M-TSINR: Multiscale Implicit Neural Representations via Mamba Encoder for Time Series Anomaly Detection

Ke Liu, Mengxuan Li, Qianqian Shen, Yang Gao, Haishuai Wang*
Zhejiang Key Lab of Accessible Perception & Intelligent Systems,
College of Computer Science, Zhejiang University
Hangzhou, Zhejiang, China
{keliu99, limx97, shenqq377, gaoyang9507, haishuai.wang}@zju.edu.cn

***Abstract*—Time series anomaly detection is a critical task in various domains, including industrial monitoring, healthcare, and finance. Existing time series anomaly detection methods typically rely on discrete grid representations to store and process data, which can lead to issues in capturing the true nature of continuous time series signals. Recently, some researchers use Implicit Neural Representation (INR) to model continuous functions, offering a more natural approach for learning and representing real-world continuous time series signals. However, current INR-based anomaly detection methods suffer from limited generalization and suboptimal performance and efficiency. To address these challenges, we propose M-TSINR, a novel framework that leverages the strengths of INR to improve both the performance and efficiency of time series anomaly detection. Specifically, we design a form of multiscale INR continuous functions, utilizing multiple neural networks to represent different temporal scales of the data, enabling the model to extract hierarchical features from fine-grained to coarse patterns. Additionally, we use a Mamba encoder to generate INR parameters, which significantly improves both the efficiency and the ability to capture complex temporal dependencies. Extensive experiments conducted on five real-world anomaly detection benchmark datasets demonstrate that M-TSINR achieves superior performance compared to other state-of-the-art anomaly detection methods.**

***Index Terms*—time series anomaly detection, implicit neural representations, unsupervised learning**

I. Introduction

Time series anomaly detection involves recognizing unusual points or patterns in temporal data that significantly diverge from expected trends or normal behavior. Such anomalies often signal critical events, including equipment failures, fraudulent activities, or emerging medical conditions. Its relevance spans multiple sectors, such as industrial systems, healthcare, financial services, and environmental monitoring [1]–[3]. Due to the complex temporal dependencies inherent in time series data, accurate anomaly detection is challenging, and traditional methods often struggle to handle high-dimensional or noisy data. As a result, more advanced techniques, such as machine learning and deep learning, have become essential for effectively detecting anomalies and providing actionable insights in real-world applications.

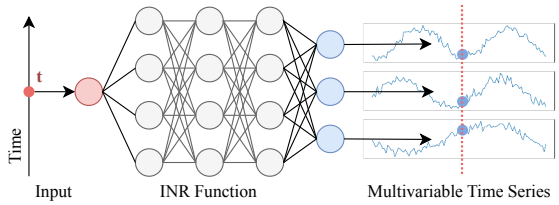

Fig. 1. The depiction of INR applied to time series. The input is timestamp t, and the output is the value of variables corresponding to time series at t.

Many deep learning-based approaches have been proposed for anomaly detection in time series data [4]–[6]. However, these methods typically rely on discrete grid representations to store and process the data. This discretization introduces several notable challenges. Firstly, the continuous nature of time series signals is compromised during the discretization process, leading to the loss of fine-grained temporal information. Consequently, subtle patterns or small deviations from normal behavior may remain undetected. Additionally, discretizing the data into fixed intervals can result in inefficiencies, as such representations fail to capture the full spectrum of temporal dependencies and smooth variations inherent in the signals. This issue becomes particularly pronounced when dealing with high-frequency or long-duration time series data, where the granularity of the grid may not align with the underlying temporal structure. These limitations underscore the need for more flexible and efficient approaches that are better suited to model the continuous characteristics of time series data.

In recent years, Implicit Neural Representation (INR) has garnered significant attention from the research community. INR is a powerful framework that utilizes neural networks to represent continuous functions, offering a more flexible and efficient approach to modeling complex data compared to traditional discrete grid-based methods [7], [8]. Unlike conventional methods that discretize signals into fixed intervals, INR directly learns continuous representations, enabling

* Corresponding author.

979-8-3315-6689-0/25/$31.00 ©2025 IEEE
DOI 10.1109/ICKG66886.2025.00037

it to model high-dimensional, continuous data with greater precision. This ability to represent data as continuous functions makes INR particularly well-suited for tasks involving real-world signals, where capturing fine-grained temporal dependencies and variations is crucial. Given the advantages of INR in modeling continuous signals, we propose to leverage INR for time series anomaly detection. By using INR, we aim to overcome the limitations of traditional methods, such as the loss of information due to discretization and the inability to fully capture temporal dependencies. It has been proven that INR-based methods can provide a more accurate and efficient solution for detecting anomalies in time series data, enabling better detection of subtle deviations and enhancing the overall performance of anomaly detection systems [9].

Although several INR-based methods have been proposed for time series anomaly detection [9], [10], they still face challenges in terms of generalization and efficiency, limiting their practical applicability. These methods use simplistic INR continuous functions, such as a single Multilayer Perceptron (MLP) to represent the entire time series or rely on monotonic and periodic functions to separately model trend and seasonality. While these approaches may be effective for certain types of data, they are limited in their ability to capture the complex, non-linear interactions and long-range dependencies present in real-world time series. As a result, they struggle to generalize to more complex or irregular temporal patterns. Additional, the continuity represented in INR functions involves numerous parameters, which leads to high memory and computational demands. Training these models can require significant time and resources, especially when dealing with time series data that spans long periods or has high dimensionality.

