Efficient Anomaly Detection of Irregular Sequences in Ct-Echo Model Space

Ao Chen, Xiren Zhou*, Huanhuan Chen*

School of Computer Science and Technology, University of Science and Technology of China
chenao57@mail.ustc.edu.cn, zhou0612@ustc.edu.cn, hchen@ustc.edu.cn

Abstract

Efficient anomaly detection of irregular sequences, especially those characterized by non-uniform sampling from discontinuous operations or unreliable sensors, presents challenges across various fields. In response, this paper introduces irregular-sequence classification in "Ct-Echo Model Space". A novel Continuous-time Echo Network (Ct-Echo) is proposed to fit irregular sequences, efficiently capturing their inherent dynamic characteristics. Ct-Echo utilizes the "Echo" mechanism, where history information influences the current state and diminishes over time, and employs Ordinary Differential Equation (ODE) to construct continuous-time transition of hidden states. Each sequence is individually fitted via Ct-Echo to derive a readout model. These fitted models, capturing the dynamic characteristics of the original data, serve as representations of the corresponding sequences, thus mapping the original data from the data space to the Ct-Echo model space. Anomaly detection is further performed in this model space, evaluating differences between models rather than directly on the original sequences. Our method enhances real-time processing and lessens reliance on the amount of labeled training data, as demonstrated by experimental studies.

Introduction

Anomaly detection of sequential data typically involves determining whether a collected sequence is normal or belongs to a specific type of anomaly. Traditional techniques for anomaly detection often presuppose consistent and regular sampling intervals, an assumption that is not universally applicable. For example, sensor data collected from mechanical equipment could exhibit irregular sampling or missing values, stemming from fluctuating operational conditions or sensor malfunctions. Such irregular sequences, characterized by observations at non-uniform intervals, present significant analytical challenges (Chowdhury et al. 2023).

Common methods address such irregularities through resampling or interpolation (Sterne et al. 2009), leading to information loss and potential distortion of original signals. Recurrent Neural Networks (RNNs) have also been introduced into such tasks (Che et al. 2018), but tend to be computationally intensive, limiting their applicability in time-sensitive scenarios. Besides, ODE-based methods, such as ODE-RNN (Rubanova, Chen, and Duvenaud 2019) and Neural Controlled Differential Equations (Neural CDE) (Kidger et al. 2020; Jhin, Lee, and Park 2023), have proven effective for handling irregular sequences through continuous-time networks. Nevertheless, existing ODE-based methods grapple with high computational complexity and extensive network parameters, hindering their applicability in real-time or data-scarce tasks.

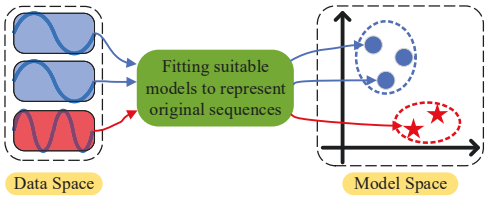


Figure 1: The MBM represents each data instance with a model that is suitably fitted to capture the dynamical characteristic inherent in the original data instance, which further serves as the data representation for further classification.

Recently, Model-Based Methods (MBMs) have emerged as promising solutions for anomaly detection in sequential data (Chen et al. 2023). MBMs, as Figure 1, represent each data instance through a suitably fitted model that captures the data-inherent dynamic characteristics. This representation marks a transformation from data-space processing to model-space analysis. Representing original data with dynamic-captured models enables the application of learning algorithms to the models rather than the original data (Chen et al. 2013). MBMs have proven computational efficiency and reduced data dependency across various time-series classification and anomaly detection tasks (Chen, Zhou, and Chen 2024; Liu, Zhou, and Chen 2024). However, existing models (e.g. Echo-State-Network(ESN)-based) used for data fitting and representation maintain a uniform approach to state updates across time steps, failing to handle data with irregular sampling or missing values.

Addressing the above considerations, this paper proposes to conduct anomaly detection of irregular sequences within "Ct-Echo Model Space". The Continuous-Time Echo Network (Ct-Echo) is introduced to fit irregular sequences, efficiently capturing the data-inherent dynamic characteris-

*Corresponding Authors: Xiren Zhou, Huanhuan Chen.

tics. Ct-Echo employs the "Echo" mechanism, where historical information resonates like an echo, influencing the current state and diminishing gradually over time. This ensures effective capture of the dynamic characteristics, that is, the temporal dependencies within each sequence. Moreover, Ct-Echo incorporates the Ordinary Differential Equation (ODE) for continuous-time updates of hidden states at irregular time intervals, constructing a continuous iteration process in the hidden layer and effectively modeling irregular sequences. Fitting with Ct-Echo is accomplished through ridge regression, eliminating the need for offline iterative training such as gradient descent. Consequently, Ct-Echo not only effectively captures dynamic characteristics in irregular sequences but also demonstrates high fitting efficiency.

