is particularly pronounced, especially in industrial settings and network service monitoring. When faced with previously unseen data distributions beyond what the model has learned, the model's generalization ability can sharply degrade, leading to a decline in anomaly detection performance. Ensuring stability and adaptability in online learning scenarios proves to be an exceptionally challenging task, especially for generative models that need to continuously learn the distribution and patterns of normal data. The degradation of performance in detecting anomalies due to the dynamic nature of data poses a significant challenge.

Opportunity. The challenge presents a substantial opportunity for research in adaptive methods for TSAD, particularly focusing on approaches grounded in Generative Adversarial Networks (GANs). Investigating GAN-based methods for adapting to changing data distributions in online learning scenarios holds promise in enhancing stability and adaptability. By leveraging the adversarial training capabilities of GANs, these methods can potentially facilitate more effective adaptation to evolving normal data distributions and patterns. This opportunity not only addresses the challenge of maintaining performance in dynamic environments but also opens avenues for advancing the field's understanding of how generative models can adapt to real-world changes.

VI. Conclusion

In this survey, we systematically review generative methods for TSAD. Unlike previous reviews that focus on deep time series analysis methods, this paper provides a comprehensive investigation and analysis of existing generative methods for TSAD. Generative anomaly detection for time series is more than just applying off-the-shelf generative models. The core challenge is the fundamental mismatch between time series characteristics (temporal dependency, dynamic patterns, multi-scale nature) and the objective of generative models. Complex cross-dimensional dependencies in multi-variate sequences are also often overlooked by generative models. Detection effectiveness heavily relies on clean training data. Anomalous segments can distort the model's learning of normal patterns (long-term dependencies, periodicity, trends), potentially leading the model to mistake anomalies as normal. In models like VAE or Diffusion, anomalies may reside in atypical locations or form separate clusters in latent space, yet still be considered part of the normal representation. Additionally, the anomaly scoring mechanism must align with temporal characteristics, as reconstruction error or negative log-likelihood may be overly sensitive to temporal misalignment. An appropriate anomaly scoring mechanism can significantly enhance anomaly detection performance.

References

[1] Carlo Adornetto and Gianluigi Greco. Gidnets: generative neural networks for solving inverse design problems via latent space exploration. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3404–3413, 2023.

[2] Md Atik Ahamed and Qiang Cheng. Timemachine: A time series is worth 4 mambas for long-term forecasting. *arXiv preprint arXiv:2403.09898*, 2024.

[3] Sarah Alnegheimish, Linh Nguyen, Laure Berti-Equille, and Kalyan Veeramachaneni. Large language models can be zero-shot anomaly detectors for time series? *arXiv preprint arXiv:2405.14755*, 2024.

[4] Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

[5] Nazanin Asadi, Abdolreza Mirzaei, and Ehsan Haghshenas. Creating discriminative models for time series classification and clustering by hmm ensembles. *IEEE transactions on cybernetics*, 46(12):2899–2910, 2015.

[6] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. Usad: Unsupervised anomaly detection on multivariate time series. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3395–3404, 2020.

[7] Shuanghao Bai, Min Zhang, Wanqi Zhou, Siteng Huang, Zhirong Luan, Donglin Wang, and Badong Chen. Prompt-based distribution alignment for unsupervised domain adaptation. *arXiv preprint arXiv:2312.09553*, 2023.

[8] Md Abul Bashar and Richi Nayak. Tanogan: Time series anomaly detection with generative adversarial networks. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1778–1785. IEEE, 2020.

[9] Ali Behrouz, Michele Santacatterina, and Ramin Zabih. Mambamixer: Efficient selective state space models with dual token and channel selection. *arXiv preprint arXiv:2403.19888*, 2024.

[10] Yuxuan Bian, Xuan Ju, Jiangtong Li, Zhijian Xu, Dawei Cheng, and Qiang Xu. Multi-patch prediction: Adapting llms for time series representation learning. *arXiv preprint arXiv:2402.04852*, 2024.

[11] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.

[12] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the

opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[13] Mohammad Braei and Sebastian Wagner. Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433*, 2020.

[14] Cody Buntain, Richard Bonneau, Jonathan Nagler, and Joshua A Tucker. Youtube recommendations and effects on sharing across online social platforms. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–26, 2021.