In this paper, we propose M-TSINR, a time series anomaly detection method based on multiscale INR continuous functions via Mamba encoder. The core of our approach lies in the design of multiscale INR continuous functions, which enable the model to learn hierarchical representations of the time series across multiple temporal scales. By capturing fine-grained temporal patterns at different scales, our method enhances the model's ability to generalize across diverse datasets, especially those with varying levels of noise, complexity, and seasonality. This multiscale representation allows the model to capture both global trends and local anomalies more effectively, addressing the limitations of existing approaches that rely on simplistic, single-scale functions. In addition, we incorporate a Mamba encoder to generate the INR parameters, optimizing the overall computational efficiency. The Mamba encoder significantly reduces the model's training time and memory requirements by generating efficient, compact representations of the time series data, without compromising the quality of the learned features. The major contributions of this paper are summarized as follows:

- We propose M-TSINR, a time series anomaly detection method that leverages INR to learn continuous functions, thereby overcoming the information loss inherent in grid-based representations. By utilizing INR, M-TSINR is able to preserve the underlying continuity of the time series data, improving the accuracy and robustness of anomaly detection in real-world scenarios.
- We design a novel form for multiscale INR continuous functions, which allows M-TSINR to capture hierarchical temporal patterns across different scales. This multiscale strategy enhances the ability to detect anomalies at various granularities by leveraging different temporal scales.
- We incorporate the Mamba encoder to efficiently generate INR parameters. It significantly decreases memory usage and training time, making the model more practical for large-scale and high-dimensional datasets.
- Extensive experiments conducted on five benchmark time series anomaly detection datasets verify that M-TSINR achieves superior performance, demonstrating its effectiveness and scalability for real-world applications in time series anomaly detection.

II. Related Work

A. Time Series Anomaly Detection

Time series anomaly detection has emerged as an essential task in multiple fields. Early approaches were primarily rooted in statistical models, often relying on assumptions about the underlying data distribution. Representative examples include z-score–based techniques [11], moving average methods [12], [13], and autoregressive models [14], [15]. While simple and interpretable, these approaches often fall short when dealing with complex or high-dimensional time series [16].

The advent of machine learning has led to the development of more sophisticated techniques for anomaly detection. Unlike traditional statistical approaches, these methods generally avoid strict distributional assumptions and instead leverage mathematical models or heuristic strategies to identify outliers. Representative categories include clustering-based methods [17], [18], distance-based approaches [19], [20], and isolation forest algorithms [21]. Despite their effectiveness, these techniques often face challenges in capturing subtle anomalies and typically rely on extensive feature engineering.

In contrast to traditional machine learning approaches, deep learning-based anomaly detection methods leverage neural networks to learn complex hierarchical representations of data automatically. Techniques such as Auto-Encoder (AE) [4], [22], Recurrent Neural Network (RNN) [23], Long Short-Term Memory (LSTM) [24], [25], Convolutional Neural Network (CNN) [5], [26], MLP [27], and Transformer [6], [28] can model intricate patterns in sequential and high-dimensional data, making them more effective at detecting subtle and context-dependent anomalies. These methods can capture long-range dependencies and nonlinear relationships, which are often challenging for traditional machine learning methods.

However, current time series anomaly detection methods often rely on discretized grid representations, where time is divided into fixed intervals. This approach can lead to information loss and fail to capture continuous dynamics and subtle anomalies that occur between discrete time steps.

B. Implicit Neural Representations

In recent years, INRs have gained significant attention as a versatile framework for modeling complex data modalities, including images [29], [30], videos [31], [32], time series [9], [33], and 3D objects [8], [34]. INRs encode data as continuous functions parameterized by neural networks, enabling flexible and efficient modeling of high-dimensional signals. The core principle is to use a neural network to map input coordinates directly to signal values, bypassing the limitations of discrete grids or fixed basis functions.

Currently, INRs have also been explored in the context of time series anomaly detection. INRAD [10] employs INR to mitigate the complexities and numerous hyperparameters (e.g., sliding windows) typically required by deep learning models. Nonetheless, its deployment in real-world scenarios is hindered by several limitations. Specifically, INRAD relies on a relatively simple MLP structure to encode time series as continuous functions, which may fall short in capturing intricate temporal patterns. In addition, its meta-learning strategy requires training a fresh INR network for each unseen sequence during testing, leading to increased computational cost and reduced practicality. To address some of these issues, TSINR [9] exploits the spectral bias of INRs to preferentially fit normal patterns while highlighting anomalies. However, its design of INR continuous functions, based on monotonic and periodic formulations to model trend and seasonality, can restrict adaptability to more complex data. Moreover, the adoption of a Transformer encoder for generating INR parameters introduces additional complexity and computational overhead. Overall, while the use of INRs in time series anomaly detection shows promise, their potential remains underexplored and calls for deeper investigation.

C. Mamba

The Transformer architecture has achieved remarkable success across multiple domains, such as natural language processing [35], [36], computer vision [37], [38], and time series analysis [39], [40]. However, its computational efficiency remains a significant bottleneck when modeling long sequences. To address this issue, several improvements have been proposed, including linear attention and structured state space model (SSM), which reduce computational complexity but still face limitations when dealing with complex modalities.

Mamba [41] is proposed to address these challenges. It treats SSM parameters as functions of the input, allowing selective propagation or forgetting of information based on the current token, which enhances efficiency in long-sequence modeling. Additionally, Mamba integrates a hardware-aware parallel algorithm to accelerate computation. This simplified architecture achieves 5x higher throughput than Transformers and scales linearly with sequence length. With its exceptional efficiency and ability to model long-range dependencies, Mamba is emerging as a promising architecture that has the potential to outperform Transformer-based models in various domains [42]–[44].