Fitting irregular sequences via Ct-Echo, each sequence is represented with a fitted model respectively[1]. The distance metric between models is further introduced, well reflecting the difference of dynamic characteristics inherent in the original sequences. Through the above, the original sequences are mapped from the data space to the Ct-Echo model space. The Ct-Echo effectively captures the dynamic characteristics of irregular sequences into fitted models, leading to closer models for sequences within the same category and more pronounced divergence for sequences across different categories due to their unique dynamics. Classification algorithms are efficiently operated on the models, allowing for the efficient anomaly detection of the sequences represented by these fitted models. The main contributions of this paper are presented as follows:

- We propose Ct-Echo, effective in capturing the dynamic characteristics of irregular sequences. It uses the "Echo" mechanism to capture temporal dependencies and ODE for continuous-time transitions of hidden states. By representing data with fitted models, further processing is conducted on the models instead of the original data.
- A notable advantage of the proposed method is its computational efficiency. Fitting with Ct-Echo is accomplished through ridge regression, followed by the readily available distance metric between fitted models. This efficiency is critical in scenarios requiring rapid data processing and immediate analysis.
- Focusing and adequately capturing the data-inherent dynamic characteristics, our method reduces dependence on the amount of labeled training data. This attribute is beneficial in situations where labeled data is limited or where collecting large datasets is not feasible, broadening its applicability across various fields.

Related Work

Analysis of Irregular Sequences

Most studies addressing sequence analysis assume uniform sampling, but in reality, data could be irregularly sampled due to inconsistent data collection or missing values. This irregularity poses challenges for methods originally developed for regularly sampled data (Chowdhury et al. 2023).

Considerable studies have contributed to the analysis of irregular sequences. Che et al. (2018) proposed GRU-D, which incorporates missing patterns into Gated Recurrent Unit (GRU). Besides, transformer-based methods, such as Multi-Time Attention Networks (mTANs) (Shukla and Marlin 2021) and Warpformer (Zhang et al. 2023), have been proposed to tackle this. Recent advancements have seen the integration of ODEs in modeling irregular sequences. Rubanova et al. (2019) introduced ODE-RNN for capturing continuous-time hidden dynamics. Furthermore, Kidger et al. (2020) extended this concept by incorporating CDEs into the Neural ODE (Chen et al. 2018) framework. Yuan et al. (2023) further proposed the Ordinary Differential Equation Recurrent State Space Model (ODE-RSSM). However, the efficacy of these learning methods diminishes when training data is scarce, primarily due to their extensive parameterization. Furthermore, the significant computational requirements associated with these methods result in extended training times and increased computational resources.

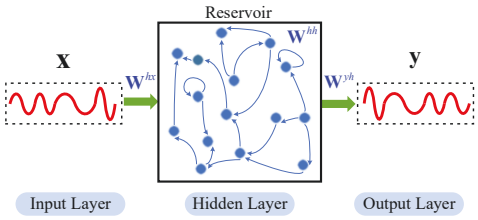


Figure 2: Illustration of ESN structure, which consists of an input layer, a hidden layer, and an output layer.

A Brief Introduction to ESN

ESN (Jaeger 2001) (Figure 2) is a form of RNN optimized for processing sequence, constructed by three parts: input layer, hidden layer (reservoir), and output layer. The distinctive feature of ESN lies in the reservoir, which is a large, randomly generated, fixed recurrent network. In MBMs, fitting data with ESN involves sequentially feeding the input sequence into the reservoir, acquiring the corresponding hidden states, and finally solving only the output weight $\mathbf{W}^{yh}$ (Chen et al. 2013), while the input weight $\mathbf{W}^{hx}$ and the reservoir weight $\mathbf{W}^{hh}$ are randomly generated and fixed under Echo State Property (ESP) (Jaeger 2001). This streamlines the fitting process and enhances the efficiency.

Model-based Methods

MBMs[2] start by fitting each sequence into a suitable model that captures its dynamic characteristics (i.e. changing rules). These models then served as more stable and parsimonious representations of the corresponding sequences. Thus, learning methods are utilized on the fitted models instead of the original data. In (Chen et al. 2013), temporal signals are segmented, fitted by Cycle topology with Regular Jumps (CRJ) (Rodan and Tiňo 2012), represented by

[1]In this paper, "Ct-Echo" refers to the network used to fit irregular sequences, yielding the fitted "Ct-Echo readout model" for sequence representation, also short-termed as "'Ct-Echo model".

[2]Also denoted by "learning in the model space"

the obtained readout models that are further efficiently classified. Extended from this, MBMs have proven effective in various domains, such as the fault diagnosis of the Barcelona water network (Quevedo et al. 2014) and diverse sequence classification tasks (Gong et al. 2018; Ma et al. 2020; Chen, Zhou, and Chen 2024). Besides, MBMs have also been employed for underground diagnosis by considering two directions' dynamic characteristics (Chen et al. 2023; Liu, Zhou, and Chen 2024). While existing studies show MBMs' effectiveness in sequential data analysis, current MBMs use ESN-based methods for data fitting. ESN, with uniform state updates across time steps, is not suited for irregular sequence fitting and dynamic capturing.