[15] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

[16] Yi Cao, Yuhua Li, Sonya Coleman, Ammar Belatreche, and Thomas Martin McGinnity. Adaptive hidden markov model with anomaly states for price manipulation detection. *IEEE transactions on neural networks and learning systems*, 26(2):318–330, 2014.

[17] Cristian I Challu, Peihong Jiang, Ying Nian Wu, and Laurent Callot. Deep generative model with hierarchical latent factors for time series anomaly detection. In *International Conference on Artificial Intelligence and Statistics*, pages 1643–1654. PMLR, 2022.

[18] Ngai Hang Chan. *Time series: applications to finance*. John Wiley & Sons, 2004.

[19] Ching Chang, Chiao-Tung Chan, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. Timedrl: Disentangled representation learning for multivariate time-series. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 625–638. IEEE, 2024.

[20] Zhengping Che, Sanjay Purushotham, Guangyu Li, Bo Jiang, and Yan Liu. Hierarchical deep generative models for multi-rate multivariate time series. In *International Conference on Machine Learning*, pages 784–793. PMLR, 2018.

[21] Cathy WS Chen, Feng-Chi Liu, and Mike KP So. A review of threshold time series models in finance. *Statistics and its Interface*, 4(2):167–181, 2011.

[22] Xuanhao Chen, Liwei Deng, Feiteng Huang, Chengwei Zhang, Zongquan Zhang, Yan Zhao, and Kai Zheng. Daemon: Unsupervised anomaly detection and interpretation for multivariate time series. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2225–2230. IEEE, 2021.

[23] Yongliang Chen, Qingying Zhao, and Laijun Lu. Combining the outputs of various k-nearest neighbor anomaly detectors to form a robust ensemble model for high-dimensional geochemical anomaly detection. *Journal of Geochemical exploration*, 231:106875, 2021.

[24] Zhipeng Chen, Zhang Peng, Xueqiang Zou, and Haoqi Sun. Deep learning based anomaly detection for muti-dimensional time series: A survey. In *China Cyber Security Annual Conference*, pages 71–92. Springer Nature Singapore Singapore, 2021.

[25] Shaowei Chen, Fangda Xu, Pengfei Wen, Shuaiwen Feng, and Shuai Zhao. A multivariate time series anomaly detection method based on generative model. In *2022 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pages 137–144. IEEE, 2022.

[26] Wenchao Chen, Long Tian, Bo Chen, Liang Dai, Zhibin Duan, and Mingyuan Zhou. Deep variational graph convolutional recurrent network for multivariate time series anomaly detection. In *International Conference on Machine Learning*, pages 3621–3633. PMLR, 2022.

[27] Ningjiang Chen, Huan Tu, Xiaoyan Duan, Liangqing Hu, and Chengxiang Guo. Semisupervised anomaly detection of multivariate time series based on a variational autoencoder. *Applied Intelligence*, 53(5):6074–6098, 2023.

[28] Guangyi Chen, Yifan Shen, Zhenhao Chen, Xiangchen Song, Yuewen Sun, Weiran Yao, Xiao Liu, and Kun Zhang. Caring: Learning temporal causal representation under non-invertible generation process. *arXiv preprint arXiv:2401.14535*, 2024.

[29] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. Financial time series forecasting with multi-modality graph neural network. *Pattern Recognition*, 121:108218, 2022.

[30] Yeji Choi, Hyunki Lim, Heeseung Choi, and Ig-Jae Kim. Gan-based anomaly detection and localization of multivariate time series data for power plant. In *2020 IEEE international conference on big data and smart computing (BigComp)*, pages 71–74. IEEE, 2020.

[31] Kai Lai Chung. Markov chains. *Springer-Verlag, New York*, 1967.

[32] Andrew A Cook, Göksel Mısırlı, and Zhong Fan. Anomaly detection for iot time-series data: A survey. *IEEE Internet of Things Journal*, 7(7):6481–6494, 2019.

[33] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[34] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[35] Estela Bee Dagum and Silvia Bianconcini. *Seasonal adjustment methods and real time trend-cycle estimation*. Springer, 2016.