However, to the best of our knowledge, no prior research has explored the use of Mamba for generating INR parameters. This gap presents an opportunity for us to investigate the potential of leveraging Mamba in the context of INR, leading to more effective solutions for time series anomaly detection.

III. Preliminaries

Structured State Space Model (SSM) [45] is designed for sequence modeling, offering a structured approach to capture dynamic relationships in sequential data. By modeling state transitions and observations explicitly, SSM provides an efficient alternative to attention-based models like Transformer, balancing expressiveness with computational efficiency. Unlike Transformer, which involves dense interactions across all time steps, SSM uses a more compact representation of state evolution, reducing complexity while capturing long-range dependencies. This makes SSM well-suited for resource-constrained or real-time applications, offering a promising direction for advancing sequence modeling in deep learning.

Mathematically, SSM transforms a 1-D input signal $u(t)$ into an d-D latent state $x(t)$, which is then projected to a 1-D output signal $y(t)$. SSM is defined as follows:

$$x'(t) = \mathbf{A}x(t) + \mathbf{B}u(t), \tag{1}$$

$$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t), \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{d\times d}$, $\mathbf{B} \in \mathbb{R}^{d\times 1}$, $\mathbf{C} \in \mathbb{R}^{1\times d}$, and $D \in \mathbb{R}^1$ are the weighting parameters. To integrate continuous-time SSM into deep models, they must first be discretized. Specifically, for the time interval $[t_a, t_b]$, the analytic solution of the hidden state variable $x(t)$ at $t = t_b$ can be expressed as follows:

$$x_t = \overline{\mathbf{A}}x_{t-1} + \overline{\mathbf{B}}u_t, \tag{3}$$

$$y_t = \overline{\mathbf{C}} * x_t, \tag{4}$$

where $\overline{\mathbf{A}}$, $\overline{\mathbf{B}}$, and $\overline{\mathbf{C}}$ denote discretized SSM matrices defined with a step size Δ:

$$\overline{\mathbf{A}} = (\mathbf{I} - \Delta/2 \cdot \mathbf{A})^{-1}(\mathbf{I} + \Delta/2 \cdot \mathbf{A}), \tag{5}$$

$$\overline{\mathbf{B}} = (\mathbf{I} - \Delta/2 \cdot \mathbf{A})^{-1}\Delta\mathbf{B}, \tag{6}$$

$$\overline{\mathbf{C}} = \mathbf{C}. \tag{7}$$

The discrete-time SSM typically consists of a state transition model and an observation model, both defined for discrete time steps. These models are commonly used in applications where the data is available in discrete time intervals. The discretization process transforms the continuous-time dynamics into a form that can be efficiently handled by computational models, facilitating their integration into deep learning frameworks.

IV. Methodology

More specifically, we design a novel form of multiscale INR continuous functions that captures fine-grained information and enhances generalization ability by learning hierarchical representations across multiple temporal scales. Additionally, we employ a Mamba encoder to generate INR parameters, effectively optimizing computational resources. In this section, we first present the problem statement and provide an overview

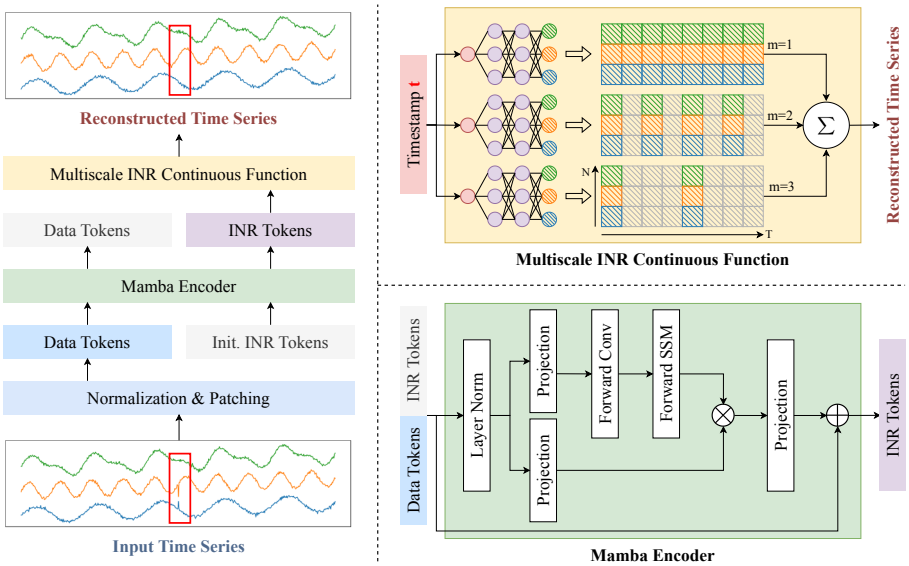

Fig. 2. The overall workflow of the proposed M-TSINR method. The INR tokens generated by the Mamba encoder serve as the parameters for the multiscale INR continuous functions. The input to these multiscale INR continuous functions is the timestamp t. The red box area indicates that the abnormal data present in the input time series data disappears in the reconstructed data, leading to an increase in reconstruction loss, which in turn allows for the precise identification of anomalies.

of the proposed methodology. Subsequently, we introduce the Mamba encoder, which is employed to generate the INR parameters, and describe the novel form of multiscale INR continuous functions that we design to enhance the ability to capture hierarchical temporal representations.