Methodology

This section outlines the irregular sequence classification in the Ct-Echo model space, unfolding in three parts:

- **Capture Dynamic Characteristics via Ct-Echo:** This initiates our method by fitting sequences through Ct-Echo, representing each sequence by a fitted model.
- **Distance Metric between Ct-Echo Models:** The next part introduces a reasonable distance metric to measure differences between Ct-Echo models.
- **Anomaly Detection in the Ct-Echo Model Space:** Representing the sequences with fitted models, supported with the distance metric between models, anomaly detection is implemented in the Ct-Echo model space.

Capture Dynamic Characteristics via Ct-Echo

Similar to ESN, Ct-Echo is composed of: the **Input layer**, the **Hidden layer** that consists of a reservoir subjected to continuous-time transitions facilitated by ODE, and the **Output layer**. In this structure, 1) an irregular sequence is sent into the input layer, 2) and then iterated in the hidden layer considering both the history and current information to generate continuous-time hidden states, 3) resulting in output through the output layer.

Among them, the reservoir in the hidden layer is a recurrent network with parameters randomly generated and fixed under the echo state property, ensuring that history information influences the current state and gradually diminishes over time. Besides, the use of ODE in the hidden layer allows for the integration of hidden states at any time point. Therefore, Ct-Echo gives a continuous-time transition of hidden states under irregular time intervals, beneficial to capturing the dynamic characteristics of irregular sequences. The Ct-Echo is adapted into two variants:

- **Ct-Echo-Decay:** The linear ODE, independent of the input sequence, is used to construct the continuous-time transition of hidden states.
- **Ct-Echo-Leaky:** The leaky integrator, considering the input sequence, is integrated into the reservoir to process continuous-time hidden states.

Ct-Echo-Decay In the first variant of Ct-Echo, the ODE processes hidden states independently of the input[3] $\mathbf{x}$. As shown in Figure 3(a), the input $\mathbf{x}$ is sent into the hidden layer sequentially through the input layer. Ct-Echo operates in the hidden layer according to the following iterative formula:

$$\begin{aligned}\mathbf{h}_m(t_n) &= \mathbf{ODESolver}(f_\theta(\mathbf{h}), \mathbf{h}(t_{n-1}), (t_{n-1}, t_n)),\\ \mathbf{h}(t_n) &= \tanh(\mathbf{W}^{hh} \cdot \mathbf{h}_m(t_n) + \mathbf{W}^{hx} \cdot \mathbf{x}(t_n)),\end{aligned} \tag{1}$$

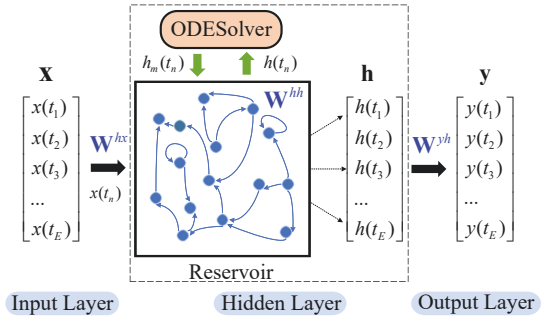
where $\mathbf{h}_m(t_n)$ represents the intermediate hidden state at time t_n, **ODESolver** is a solver for ODE, f_θ denotes the ODE governed by parameters θ, $\mathbf{h}(t_{n-1})$ is the previous hidden state (initial value of **ODESolver**), (t_{n-1}, t_n) is the time interval, $\mathbf{h}(t_n)$ is the updated hidden state, $\mathbf{W}^{hh}$ and $\mathbf{W}^{hx}$ are the weights for hidden layer (reservoir) and input layer, respectively, and $\mathbf{x}(t_n)$ is the data point at time t_n.

During the iteration (Figure 3(b)) in Ct-Echo-Decay, the previous hidden state is continuously transitioned to the current step as an intermediate hidden state using the ODE. This intermediate hidden state, combined with the current input value, then determines the current hidden state. As iterations progress, the above process persists, linking each data point to prior ones, thus effectively preserving the dynamic characteristics within the sequence.