[36] Enyan Dai and Jie Chen. Graph-augmented normalizing flows for anomaly detection of multiple time series. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022.

[37] Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu C Aggarwal, and Mahsa Salehi. Deep learning for time series anomaly detection: A survey. *arXiv preprint arXiv:2211.05244*, 2022.

[38] Jan G De Gooijer and Rob J Hyndman. 25 years of time series forecasting. *International journal of forecasting*, 22(3):443–473, 2006.

[39] Ruizhi Deng, Bo Chang, Marcus A Brubaker, Greg Mori, and Andreas Lehrmann. Modeling continuous stochastic processes with dynamic normalizing flows. *Advances in Neural Information Processing Systems*, 33:7805–7815, 2020.

[40] Nan Ding, Huanbo Gao, Hongyu Bu, and Haoxuan Ma. Radm: Real-time anomaly detection in multivariate time series based on bayesian network. In *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 129–134. IEEE, 2018.

[41] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[42] Jiaxiang Dong, Haixu Wu, Haoran Zhang, Li Zhang, Jianmin Wang, and Mingsheng Long. Simmtm: A simple pre-training framework for masked time-series modeling. *Advances in Neural Information Processing Systems*, 36, 2024.

[43] Werner Dreyer, Angelika Kotz Dittrich, and Duri Schmidt. Research perspectives for time series management systems. *ACM SIGMOD Record*, 23(1):10–15, 1994.

[44] Bowen Du, Xuanxuan Sun, Junchen Ye, Ke Cheng, Jingyuan Wang, and Leilei Sun. Gan-based anomaly detection for multivariate time series using polluted training set. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[45] Elise Epaillard and Nizar Bouguila. Proportional data modeling with hidden markov models based on generalized dirichlet and beta-liouville mixtures applied to anomaly detection in public areas. *Pattern Recognition*, 55:125–136, 2016.

[46] Cynthia Freeman, Jonathan Merriman, Ian Beaver, and Abdullah Mueen. Experimental comparison and survey of twelve time series anomaly detection algorithms. *Journal of Artificial Intelligence Research*, 72:849–899, 2021.

[47] Keke Gao, Wenbin Feng, Xia Zhao, Chongchong Yu, Weijun Su, Yuqing Niu, and Lu Han. Anomaly detection for time series with difference rate sample entropy and generative adversarial networks. *Complexity*, 2021:1–13, 2021.

[48] Shanghua Gao, Teddy Koker, Owen Queen, Thomas Hartvigsen, Theodoros Tsiligkaridis, and Marinka Zitnik. Units: Building a unified time series model. *arXiv preprint arXiv:2403.00131*, 2024.

[49] Astha Garg, Wenyu Zhang, Jules Samaran, Ramasamy Savitha, and Chuan-Sheng Foo. An evaluation of anomaly detection and

diagnosis in multivariate time series. *IEEE Transactions on Neural Networks and Learning Systems*, 33(6):2508–2517, 2021.

[50] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Tadgan: Time series anomaly detection using generative adversarial networks. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 33–43. IEEE, 2020.

[51] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International conference on machine learning*, pages 881–889. PMLR, 2015.

[52] Jonathan Goh, Sridhar Adepu, Khurum Nazir Junejo, and Aditya Mathur. A dataset to support research in the design of secure water treatment systems. In *Critical Information Infrastructures Security: 11th International Conference, CRITIS 2016, Paris, France, October 10–12, 2016, Revised Selected Papers 11*, pages 88–99. Springer, 2017.

[53] Gastón García González, Pedro Casas, Alicia Fenández, and Gabriel Gómez. Network anomaly detection with net-gan, a generative adversarial network for analysis of multivariate time-series. In *Proceedings of the SIGCOMM'20 Poster and Demo Sessions*, pages 62–64. 2020.

[54] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[55] Nico Görnitz, Mikio Braun, and Marius Kloft. Hidden markov anomaly detection. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 1833–1842, 2015.

[56] Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

[57] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

[58] Yueyan Gu and Farrokh Jazizadeh. Degan: Time series anomaly detection using generative adversarial network discriminators and density estimation. *arXiv preprint arXiv:2210.02449*, 2022.