A. Problem Statement

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ denote a time-series data sequence of length T, where each observation $\mathbf{x}_t \in \mathbb{R}^N$ represents a vector of N-dimensional features at time t. The goal of anomaly detection is to identify a subset of time points $\mathcal{A} \subseteq \{1, 2, \ldots, T\}$, where each index $t \in \mathcal{A}$ corresponds to an abnormal observation. More specifically, the identification of anomalies is based on the quality of the reconstruction performance. The mean squared error (MSE) loss function is used as the reconstruction error, which is defined as follows:

$$\mathcal{L} = \left\| \mathbf{X} - \hat{\mathbf{X}} \right\|_2^2, \tag{8}$$

where $\hat{\mathbf{X}}$ denotes the reconstructed time series. The points with higher reconstruction errors compared to normal patterns are flagged as anomalies, indicating that their characteristics deviate from the expected behavior modeled by the system.

B. Overview

Fig. 2 illustrates the overall workflow of the proposed M-TSINR method. The input time series data is normalized and divided into patches to obtain the corresponding data tokens $\mathbf{D}$. Next, we introduce a set of learnable parameters in the form of INR tokens $\mathbf{I}_{\text{init}}$, which are randomly initialized as continuous vector representations. These INR tokens are integrated with the data tokens by concatenation, creating a combined input $\mathbf{X}'$ that is passed through the Mamba encoder. Within the Mamba encoder, the knowledge interacts with data tokens and INR tokens. The learned INR tokens $\mathbf{I}$ serve as the network parameters for the designed multiscale INR continuous functions. These continuous functions take timestamps $\{t\}_{i=1}^{T}$ as input and implicitly learn the multiscale temporal dynamics. It then reconstructs the input signal based on these learned representations, allowing the model to capture long-range dependencies and variations in the time series. The details of the Mamba encoder and the multiscale INR continuous functions are discussed in Sections IV-C and IV-D.

C. Mamba Encoder

We use a Mamba encoder to generate INR parameters instead of the Transformer encoder. Mamba is more computationally efficient for long-sequence modeling, as it scales linearly with sequence length, unlike Transformers, which suffer from quadratic complexity. This results in faster training and inference, as well as reduced memory usage.

Theoretically, it takes the concatenation of data tokens $\mathbf{D}$ and the randomly initialized INR tokens $\mathbf{I}_{\text{init}}$ as input:

$$\mathbf{X}' = [\mathbf{D}, \mathbf{I}_{\text{init}}]. \tag{9}$$

Then layer normalization is applied to the input $\mathbf{X}'$ and the normalized input is passed through two parallel projection layers, producing the intermediate representations $\mathbf{X}_1$ and $\mathbf{X}_2$:

$$\mathbf{X}_1 = \text{Projection}_1\left(\text{LayerNorm}\left(\mathbf{X}'\right)\right), \tag{10}$$

$$\mathbf{X}_2 = \text{Projection}_2\left(\text{LayerNorm}\left(\mathbf{X}'\right)\right). \tag{11}$$

The projection output $\mathbf{X}_1$ undergoes a forward convolution and a subsequent forward SSM operation, while $\mathbf{X}_2$ interacts with $\mathbf{X}_1$ through element-wise multiplication. The resulting product is further passed through another projection layer. Finally, the

original input $\mathbf{X}'$ is added to the output of this projection layer, resulting in the learned INR tokens $\mathbf{I}$, which serve as the final output of the encoder. The entire process can be defined as follows:

$$\mathbf{I} = \mathbf{X}' + \text{Projection}_3\left(\mathbf{X}_2 \odot \text{SSM}\left(\text{Conv}\left(\mathbf{X}_1\right)\right)\right). \quad (12)$$

D. Multiscale INR Continuous Functions

We design a novel form of multiscale INR continuous functions to implicitly capture both fine-grained and global temporal patterns, thereby improving generalization and robustness to varying data dynamics. More specifically, the goal is to represent the time series data as a form of continuous functions $f(t)$ using multiscale representations.

In this multiscale strategy, a hyperparameter M denotes the multiscale factor, which determines the number of scales used in the INR continuous functions. For each scale $m \in \{1, 2, \ldots, M\}$, a different continuous function $f^{(m)}(t)$ is learned to represent the time series data at that scale. Here, the data is modeled at different granularities, which are determined by how the time steps are sampled at each scale. For example, $m = 1$ represents the original time series, while $m = 2$ represents a downsampled version of the time series with a step size of 2, and so on.

Mathematically, for each scale m, the function $f^{(m)}(t)$ is represented by an MLP parameterized by θ_m, which encodes the time series data at that scale:

$$f^{(m)}\left(t; \theta_m\right) = \text{MLP}_m\left(t; \theta_m\right), \quad (13)$$

where θ_m represents the parameters of the MLP at the m^{th} scale and θ_m is derived from the INR tokens $\mathbf{I}$. Then the outputs from all M scales are combined using a weighted summation to reconstruct the data. This yields the reconstructed feature at timestamp t, which is computed as follows:

$$f(t) = \sum_{m=1}^{M} \alpha_m \cdot f^{(m)}\left(t; \theta_m\right), \quad (14)$$

where α_m represents the learnable weight assigned to the m^{th} scale. At this point, we obtain the reconstructed data.

V. Experiments

A. Datasets

To evaluate the performance of the proposed M-TSINR method, we use five publicly available multivariate anomaly detection datasets: PSM [46], MSL [47], SMAP [47], PTB-XL [48], and SKAB [49]. These datasets are all openly accessible for download and have been pre-divided into training and test sets. Each dataset presents distinct challenges for anomaly detection in time series data, spanning various domains such as mechanical systems, predictive maintenance, environmental monitoring, healthcare, and industrial applications. The inclusion of diverse data sources ensures a comprehensive evaluation of the robustness and generalization capability of our M-TSINR method.