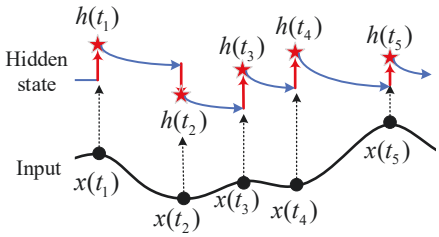
In this paper, we choose the linear ODE function as f_θ within the ODESolver in Formula (1), given as

$$\frac{d\mathbf{h}}{dt} = f_\theta(\mathbf{h}) = \tau \cdot \mathbf{h}, \tag{2}$$

where $\tau < 0$ is the decay rate. This ODE is solvable analytically, and its solution describes an exponential decay pro-



(a) Structure of Ct-Echo-Decay



(b) Flow of Ct-Echo-Decay in hidden layer

Figure 3: Illustration of Ct-Echo-Decay. (a) is the structure, consisting of an input layer, a hidden layer, and an output layer. It uses ODEs in the hidden layer to help the reservoir establish continuous-time hidden states. (b) is the processing flow in the hidden layer. The blue line is the continuous-time transition achieved by ODE, and the red line is the hidden state computation in the reservoir.

[3]An irregular sequence is $\mathbf{x} = [x(t_1), x(t_2), \ldots, x(t_E)]$, t is the sampling time, and E is the number of sampling time points.

Algorithm 1: Ct-Echo-Decay

Input: Sequence $\mathbf{x}$, sampling time t, number of sampling points E, input and hidden weights $\mathbf{W}^{hx}$ and $\mathbf{W}^{hh}$, function f_θ.
Output: The output weights $\{\mathbf{W}^{yh}, \mathbf{b}\}$

Initialize $\mathbf{h}(t_1) = \tanh(\mathbf{W}^{hx} \cdot \mathbf{x}(t_1))$
for $n = 2$ to E **do**
  $\mathbf{h}_m(t_n) = \mathbf{ODESolver}(f_\theta(\mathbf{h}), \mathbf{h}(t_{n-1}), (t_{n-1}, t_n))$,
  $\mathbf{h}(t_n) = \tanh(\mathbf{W}^{hh} \cdot \mathbf{h}_m(t_n) + \mathbf{W}^{hx} \cdot \mathbf{x}(t_n))$,
Compute weights of $\mathbf{x}(t+1) = \mathbf{W}^{yh}\mathbf{h}(t) + \mathbf{b}$ on observed timestep through Ridge Regression
return $\{\mathbf{W}^{yh}, \mathbf{b}\}$

Algorithm 2: Ct-Echo-Leaky

Input: Sequence $\mathbf{x}$, sampling time t, number of sampling points E, input weight $\mathbf{W}^{hx}$, hidden weight $\mathbf{W}^{hh}$, function f_θ.
Output: The output weights $\{\mathbf{W}^{yh}, \mathbf{b}\}$

Initialize $\mathbf{h}(t_0) = \tilde{\mathbf{0}}$
$\mathbf{x} \leftarrow \mathbf{CubicSplineInterpolation}(\mathbf{x})$
for $n = 1$ to E **do**
  $\mathbf{h}_m(t_n) = \mathbf{ODESolver}(f_\theta(\mathbf{h}, \mathbf{x}), \mathbf{h}(t_{n-1}), (t_{n-1}, t_n))$,
  $\mathbf{h}(t_n) = \tanh(\mathbf{h}_m(t_n))$,
Compute weights of $\mathbf{x}(t+1) = \mathbf{W}^{yh}\mathbf{h}(t) + \mathbf{b}$ on observed timestep through Ridge Regression
return $\{\mathbf{W}^{yh}, \mathbf{b}\}$

cess. The analytical solution with an initial value $\mathbf{h}(t_{n-1})$ at any time t is given by

$$\mathbf{h}(t_{n-1}) \cdot e^{\tau \cdot (t - t_{n-1})}. \tag{3}$$

Therefore, the intermediate state $\mathbf{h}_m(t_n)$ in Formula (1) is given by value at t_n:

$$\mathbf{h}_m(t_n) = \mathbf{h}(t_{n-1}) \cdot e^{\tau \cdot (t_n - t_{n-1})}. \tag{4}$$

Consequently, the decay mechanism leads to the gradual fading of the previous information in the hidden states, which is modulated by the variability of the time intervals. The pseudocode of the fitting process of a sequence $\mathbf{x}$ using Ct-Echo-Decay is presented in Algorithm 1.

Ct-Echo-Leaky The second variant of Ct-Echo incorporates the ODE into the reservoir, making the ODE directly consider input $\mathbf{x}$. As shown in Figure 4(a), sending the input $\mathbf{x}$ into the hidden layer through the input layer sequentially, the iterative process in Ct-Echo-Leaky is defined as follows:

$$\begin{aligned} &\mathbf{h}_m(t_n) = \mathbf{ODESolver}(f_\theta(\mathbf{h}, \mathbf{x}), \mathbf{h}(t_{n-1}), (t_{n-1}, t_n)), \\ &\mathbf{h}(t_n) = \tanh(\mathbf{h}_m(t_n)), \end{aligned} \tag{5}$$

where $\mathbf{h}_m(t_n)$ is the intermediate state computed by the ODE solver[4], which integrates the ODE function over the time interval (t_{n-1}, t_n). The hidden state $\mathbf{h}(t_n)$ is obtained by applying the $\tanh$ function.