[59] Dibyajyoti Guha, Rajdeep Chatterjee, and Biplab Sikdar. Anomaly detection using lstm-based variational autoencoder in unsupervised data in power grid. *IEEE Systems Journal*, 17(3):4313–4323, 2023.

[60] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[61] Zilong He, Pengfei Chen, Xiaoyun Li, Yongfeng Wang, Guangba Yu, Cailin Chen, Xinrui Li, and Zibin Zheng. A spatiotemporal deep learning approach for unsupervised anomaly detection in cloud systems. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[62] Geoffrey E Hinton. Boltzmann machine. *Scholarpedia*, 2(5):1668, 2007.

[63] Thi Kieu Khanh Ho, Ali Karami, and Narges Armanfard. Graph-based time-series anomaly detection: A survey. *arXiv preprint arXiv:2302.00058*, 2023.

[64] Maximilian Hoh, Alfred Schöttl, Henry Schaub, and Franz Wenninger. A generative model for anomaly detection in time series data. *Procedia Computer Science*, 200:629–637, 2022.

[65] Tian Huang, Yan Zhu, Qiannan Zhang, Yongxin Zhu, Dongyang Wang, Meikang Qiu, and Lei Liu. An lof-based adaptive anomaly detection scheme for cloud computing. In *2013 IEEE 37th Annual computer software and applications conference workshops*, pages 206–211. IEEE, 2013.

[66] Jiajia Huang, Ernest Kurniawan, and Sumei Sun. Cellular kpi anomaly detection with gan and time series decomposition. In *ICC 2022-IEEE International Conference on Communications*, pages 4074–4079. IEEE, 2022.

[67] Tao Huang, Pengfei Chen, and Ruipeng Li. A semi-supervised vae based active anomaly detection framework in multivariate time series for online systems. In *Proceedings of the ACM Web Conference 2022*, pages 1797–1806, 2022.

[68] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 387–395, 2018.

[69] Tsuyoshi Idé, Ankush Khandelwal, and Jayant Kalagnanam. Sparse gaussian markov random field mixtures for anomaly detection. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, pages 955–960. IEEE, 2016.

[70] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data mining and knowledge discovery*, 33(4):917–963, 2019.

[71] Søren Kejser Jensen, Torben Bach Pedersen, and Christian Thomsen. Time series management systems: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 29(11):2581–2600, 2017.

[72] Guoqian Jiang, Ping Xie, Haibo He, and Jun Yan. Wind turbine fault detection using a denoising autoencoder with temporal information. *IEEE/Asme transactions on mechatronics*, 23(1):89–100, 2017.

[73] Junji Jiang, Likang Wu, Hongke Zhao, Hengshu Zhu, and Wei Zhang. Forecasting movements of stock time series based on hidden state guided deep learning approach. *Information Processing & Management*, 60(3):103328, 2023.

[74] Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

[75] Shruti Kaushik, Abhinav Choudhury, Pankaj Kumar Sheron, Nataraj Dasgupta, Sayee Natarajan, Larry A Pickett, and Varun Dutt. Ai in healthcare: time-series forecasting using statistical, neural, and ensemble architectures. *Frontiers in big data*, 3:4, 2020.

[76] Siwon Kim, Kukjin Choi, Hyun-Soo Choi, Byunghan Lee, and Sungroh Yoon. Towards a rigorous evaluation of time-series anomaly detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7194–7201, 2022.

[77] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019.

[78] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[79] György Kovács, Gheorghe Sebestyen, and Anca Hangan. Evaluation metrics for anomaly detection algorithms in time-series. *Acta Universitatis Sapientiae, Informatica*, 11(2):113–130, 2019.

[80] Takashi Kuremoto, Shinsuke Kimura, Kunikazu Kobayashi, and Masanao Obayashi. Time series forecasting using a deep belief network with restricted boltzmann machines. *Neurocomputing*, 137:47–56, 2014.

[81] Alexander Lavin and Subutai Ahmad. Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark. In *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*, pages 38–44. IEEE, 2015.

[82] Gen Li and Jason J Jung. Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges. *Information Fusion*, 91:93–102, 2023.