B. Baseline Methods

To validate the performance of the proposed method, we compare it with 12 state-of-the-art methods. These methods include both approaches specifically designed for time series anomaly detection as well as general time series modeling methods. The selected baselines encompass a variety of architectures, including CNN, Transformer, MLP, and INR. The 12 baseline methods are as follows: TSINR [9], FPT [6], TimesNet [5], ETSformer [39], FEDformer [40], LightTS [50], DLinear [27], Autoformer [51], Pyraformer [52], Anomaly-Transformer [28], Informer [53] and Transformer [54]. The comparison allows us to assess the effectiveness of our method across different domains and model architectures.

C. Evaluation Metrics

To evaluate the performance of the proposed method, we employ three evaluation metrics: F1 score, ROC-AUC score, and VUS score [55]. The F1 score is widely regarded as the most commonly used metric for anomaly detection, as it provides a harmonic balance between precision and recall and is threshold-dependent, meaning its value varies with different decision thresholds. In contrast, both ROC-AUC score and VUS score are threshold-independent metrics, that evaluate model performance across all possible decision thresholds, thus offering a more comprehensive assessment of the model's ability to identify anomalies.

D. Experimental Settings

To ensure a fair comparison, we employ the classical reconstruction error consistently across all baseline models. Additionally, we exclude the pre-trained large model encoder from the TSINR method for this evaluation. Data processing is performed using uniform methods and parameter configurations for all experiments, and these settings align with those used in previous works [6], [9], [28]. Specifically, we apply a fixed sliding window size of 100 across all datasets. To compute F1 score, the threshold is determined by the hyper-parameter γ, which represents the proportion of test data considered as anomalies. For the SKAB dataset, γ is set to 10, while for all other datasets, it is set to 1. These settings follow those adopted in previous works [6], [9], [28]. For main results, our M-TSINR model utilizes 5 layers for each MLP architecture within the multiscale INR continuous functions, with a hidden dimension of 64. The Mamba encoder is configured with 3 blocks and the hidden state dimension is set to 256. All experiments are conducted using the ADAM optimizer [56] with an initial learning rate of 10^{-3}. Each dataset is processed using a single NVIDIA RTX 3090 GPU with 24GB.

E. Main Results

Shown in Table I, we compare our M-TSINR method with 12 state-of-the-art approaches across 5 multivariate anomaly detection datasets. Our method consistently outperforms all baselines on all datasets, demonstrating superior anomaly detection performance. Although our method does not achieve

TABLE I
THE OVERALL RESULTS ON 5 REAL-WORLD MULTIVARIATE DATASETS. THE F1 SCORE, ROC-AUC SCORE, AND VUS SCORE ARE REPORTED. THE BEST RESULTS ARE IN **BOLD** AND THE SECOND BEST RESULTS ARE UNDERLINED.

Dataset	Metric	Trans.	FED.	Anomaly.	Auto.	Pyra.	In.	ETS.	LightTS	DLinear	TimesNet	FPT	TSINR	Ours
PSM	F1	0.682	0.900	0.904	0.883	0.937	0.908	0.946	0.959	0.966	**0.974**	0.972	0.927	0.934
	AUC	0.721	0.662	0.666	0.661	0.704	0.712	0.622	0.585	0.565	0.590	0.578	0.720	**0.725**
	VUS	0.667	0.563	0.608	0.556	0.657	0.670	0.609	0.570	0.543	0.575	0.568	0.671	**0.674**
MSL	F1	0.819	0.824	0.809	0.822	0.819	0.821	0.762	0.807	0.819	0.818	0.820	0.833	**0.834**
	AUC	0.624	0.550	0.532	0.550	0.602	0.613	0.596	0.601	0.615	0.623	0.590	0.654	**0.725**
	VUS	0.607	0.525	0.518	0.525	0.569	0.599	0.555	0.569	0.580	0.591	0.552	0.632	**0.704**
SMAP	F1	0.733	0.690	0.675	0.741	0.678	0.734	0.682	0.674	0.675	0.694	0.730	0.794	**0.868**
	AUC	0.526	0.450	0.456	0.450	0.452	0.490	0.401	0.380	0.397	0.455	0.474	0.556	**0.571**
	VUS	0.502	0.418	0.450	0.415	0.438	0.482	0.363	0.343	0.373	0.412	0.444	0.541	**0.548**
PTB-XL	F1	0.393	0.351	0.403	0.353	0.323	0.353	0.392	0.264	0.228	0.238	0.366	0.400	**0.405**
	AUC	0.604	0.485	0.583	0.485	0.536	0.560	0.589	0.545	0.516	0.618	0.627	0.639	**0.644**
	VUS	0.456	0.339	0.436	0.339	0.386	0.417	0.453	0.401	0.365	0.471	0.486	0.510	**0.514**
SKAB	F1	0.871	0.820	0.934	0.893	0.933	0.928	0.921	0.829	0.925	0.923	0.922	0.939	**0.941**
	AUC	0.536	0.429	0.493	0.440	0.570	0.496	0.482	0.480	0.504	0.496	0.496	0.575	**0.582**
	VUS	0.535	0.421	0.492	0.434	0.569	0.495	0.482	0.474	0.504	0.496	0.496	0.575	**0.583**
Average	F1	0.700	0.717	0.745	0.738	0.738	0.749	0.741	0.707	0.723	0.729	0.762	0.779	**0.796**
	AUC	0.602	0.515	0.546	0.517	0.573	0.574	0.538	0.518	0.519	0.556	0.553	0.629	**0.649**
	VUS	0.554	0.453	0.501	0.454	0.524	0.533	0.493	0.472	0.473	0.509	0.509	0.586	**0.605**

TABLE II
ABLATION STUDIES ON THE MAMBA ENCODER. THE F1 SCORE IS REPORTED AND THE BEST RESULTS ARE IN **BOLD**.