According to the iteration (Figure 4(b)) of Ct-Echo-Leaky, the input sequence is first represented by a spline curve[5]. Subsequently, the ODE integrates considering both the input and proceeding hidden states to generate a continuous hidden process, giving hidden states at any time step.

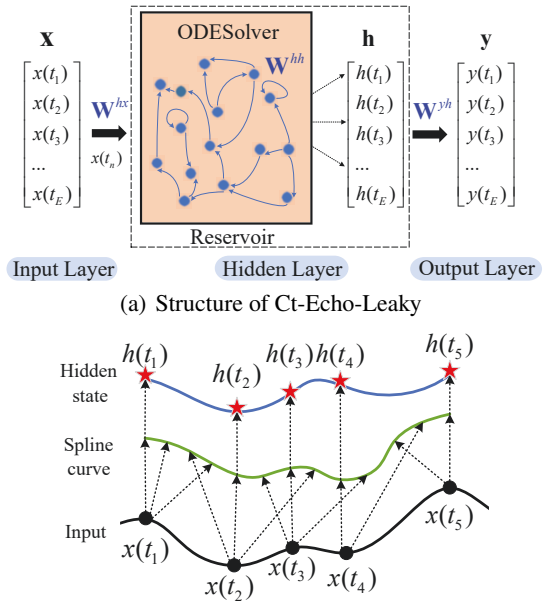
Notebly in Ct-Echo-Leaky, we utilize the leaky integrator (Jaeger et al. 2007) to model irregular sequences, given as:

$$\frac{d\mathbf{h}}{dt} = f_\theta(\mathbf{h}, \mathbf{x}) = -a \cdot \mathbf{h} + \tanh(\mathbf{W}^{hh} \cdot \mathbf{h} + \mathbf{W}^{hx} \cdot \mathbf{x}), \tag{6}$$

where a acts as the leaky rate, and the $\tanh$ function captures the non-linear interactions between the hidden state $\mathbf{h}$ and the input $\mathbf{x}$. This design ensures a balanced consideration of both inputs and hidden states, useful for modeling time-dependent processes. The pseudocode of Ct-Echo-Leaky is presented in Algorithm 2.

Fitting Sequence by "Next-Step Prediction" To fit the sequence and capture its dynamic characteristics via Ct-Echo, we adopt the "next-step prediction" approach (Chen et al. 2015). This approach predicts the next step value using current and past values in the series, effectively capturing the dynamic characteristics of the data and obtaining the readout model as the representation of the original data.

Specifically, 1) a sequence $\mathbf{x}$ is sequentially sent into the hidden layer through the input layer. 2) $\mathbf{x}$ is then iterated in the hidden layer to obtain the hidden states $\mathbf{h}$. During the iteration process through Formula (1) or (5), the hidden state



(a) Structure of Ct-Echo-Leaky

(b) Flow of Ct-Echo-Leaky in hidden layer

Figure 4: Illustration of Ct-Echo-Leaky. (a) is the structure, consisting of an input layer, a hidden layer, and an output layer. It integrates ODEs into the reservoir in the hidden layer to establish continuous-time hidden states. (b) is the processing flow in the hidden layer. The green line is the spline curve achieved through the input, and the blue line is the continuous-time hidden state obtained in the reservoir.

[4]In this variant, the ODE complicates the integration process, requiring numerical solution methods. Runge-Kutta (4,5) is utilized in this paper (Dormand and Prince 1980).

[5]To handle irregularly sampled input data $\mathbf{x}$, cubic spline interpolation (De Boor 1978) is employed.

at any time is influenced by the current input value as well as proceeding hidden states, and the use of ODE ensures continuous-time transitions of hidden states across irregular time intervals, thus effectively retaining the dynamic characteristics within the sequence. 3) Finally, the mapping between the predicted value $\mathbf{y}$ and the hidden state from the previous time step is established, given as:

$$\mathbf{y}(\mathbf{h}) = \mathbf{W}^{yh}\mathbf{h} + \mathbf{b}, \tag{7}$$

where the output weights $\mathbf{W}^{yh}$ and the bias $\mathbf{b}$ can be directly determined by extending the ridge regression solution:

$$\tilde{\mathbf{W}} = (\tilde{\mathbf{H}}^T\tilde{\mathbf{H}} + \lambda\mathbf{I})^{-1}\tilde{\mathbf{H}}^T\mathbf{X}, \tag{8}$$

where $\tilde{\mathbf{W}}$ is the concat of the output weights $\mathbf{W}^{yh}$ and the bias $\mathbf{b}$, $\tilde{\mathbf{H}}$ is the augmented matrix of hidden states with additional all-ones columns. $\mathbf{X}$ represents the target values, which are the data points at the next time step corresponding to each hidden state. λ is the regularization parameter, and $\mathbf{I}$ is the identity matrix.