[83] Kun-Lun Li, Hou-Kuan Huang, Sheng-Feng Tian, and Wei Xu. Improving one-class svm for anomaly detection. In *Proceedings of the 2003 international conference on machine learning and cybernetics (IEEE Cat. No. 03EX693)*, volume 5, pages 3077–

3081. IEEE, 2003.
[84] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.
[85] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. Anomaly detection with generative adversarial networks for multivariate time series. *arXiv preprint arXiv:1809.04758*, 2018.
[86] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International conference on artificial neural networks*, pages 703–716. Springer, 2019.
[87] Longyuan Li, Junchi Yan, Haiyang Wang, and Yaohui Jin. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *IEEE transactions on neural networks and learning systems*, 32(3):1177–1191, 2020.
[88] Yifan Li, Xiaoyan Peng, Jia Zhang, Zhiyong Li, and Ming Wen. Dct-gan: Dilated convolutional transformer-based gan for time series anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 2021.
[89] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 3220–3230, 2021.
[90] Longyuan Li, Junchi Yan, Qingsong Wen, Yaohui Jin, and Xiaokang Yang. Learning robust deep state space for unsupervised anomaly detection in contaminated time-series. *IEEE Transactions on Knowledge and Data Engineering*, 2022.
[91] Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.
[92] Jessica Lin, Eamonn Keogh, Stefano Lonardi, Jeffrey P Lankford, and Donna M Nystrom. Visually mining and monitoring massive time series. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 460–469, 2004.
[93] Shuyu Lin, Ronald Clark, Robert Birke, Sandro Schönborn, Niki Trigoni, and Stephen Roberts. Anomaly detection for time series using vae-lstm hybrid model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4322–4326. Ieee, 2020.
[94] Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. Diffusion models for time-series applications: a survey. *Frontiers of Information Technology & Electronic Engineering*, pages 1–23, 2023.
[95] Jiexi Liu and Songcan Chen. Timesurl: Self-supervised contrastive learning for universal time series representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13918–13926, 2024.
[96] Yong Liu, Haoran Zhang, Chenyu Li, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer: Generative pre-trained transformers are large time series models. In *Forty-first International Conference on Machine Learning*.
[97] Yunxiao Liu, Youfang Lin, QinFeng Xiao, Ganghui Hu, and Jing Wang. Self-adversarial variational autoencoder with spectral residual for time series anomaly detection. *Neurocomputing*, 458:349–363, 2021.
[98] Chengyi Liu, Wenqi Fan, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. Generative diffusion models on graphs: Methods and applications. In *IJCAI*, 2023.
[99] Chengyi Liu, Wenqi Fan, Yunqing Liu, Jiatong Li, Hang Li, Hui Liu, Jiliang Tang, and Qing Li. Generative diffusion models on graphs: Methods and applications. *arXiv preprint arXiv:2302.02591*, 2023.
[100] Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *arXiv preprint arXiv:2402.02370*, 2024.
[101] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.
[102] Igor Melnyk, Arindam Banerjee, Bryan Matthews, and Nikunj Oza. Semi-markov switching vector autoregressive model-based anomaly detection in aviation systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1065–1074, 2016.
[103] Kristian Miok, Dong Nguyen-Doan, Daniela Zaharie, and Marko Robnik-Šikonja. Generating data using monte carlo dropout. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 509–515. IEEE, 2019.
[104] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE engineering in medicine and biology magazine*, 20(3):45–50, 2001.
[105] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.
[106] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.
[107] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters*, 3(3):1544–1551, 2018.
[108] Badri N Patro and Vijay S Agneeswaran. Simba: Simplified mamba-based architecture for vision and multivariate time series. *arXiv preprint arXiv:2403.15360*, 2024.
[109] Hengzhi Pei, Kan Ren, Yuqing Yang, Chang Liu, Tao Qin, and Dongsheng Li. Towards generating real-world time series data. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 469–478. IEEE, 2021.
[110] Ioana Pintilie, Andrei Manolache, and Florin Brad. Time series anomaly detection using diffusion-based models. In *2023 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 570–578. IEEE, 2023.
[111] Alan Preciado-Grijalva and Victor Rodrigo Iza-Teran. Anomaly detection of wind turbine time series using variational recurrent autoencoders. *arXiv preprint arXiv:2112.02468*, 2021.
[112] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China technological sciences*, 63(10):1872–1897, 2020.
[113] Lawrence Rabiner and Biinghwang Juang. An introduction to hidden markov models. *ieee assp magazine*, 3(1):4–16, 1986.
[114] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
[115] Huorong Ren, Zhixing Ye, and Zhiwu Li. Anomaly detection based on a dynamic markov model. *Information Sciences*, 411:52–65, 2017.
[116] Shyam Sundar Saravanan, Tie Luo, and Mao Van Ngo. Tsi-gan: Unsupervised time series anomaly detection using convolutional cycle-consistent generative adversarial networks. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 39–54. Springer, 2023.
[117] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. Anomaly detection in time series: a comprehensive evaluation. *Proceedings of the VLDB Endowment*, 15(9):1779–1797, 2022.
[118] V Shanmuganathan and A Suresh. Lstm-markov based efficient anomaly detection algorithm for iot environment. *Applied Soft Computing*, 136:110054, 2023.
[119] Kamran Shaukat, Talha Mahboob Alam, Suhuai Luo, Shakir Shabbir, Ibrahim A Hameed, Jiaming Li, Syed Konain Abbas, and Umair Javed. A review of time-series anomaly detection techniques: A step to future perspectives. In *Advances in Infor-*