Mamba Encoder	PSM	MSL	SMAP	PTB-XL	SKAB
✗	0.926	0.822	0.816	0.354	0.938
✓	**0.934**	**0.834**	**0.868**	**0.405**	**0.941**

TABLE III
ABLATION STUDIES ON THE MULTISCALE FACTOR M OF THE MULTISCALE INR CONTINUOUS FUNCTIONS. THE F1 SCORE IS REPORTED AND THE BEST RESULTS ARE IN **BOLD**.

Dataset	PSM	MSL	SMAP	PTB-XL	SKAB
M=1	0.929	0.831	0.803	0.391	0.937
M=2	**0.934**	**0.834**	0.831	**0.405**	**0.941**
M=3	0.930	0.825	**0.868**	0.397	0.940

the highest F1 score on the PSM dataset, it attains the highest ROC-AUC and VUS scores, indicating that it excels in distinguishing between normal and abnormal points even when threshold values are varied. The ROC-AUC and VUS metrics, being threshold-free, provide a more comprehensive evaluation of model performance by considering all possible decision thresholds. In contrast to F1 score, which can be sensitive to a particular threshold choice, these metrics assess the overall ability of the model to discriminate between classes across a broad range of thresholds. Our method's superior performance in both ROC-AUC and VUS scores reflects its robustness and versatility, as it is able to consistently differentiate abnormal data without relying on a specific threshold. This makes it particularly advantageous in real-world anomaly detection tasks, where thresholds are often difficult to determine and may vary across different applications or datasets.

F. Ablation Studies

1) Analysis of the Mamba Encoder: In this section, we conduct an ablation analysis to evaluate the effectiveness of the Mamba encoder in generating INR parameters for anomaly detection tasks, compared to the conventional Transformer encoder. The performance is evaluated using the F1 score, a commonly used metric for assessing the accuracy of anomaly detection models. The results, presented in Table II, clearly indicate that the Mamba encoder consistently outperforms the Transformer encoder. he enhanced performance of the Mamba encoder can be attributed to its superior ability to capture intricate temporal patterns and generate more accurate INR parameters, making it a more effective choice. These findings validate the advantage of the Mamba encoder over conventional Transformer-based architectures, particularly in applications where high-fidelity anomaly detection is essential. In addition to performance improvements, we further analyze the efficiency gains offered by the proposed method, as discussed in Section V-G.

2) Analysis of the Multiscale INR Continuous Functions: Further, we investigate the impact of the designed multiscale INR continuous functions on the model's anomaly detection performance across various datasets. The results of the ablation experiments, as shown in III, indicate that the multiscale strategy significantly enhances the anomaly detection performance. Notably, for the PSM, MSL, PTB-XL, and SKAB datasets, $M = 2$ yields the best performance in terms of F1 score, demonstrating that incorporating two scales effectively captures both fine-grained and coarse temporal patterns in these datasets. In contrast, the results for the SMAP dataset show that $M = 3$ produces the best performance. This suggests that the SMAP dataset may require a finer-grained representation, likely due to the presence of more point anomalies, which are often more challenging to detect and require more detailed temporal information. The third scale provides additional resolution that allows the model to better identify such point anomalies, which may be more difficult to capture with fewer temporal scales. The variation in the optimal value of M across datasets reflects the diverse characteristics of the data. These findings verify the effectiveness of the multiscale INR strategy, showing its potential to improve anomaly detection in time

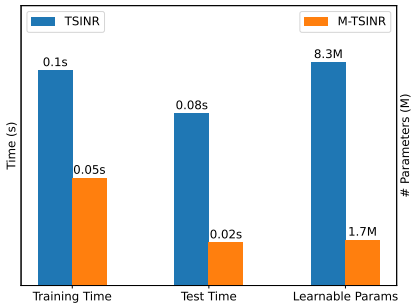

Fig. 3. The efficiency of M-TSINR on the data with 128 batch size compared to TSINR. It is evident that M-TSINR significantly outperforms TSINR in terms of training time, testing time, and learnable parameters, demonstrating its superior efficiency.

series data by providing a more flexible and detailed feature extraction framework.

G. Efficiency Analysis

In this section, we provide a comparative analysis of the efficiency of M-TSINR against the current state-of-the-art INR-based time series anomaly detection method, TSINR. The results, illustrated in the Fig. 3, highlight the significant efficiency improvements achieved by M-TSINR, particularly in terms of both training and testing time, as well as the number of learnable model parameters. Specifically, M-TSINR demonstrates a 50% reduction in training time compared to TSINR, and its testing time is reduced to just one-quarter of that required by TSINR. Additionally, M-TSINR has a considerably smaller model size, with only 1.7 million learnable parameters, compared to 8.3 million parameters in TSINR. These substantial reductions in computational time and model size underscores the efficiency advantage of M-TSINR, making it a more suitable choice for real-time or resource-constrained applications.