The output weights and bias can be then extracted from $\tilde{\mathbf{W}}$, obtaining the Ct-Echo readout model as Formula (7). This transformation allows for a concise representation of each sequence with the readout model.

Distance Metric between Ct-Echo Models

Representing each sequence with a fitted model, the distance between the Ct-Echo readout models should be defined to support further classification of models (Chen et al. 2013).

Suppose two distinct Ct-Echo readout models, $\mathbf{y}_1$ and $\mathbf{y}_2$ mapping from $\mathbb{R}^N$ (hidden layer dimension) to $\mathbb{R}^M$ (output layer dimension), are formulated as:

$$\begin{aligned} \mathbf{y}_1(\mathbf{h}) &= \mathbf{W}_1^{yh}\mathbf{h} + \mathbf{b}_1, \\ \mathbf{y}_2(\mathbf{h}) &= \mathbf{W}_2^{yh}\mathbf{h} + \mathbf{b}_2. \end{aligned} \tag{9}$$

The definition of the 2-norm distance (Chen et al. 2013) between $\mathbf{y}_1(\mathbf{h})$ and $\mathbf{y}_2(\mathbf{h})$ is given as:

$$L_2(\mathbf{y}_1, \mathbf{y}_2) = \left(\int_C ||\mathbf{y}_1(\mathbf{h}) - \mathbf{y}_2(\mathbf{h})||^2 d\mathbf{h} \right)^{1/2}, \tag{10}$$

where $C = [-1, 1]^N$ is the domain of integration[6].

By integrating Formula (9) into (10), the distance between two Ct-Echo readout models could be estimated by:

$$Dis(\mathbf{y}_1, \mathbf{y}_2) = \frac{1}{3}\sum_{j=1}^{N}\sum_{i=1}^{M}\omega_{i,j}^2 + ||\mathbf{b}'||^2, \tag{11}$$

where $\omega_{i,j}$ is the (i, j) element of $\mathbf{W}' = \mathbf{W}_1^{yh} - \mathbf{W}_2^{yh}$ and $\mathbf{b}' = \mathbf{b}_1 - \mathbf{b}_2$. Utilizing Formula (11), the distance between any two Ct-Echo readout models could be effectively measured, thus distance-based learning algorithms could be applied in the model space for further analysis.

[6]According to the iterative formula (1) and (5), hidden states $\mathbf{h}$ are assumed to lie within the hypercube $[-1, 1]^N$.

Anomaly Detection in the Ct-Echo Model Space

Anomaly detection involves classifying unknown irregular sequences as either normal or specific types of anomalies. Our approach consists of two phases: training and detection.

Training phase Each labeled irregular sequence is first fitted through Ct-Echo separately to obtain a corresponding readout model. The fitted model represents the original data, mapping the corresponding sequence into the Ct-Echo model space supported by the distance metric given in the above subsection. A classifier, such as Support Vector Machine (SVM) (Cortes and Vapnik 1995) or K-Nearest Neighbors (KNN) (Altman 1992), is then trained on these models.

Detection phase A newly collected irregular sequence is first fitted with Ct-Echo, obtaining the readout model. This model is then evaluated using the previously trained classifier to determine whether it is normal or belongs to a specific type of anomaly, which directly corresponds to the category of the original sequence.

Experimental Study

This section details experiments tested on several datasets, followed by analysis of comparative experimental results and the presentation of additional analytical experiments.

Introduction of the Utilized Datasets

Three datasets, **CWRU**, **SU**, and **WHU** are utilized, introduced as follows[7]. For the default setting: **1) Assume sequences with irregular sampling**, we randomly delete 50% data points from each sequence (i.e., with the missing rate of 50%)[8]; **2) Supposing labeled-data-limited scenarios**, for training, we use only 200 sequences per sub-dataset for CWRU and SU, and 90 for WHU; for testing, we use 1800 sequences per sub-dataset for CWRU, 4800 per sub-dataset for SU, and 90 for WHU.

- **CWRU Dataset (Loparo 2012):** Comprises 2-dimensional vibration signals from bearings under four load conditions (sub-datasets A, B, C, D), each with 200 instances per category. A mixed dataset is denoted as E. The dataset, with 2048 time steps per instance, supports a 10-category classification task (one normal category and nine distinct anomaly categories).
- **SU Dataset (Shao et al. 2019):** Contains 8-dimensional data for gearbox and bearing under two working conditions, denoted as sub-datasets G20, G30 for gearboxes, and B20, B30 for bearings, each with 1000 instances per category. The dataset includes four faults and one normal category, and each sample consists of 1024 time steps.
- **WHU Dataset (Liu et al. 2019):** Includes 1-dimensional vibration data from a rotor system, collected at 1200 revolutions per minute and a sampling frequency of 2048

[7]The experiments are conducted on a PC with CPU: Intel(R) Core(TM) i9-13900k, and GPU: NVIDIA GeForce RTX 3080.