mation and Communication: Proceedings of the 2021 Future of Information and Communication Conference (FICC), Volume 1, pages 865–877. Springer, 2021.

[120] Yunfei Shi, Bin Wang, Yanwei Yu, Xianfeng Tang, Chao Huang, and Junyu Dong. Robust anomaly detection for multivariate time series through temporal gcns and attention-based vae. *Knowledge-Based Systems*, 275:110725, 2023.

[121] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. Anomaly detection in streams with extreme value theory. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1067–1075, 2017.

[122] Emma Stewart, Anna Liao, and Ciaran Roberts. Open μpmu: A real world reference distribution micro-phasor measurement unit data set for research and application development. 2016.

[123] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2828–2837, 2019.

[124] Yong Sun, Wenbo Yu, Yuting Chen, and Aishwarya Kadam. Time series anomaly detection based on gan. In *2019 sixth international conference on social networks analysis, management and security (SNAMS)*, pages 375–382. IEEE, 2019.

[125] Peiwang Tang and Xianchao Zhang. Mtsmae: Masked autoencoders for multivariate time-series forecasting. In *2022 IEEE 34th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 982–989. IEEE, 2022.

[126] Achille Thin, Nikita Kotelevskii, Arnaud Doucet, Alain Durmus, Eric Moulines, and Maxim Panov. Monte carlo variational auto-encoders. In *International Conference on Machine Learning*, pages 10247–10257. PMLR, 2021.

[127] Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*, pages 1578–1585. IEEE, 2017.

[128] Chang Wang, Yongxin Zhu, Weiwei Shi, Victor Chang, Pandi Vijayakumar, Bin Liu, Yishu Mao, Jiabao Wang, and Yiping Fan. A dependable time series analytic framework for cyber-physical systems of iot-based smart grid. *ACM Transactions on Cyber-Physical Systems*, 3(1):1–18, 2018.

[129] Xixuan Wang, Dechang Pi, Xiangyan Zhang, Hao Liu, and Chang Guo. Variational transformer-based anomaly detection approach for multivariate time series. *Measurement*, 191:110791, 2022.

[130] Lei Wang, Liang Zeng, and Jian Li. Aec-gan: adversarial error correction gans for auto-regressive long time-series generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 10140–10148, 2023.

[131] Yanling Wang, Yuchen Liu, Qian Wang, Cong Wang, and Chenliang Li. Poisoning self-supervised learning based sequential recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 300–310, 2023.

[132] Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Yong Liu, Mingsheng Long, and Jianmin Wang. Deep time series models: A comprehensive survey and benchmark. *arXiv preprint arXiv:2407.13278*, 2024.

[133] Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Han Zhao, Daling Wang, and Yifei Zhang. Is mamba effective for time series forecasting? *arXiv preprint arXiv:2403.11144*, 2024.

[134] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. *arXiv preprint arXiv:2002.12478*, 2020.