The efficiency improvements can be attributed to the combined contributions of the Mamba encoder and the multiscale INR continuous functions. The Mamba encoder, by design, improves the representation of temporal dependencies, leading to more compact and efficient learning of the INR parameters. Furthermore, the multiscale INR approach, by leveraging multiple MLPs at different scales, enables the model to capture the most relevant temporal patterns with fewer parameters and reduced computational overhead. Together, these innovations result in a more efficient model without sacrificing performance, demonstrating the potential of M-TSINR for both improved anomaly detection and enhanced computational efficiency in time series analysis.

VI. Conclusion

Time series anomaly detection is crucial for identifying irregularities in various fields. However, the existing methods face challenges due to the reliance on discrete grid-based representations, which can result in information loss and inadequate modeling of complex temporal patterns. In this paper, we address these challenges by proposing M-TSINR, a time series anomaly detection method that utilizes INR to model continuous functions. Unlike traditional grid-based methods, INR enables us to learn a continuous representation of the time series, preserving fine-grained temporal information and enhancing the model's ability to detect anomalies. To further improve performance, we introduce a multiscale INR framework, allowing the model to capture temporal patterns at different scales. Additionally, we employ the Mamba encoder to efficiently generate INR parameters, optimizing both computational efficiency and memory usage. Experiments show that M-TSINR achieves the superior results across five time series anomaly detection benchmark datasets. Ablation studies further demonstrate the effectiveness of both the Mamba encoder and multiscale INR continuous functions. Efficiency analysis confirms that our method significantly outperforms the current state-of-the-art INR-based anomaly detection methods, making it a more practical and scalable solution for real-world applications.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (62202422 and 62372408), Natural Science Foundation of Shandong Province (ZR2021MH227), and Jinan Science and Technology Bureau plan (202019181).

References

[1] M. Li, P. Peng, H. Sun, M. Wang, and H. Wang, "An order-invariant and interpretable dilated convolution neural network for chemical process fault detection and diagnosis," *IEEE Transactions on Automation Science and Engineering*, pp. 1–11, 2023.

[2] M. Li, P. Peng, J. Zhang, H. Wang, and W. Shen, "Sccam: Supervised contrastive convolutional attention mechanism for ante-hoc interpretable fault diagnosis with limited fault samples," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–12, 2023.

[3] A. Razaque, M. Abenova, M. Alotaibi, B. Alotaibi, H. Alshammari, S. Hariri, and A. Alotaibi, "Anomaly detection paradigm for multivariate time series data mining for healthcare," *Applied Sciences*, vol. 12, no. 17, p. 8902, 2022.

[4] Z. Wang, C. Pei, M. Ma, X. Wang, Z. Li, D. Pei, S. Rajmohan, D. Zhang, Q. Lin, H. Zhang *et al.*, "Revisiting vae for unsupervised time series anomaly detection: A frequency perspective," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 3096–3105.

[5] H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long, "Timesnet: Temporal 2d-variation modeling for general time series analysis," *arXiv preprint arXiv:2210.02186*, 2022.

[6] T. Zhou, P. Niu, L. Sun, R. Jin *et al.*, "One fits all: Power general time series analysis by pretrained lm," *Advances in neural information processing systems*, vol. 36, 2024.

[7] E. Dupont, H. Kim, S. Eslami, D. Rezende, and D. Rosenbaum, "From data to functa: Your data point is a function and you can treat it like one," *arXiv preprint arXiv:2201.12204*, 2022.

[8] K. Liu, F. Liu, H. Wang, N. Ma, J. Bu, and B. Han, "Partition speeds up learning implicit neural representations based on exponential-increase hypothesis," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 5474–5483.

[9] M. Li, K. Liu, H. Chen, J. Bu, H. Wang, and H. Wang, "Tsinr: Capturing temporal continuity via implicit neural representations for time series anomaly detection," *arXiv preprint arXiv:2411.11641*, 2024.

[10] K. Jeong and Y. Shin, "Time-series anomaly detection with implicit neural representation," *arXiv preprint arXiv:2201.11950*, 2022.

[11] N. B. Chikodili, M. D. Abdulmalik, O. A. Abisoye, and S. A. Bashir, "Outlier detection in multivariate time series data using a fusion of k-medoid, standardized euclidean distance and z-score," in *International Conference on Information and Communication Technology and Applications*. Springer, 2020, pp. 259–271.

[12] K. M. Carter and W. W. Streilein, "Probabilistic reasoning for streaming anomaly detection," in *2012 IEEE Statistical Signal Processing Workshop (SSP)*. IEEE, 2012, pp. 377–380.

[13] Z. Zhou and P. Tang, "Improving time series anomaly detection based on exponentially weighted moving average (ewma) of season-trend model residuals," in *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2016, pp. 3414–3417.

[14] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series," 2005.

[15] D. J. Hill and B. S. Minsker, "Anomaly detection in streaming environmental sensor data: A data-driven modeling approach," *Environmental Modelling & Software*, vol. 25, no. 9, pp. 1014–1022, 2010.

[16] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[17] S.-S. Yu, S.-W. Chu, C.-M. Wang, Y.-K. Chan, and T.-C. Chang, "Two improved k-means algorithms," *Applied Soft Computing*, vol. 68, pp. 747–755, 2018.

[18] M. Çelik, F. Dadaşer-Çelik, and A. Ş. Dokuz, "Anomaly detection in temperature data using dbscan algorithm," in *2011 international symposium on innovations in intelligent systems and applications*. IEEE, 2011, pp. 91–95.

[19] L. Cui, Q. Zhang, Y. Shi, L. Yang, Y. Wang, J. Wang, and C. Bai, "A method for satellite time series anomaly detection based on fast-dtw and improved-knn," *Chinese Journal of Aeronautics*, vol. 36, no. 2, pp. 149–159, 2023.