[8]Experiments and discussions under different missing rates are subsequently presented, where the missing rate refers to the proportion of data points removed from each sequence.

	CWRU					SU				WHU
	A	B	C	D	E	G20	G30	B20	B30	
NFFT-SVM	32.56 ± 1.20	23.28 ± 1.18	21.83 ± 1.33	22.67 ± 1.25	18.74 ± 1.07	84.06 ± 1.67	69.19 ± 1.35	79.10 ± 1.23	64.44 ± 1.91	34.09 ± 2.90
GRU-Δt	70.69 ± 4.75	63.56 ± 2.80	58.85 ± 4.71	71.38 ± 3.94	55.63 ± 2.97	$\mathbf{97.39 \pm 0.82}$	94.77 ± 1.72	96.95 ± 1.03	97.49 ± 1.10	67.11 ± 1.31
GRU-Int	33.63 ± 4.57	21.31 ± 5.43	27.94 ± 5.68	43.69 ± 4.21	34.94 ± 6.55	93.02 ± 1.57	83.86 ± 1.40	90.70 ± 2.28	90.51 ± 2.65	81.58 ± 3.57
mTANs	13.33 ± 1.35	13.38 ± 0.96	12.79 ± 1.41	13.06 ± 1.03	10.78 ± 1.22	84.56 ± 1.34	64.73 ± 1.58	75.25 ± 1.11	59.52 ± 1.89	32.23 ± 4.23
Warpformer	81.94 ± 1.13	64.67 ± 1.58	67.11 ± 1.76	80.22 ± 1.07	61.60 ± 1.66	90.31 ± 1.73	82.04 ± 2.13	97.56 ± 1.11	97.50 ± 1.05	93.33 ± 2.11
ODE-RNN	20.25 ± 2.25	31.25 ± 3.88	26.25 ± 2.75	23.75 ± 2.51	27.87 ± 2.18	44.40 ± 1.31	25.20 ± 1.27	26.35 ± 3.39	31.85 ± 3.77	23.68 ± 0.23
Neural CDE	10.22 ± 0.25	10.09 ± 0.27	11.13 ± 0.21	10.58 ± 0.25	10.34 ± 0.22	19.37 ± 0.26	19.16 ± 0.28	20.83 ± 0.21	21.22 ± 0.32	25.34 ± 1.54
MBM-Int	61.22 ± 1.45	67.67 ± 1.05	68.89 ± 0.99	82.22 ± 0.84	65.13 ± 0.62	73.83 ± 1.58	83.67 ± 1.27	96.08 ± 1.32	95.69 ± 1.01	95.56 ± 2.19
MBM-Δt	66.91 ± 1.01	64.83 ± 0.98	67.23 ± 1.02	81.00 ± 0.72	61.61 ± 0.82	91.13 ± 1.30	96.44 ± 0.50	98.30 ± 0.29	98.14 ± 0.26	91.93 ± 2.85
Proposed (Decay)	75.22 ± 2.10	75.78 ± 1.66	80.72 ± 2.36	86.67 ± 1.22	70.09 ± 1.10	94.31 ± 0.96	$\mathbf{98.15 \pm 0.55}$	99.21 ± 0.28	$\mathbf{99.69 \pm 0.13}$	95.56 ± 2.03
Proposed (Leaky)	$\mathbf{83.06 \pm 0.88}$	$\mathbf{82.83 \pm 1.06}$	$\mathbf{87.67 \pm 1.20}$	$\mathbf{92.67 \pm 0.81}$	$\mathbf{78.27 \pm 0.65}$	91.25 ± 0.58	97.10 ± 0.58	$\mathbf{99.50 \pm 0.11}$	$\mathbf{99.69 \pm 0.09}$	$\mathbf{96.67 \pm 1.87}$

Table 1: Accuracy(%) (mean±std across five runs) of Classification on Multiple Datasets with 50% Missing Rate

Hz. Each instance is 2048 time steps long, with 45 instances per category. It encompasses four categories: normal, contact-rubbing, unbalanced, and misalignment.