[135] Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*, 2022.

[136] Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

[137] Xianli Wu, Huchang Liao, and Ming Tang. Decision making towards large-scale alternatives from multiple online platforms by a multivariate time-series-based method. *Expert Systems with Applications*, 212:118838, 2023.

[138] Chunjing Xiao, Zehua Gou, Wenxin Tai, Kunpeng Zhang, and Fan Zhou. Imputation-based time-series anomaly detection with conditional weight-incremental diffusion models. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2742–2751, 2023.

[139] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022.

[140] Kiyoung Yang and Cyrus Shahabi. An efficient k nearest neighbor search for multivariate time series. *Information and Computation*, 205(1):65–98, 2007.

[141] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

[142] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2023*, pages 1063–1073, 2023.

[143] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[144] Hsiang-Fu Yu, Nikhil Rao, and Inderjit S Dhillon. Temporal regularized matrix factorization for high-dimensional time series prediction. *Advances in neural information processing systems*, 29, 2016.

[145] Jianye Zhang and Peng Yin. Multivariate time series missing data imputation using recurrent denoising autoencoder. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 760–764. IEEE, 2019.

[146] Jiuqi Elise Zhang, Di Wu, and Benoit Boulet. Time series anomaly detection for smart grids: A survey. In *2021 IEEE Electrical Power and Energy Conference (EPEC)*, pages 125–130. IEEE, 2021.

[147] Xiang Zhang, Ziyuan Zhao, Theodoros Tsiligkaridis, and Marinka Zitnik. Self-supervised contrastive pre-training for time series via time-frequency consistency. *Advances in Neural Information Processing Systems*, 35:3988–4003, 2022.

[148] Bendong Zhao, Huanzhang Lu, Shangfeng Chen, Junliang Liu, and Dongya Wu. Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, 28(1):162–169, 2017.

[149] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. Beatgan: Anomalous rhythm detection using adversarially generated time series. In *IJCAI*, volume 2019, pages 4433–4439, 2019.

[150] Tian Zhou, Peisong Niu, Xue Wang, Liang Sun, and Rong Jin. One fits all: Universal time series analysis by pretrained lm and specially designed adaptors. *arXiv preprint arXiv:2311.14782*, 2023.

[151] Haiqi Zhu, Chunzhi Yi, Seungmin Rho, Shaohui Liu, and Feng Jiang. An interpretable multivariate time-series anomaly detection method in cyber-physical systems based on adaptive mask. *IEEE Internet of Things Journal*, 2023.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.
Citation information: DOI: 10.1109/TAI.2025.3614213,

Jie Cao received the Ph.D. degree in information science and engineering from Southeast University, Nanjing, China, in 2002.

He is currently a Professor of Research Institute of Big Knowledge, Hefei University of Technology, Hefei, China. He has published 100+ articles in prestigious conferences and journals, including ACM *SIGKDD Conference on Knowledge Discovery and Data Mining* (KDD), the *International Joint Conference on Artificial Intelligence* (IJCAI), the IEEE *International Conference on Data Mining* (ICDM), IEEE *Transactions on Knowledge and Data Engineering* (TKDE), IEEE *Transactions on Fuzzy Systems* (TFS), IEEE *Transactions on Cybernetics* (TCYB), and IEEE *Transactions on Neural Networks and Learning Systems* (TNNLS). His main research interests include data mining, business intelligence, social computing, and deep learning.

Dr. Cao has been selected in the Program for New Century Excellent Talents in University and awarded with the Young and Mid-Aged Expert with Outstanding Contribution in Jiangsu Province. He is the Associate Editor of the *World Wide Web-Internet and Web Information Systems* (WWW) and *Neurocomputing*.

Jiawei Miao received the B.E. degree in computer science and technology from Jiangsu University in 2019 and the M.E. degree in software engineering from Nanjing University of Finance and Economics in 2024. He is currently pursuing the Ph.D. degree in Management Science and Engineering at the School of Management, Hefei University of Technology. His current research interests include data mining, machine learning, artificial intelligence and time series anomaly detection.

Haicheng Tao (Member, IEEE) received the PhD degree in computer science from Nanjing University of Science and Technology, Nanjing, China. He is currently a lecturer with the Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing, China. His main research interests include data mining, machine learning, artificial intelligence, and optimization.