[20] M. M. Breunig, H. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.

[21] F. Liu, K. Ting, and Z. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[22] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," in *Proceedings of the MLSDA 2014 2nd workshop on machine learning for sensory data analysis*, 2014, pp. 4–11.

[23] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2828–2837.

[24] Y. Wang, X. Du, Z. Lu, Q. Duan, and J. Wu, "Improved lstm-based time-series anomaly detection in rail transit operation environments," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 12, pp. 9027–9036, 2022.

[25] T. Ergen and S. S. Kozat, "Unsupervised anomaly detection with lstm neural networks," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 8, pp. 3127–3141, 2019.

[26] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "Deepant: A deep learning approach for unsupervised anomaly detection in time series," *Ieee Access*, vol. 7, pp. 1991–2005, 2018.

[27] A. Zeng, M. Chen, L. Zhang, and Q. Xu, "Are transformers effective for time series forecasting?" in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 9, 2023, pp. 11 121–11 128.

[28] J. Xu, H. Wu, J. Wang, and M. Long, "Anomaly transformer: Time series anomaly detection with association discrepancy," *arXiv preprint arXiv:2110.02642*, 2021.

[29] K. Liu, N. Ma, Z. Wang, J. Gu, J. Bu, and H. Wang, "Implicit neural distance optimization for mesh neural subdivision," in *2023 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2023, pp. 2039–2044.

[30] D. Qin, H. Wang, Z. Liu, H. Xu, S. Zhou, and J. Bu, "Hilbert distillation for cross-dimensionality networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 11 726–11 738, 2022.

[31] L. Mai and F. Liu, "Motion-adjustable neural implicit video representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 10 738–10 747.

[32] J. Guo, K. Liu, J. Yao, Z. Wang, J. Bu, and H. Wang, "Metanerv: Meta neural representations for videos with spatial-temporal guidance," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 3, 2025, pp. 3257–3265.

[33] M. Li, K. Liu, J. Guo, J. Bu, H. Wang, and H. Wang, "Imputeinr: Time series imputation via implicit neural representations for disease diagnosis with missing data," *arXiv preprint arXiv:2505.10856*, 2025.

[34] S. Zhang, K. Liu, J. Gu, X. Cai, Z. Wang, J. Bu, and H. Wang, "Attention beats linear for fast implicit neural representation generation," *arXiv preprint arXiv:2407.15355*, 2024.

[35] X. Ma, P. Zhang, S. Zhang, N. Duan, Y. Hou, M. Zhou, and D. Song, "A tensorized transformer for language modeling," *Advances in neural information processing systems*, vol. 32, 2019.

[36] J. Niu, W. Lu, and G. Penn, "Does bert rediscover a classical nlp pipeline?" in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 3143–3153.

[37] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[38] B. Heo, S. Park, D. Han, and S. Yun, "Rotary position embedding for vision transformer," in *European Conference on Computer Vision*. Springer, 2025, pp. 289–305.

[39] G. Woo, C. Liu, D. Sahoo, A. Kumar, and S. Hoi, "Etsformer: Exponential smoothing transformers for time-series forecasting," *arXiv preprint arXiv:2202.01381*, 2022.

[40] T. Zhou, Z. Ma, Q. Wen, X. Wang, L. Sun, and R. Jin, "Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting," in *International Conference on Machine Learning*. PMLR, 2022, pp. 27 268–27 286.

[41] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[42] X. Liu, C. Zhang, and L. Zhang, "Vision mamba: A comprehensive survey and taxonomy," *arXiv preprint arXiv:2405.04404*, 2024.

[43] Z. Wang, F. Kong, S. Feng, M. Wang, X. Yang, H. Zhao, D. Wang, and Y. Zhang, "Is mamba effective for time series forecasting?" *Neurocomputing*, p. 129178, 2024.

[44] Q. Shen, Z. Wu, X. Yi, P. Zhou, H. Zhang, S. Yan, and X. Wang, "Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction," *arXiv preprint arXiv:2403.18795*, 2024.

[45] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *arXiv preprint arXiv:2111.00396*, 2021.

[46] A. Abdulaal, Z. Liu, and T. Lancewicki, "Practical approach to asynchronous multivariate time series anomaly detection and localization," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 2485–2494.

[47] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 387–395.

[48] P. Wagner, N. Strodthoff, R. Bousseljot, D. Kreiseler, F. I. Lunze, W. Samek, and T. Schaeffter, "Ptb-xl, a large publicly available electrocardiography dataset," *Scientific data*, vol. 7, no. 1, pp. 1–15, 2020.

[49] I. D. Katser and V. O. Kozitsin, "Skoltech anomaly benchmark (skab)," https://www.kaggle.com/dsv/1693952, 2020.

[50] T. Zhang, Y. Zhang, W. Cao, J. Bian, X. Yi, S. Zheng, and J. Li, "Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures," *arXiv preprint arXiv:2207.01186*, 2022.

[51] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 419–22 430, 2021.

[52] S. Liu, H. Yu, C. Liao, J. Li, W. Lin, A. X. Liu, and S. Dustdar, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *International conference on learning representations*, 2021.

[53] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 12, 2021, pp. 11 106–11 115.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[55] J. Paparrizos, P. Boniol, T. Palpanas, R. S. Tsay, A. Elmore, and M. J. Franklin, "Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection," *Proceedings of the VLDB Endowment*, vol. 15, no. 11, pp. 2774–2787, 2022.

[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.