Youquan Wang received the Ph.D. degree in Computer Application Technology from Nanjing University of Science and Technology, Nanjing, China, in 2017.

He is now an associate professor at the Jiangsu Provincial Key Laboratory of E-Business at the Nanjing University of Finance and Economics. He has published 30+ refereed journal and conference papers in these areas. He is the Member of the IEEE, ACM and CCF. His research interests include deep learning and data mining.

Jia Wu (SM'21) received the Ph.D. degree in computer science from the University of Technology Sydney, Australia. He is currently an ARC DECRA Fellow in the Department of Computing, Macquarie University, Sydney, Australia. His current research interests include data mining and machine learning. Since 2009, he has published 100+ refereed journal and conference papers, including IEEE *Transactions on Pattern Analysis and Machine Intelligence* (TPAMI), IEEE *Transactions on Knowledge and Data Engineering* (TKDE), ACM *SIGKDD Conference on Knowledge Discovery and Data Mining* (KDD), *The Web Conference* (WWW), and *Neural Information Processing Systems* (NeurIPS).

Dr Wu was the recipient of SDM'18 Best Paper Award in Data Science Track, IJCNN'17 Best Student Paper Award, and ICDM'14 Best Paper Candidate Award. He is the Associate Editor of the ACM *Transactions on Knowledge Discovery from Data* (TKDD) and *Neural Networks* (NN).

Zidong Wang (Fellow, IEEE) received the B.Sc. degree in mathematics in 1986 from Suzhou University, Suzhou, China, the M.Sc. degree in applied mathematics and the Ph.D. degree in electrical engineering both from Nanjing University of Science and Technology, Nanjing, China, in 1990 and 1994, respectively. He is currently a Professor of Dynamical Systems and Computing in the Department of Computer Science, Brunel University London, Uxbridge, U.K. From 1990 to 2002, he held teaching and research appointments in universities in China, Germany, and the U.K. Prof. Wang's research interests include dynamical systems, signal processing, bioinformatics, control theory and applications. He has published a number of papers in international journals. He is a holder of the Alexander von Humboldt Research Fellowship of Germany, the JSPS Research Fellowship of Japan, and the William Mong Visiting Research Fellowship of Hong Kong. Prof. Wang serves (or has served) as the Editor-in-Chief for International Journal of Systems Science, the Editor-in-Chief for Neurocomputing, the Editor-in-Chief for Systems Science and Control Engineering, and an Associate Editor for 12 international journals, including IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY, IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON SIGNAL PROCESSING, and IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS–PART C. He is a Member of the Academia Europaea, a Member of the European Academy of Sciences and Arts, an Academician of the International Academy for Systems and Cybernetic Sciences, a Fellow of the IEEE, a Fellow of the Royal Statistical Society, and a member of program committee for many international conferences.

Xindong Wu is Director and Professor of the Key Laboratory of Knowledge Engineering with Big Data (sponsored by the Ministry of Education of China), Hefei University of Technology, China. His research interests include big data analytics, data mining and knowledge engineering. He received his Bachelor's and Master's degrees in Computer Science from the Hefei University of Technology, China, and his Ph.D. degree in Artificial Intelligence from the University of Edinburgh, Britain. He is a Foreign Member of the Russian Academy of Engineering, and a Fellow of IEEE and the AAAS (American Association for the Advancement of Science).

Dr. Wu is the Steering Committee Chair of the IEEE International Conference on Data Mining (ICDM), and the Editor in-Chief of Knowledge and Information Systems (KAIS, by Springer). He was the Editor-in-Chief of the IEEE Transactions on Knowledge and Data Engineering (TKDE) between 2005 and 2008 and Co-Editor-in-Chief of the ACM Transactions on Knowledge Discovery from Data Engineering between 2017 and 2020. He served as a program committee chair/co-chair for ICDM 2003 (the 3rd IEEE International Conference on Data Mining), KDD 2007 (the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining), CIKM 2010 (the 19th ACM Conference on Information and Knowledge Management), and ICBK 2017 (the 8th IEEE International Conference on Big Knowledge).