The Evaluated Methods

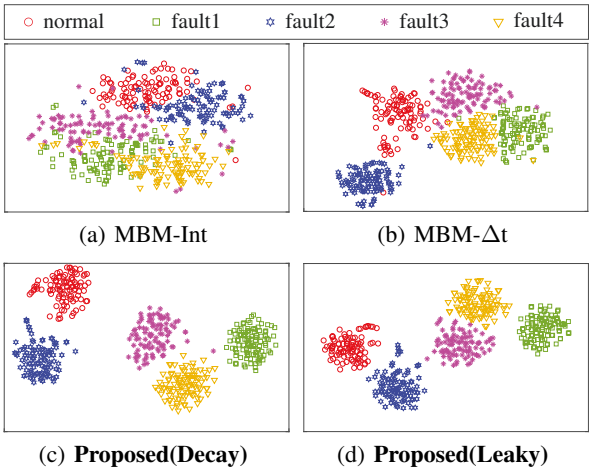
Addressing irregular sequence, we employ **nine** baseline methods: 1) Feature extraction with traditional machine learning: **NFFT-SVM** utilizes Nonequispaced Fast Fourier Transforms (NFFT) (Keiner, Kunis, and Potts 2009) for feature extraction from irregular sequence data, followed by classification using SVM. 2) RNN-based methods: **GRU-Δt** enhances GRU by incorporating the time intervals between data points as additional input; **GRU-Int** enhances GRU by interpolating irregular sequence data to regular intervals. 3) Transformer-based methods: **mTANs** (Shukla and Marlin 2021) and **Warpformer** (Zhang et al. 2023). 4) ODE-based methods: **ODE-RNN** (Rubanova, Chen, and Duvenaud 2019) and **Neural CDE** (Kidger et al. 2020). 5) MBM methods: **MBM-Int** involves interpolation of irregular data, followed by SVM classification in model space; **MBM-Δt** incorporates the time intervals between data points as additional input, followed by SVM classification in model space.

Two variants of the proposed method[9] are denoted as: **1) Proposed(Decay)** employs Ct-Echo-Decay for fitting irregular sequences; **2) Proposed(Leaky)** employs Ct-Echo-Leaky for fitting irregular sequences. Both methods choose SVM for the classification in the Ct-Echo model space.

Experimental Results and Disscussions

Results in Table 1 show that while NFFT-SVM and MBMs exhibit some capability when handling irregular sequences, they lack overall accuracy due to their reliance on discrete representations, which fail to capture the continuous-time dynamic characteristics. Deep learning methods encounter limitations in data-scarce environments due to their reliance on comprehensive and diverse datasets for optimization. Specifically, mTANs, ODE-RNN, and Neural CDE exhibit underperformance. Warpformer, GRU-Δt, and GRU-Int, although comparatively more effective, still fall short of ours.

[9]In our experiments, the spectral radius and size of the reservoir are set to 0.8 and 10 respectively. The decay rate of Proposed(Decay) is -0.05 and the leaky rate of Proposed(Leaky) is 1.



(a) MBM-Int (b) MBM-Δt

(c) **Proposed(Decay)** (d) **Proposed(Leaky)**

Figure 5: t-SNE visualizations of the B30 dataset after processing by two MBMs and two proposed methods. Each subplot represents the data clustered into five categories, showcasing the separation achieved by (a) MBM-Int, (b) MBM-Δt, (c) **Proposed(Decay)**, and (d) **Proposed(Leaky)**.

The proposed methods show considerable results, with the Proposed(Leaky) demonstrating overall superior performance. As Figure 5, the proposed methods exhibit better clustering across different categories compared to those that solely rely on interpolation or intervals. The results stem from: 1) We utilize "Echo" to preserve the temporal dependencies within each sequence and incorporate ODE within the hidden layer to manage continuous-time transitions between irregular data points, effectively and efficiently capturing dynamic characteristics within irregular sequences; 2) Capturing sequence-inherent dynamic characteristics, Ct-Echo facilitates the transformation of irregular sequences from data space to Ct-Echo model space, enabling more efficient classification with limited data.

Discussion about Training Time We evaluate the training times of five better-performed methods (GRU-Δt, GRU-Int, Warpformer, MBM-Int, and MBM-Δt) with the Proposed(Decay) and Proposed(Leaky). Table 2 illustrates the

	GRU -Δt	GRU -Int	Warp former	MBM -Δt	MBM -Int	**Proposed (Decay)**	**Proposed (Leaky)**
CWRU	76	132	55	2.4	1.1	**0.2**	13
SU	53	99	67	1.7	0.8	**0.1**	7.3
WHU	26	52	15	0.8	0.4	<**0.1**	2

Table 2: Average Training Time(s) of Better-performed Methods on Multiple Datasets with 50% Missing Rate

	Missing Rate	GRU-Δt	GRU-Int	Warp former	MBM-Int	MBM-Δt	**Proposed (Decay)**	**Proposed (Leaky)**
A	10%	85.03	90.56	92.83	95.89	94.53	**98.39**	97.67
	20%	81.25	81.12	94.50	92.22	90.27	**94.96**	94.33
	30%	77.94	73.11	85.67	85.17	85.40	91.33	**92.33**
	40%	73.62	62.37	82.17	78.67	76.57	82.00	**85.78**
	50%	70.69	33.63	81.94	61.22	66.91	75.22	**83.06**
B	10%	82.63	75.56	91.06	97.78	97.24	98.33	**98.94**
	20%	79.75	67.75	90.56	93.22	94.50	96.61	**96.83**
	30%	70.63	44.50	84.06	85.28	88.13	93.56	**94.06**
	40%	56.06	49.18	75.28	80.56	78.12	85.17	**88.39**
	50%	63.56	21.31	64.67	67.67	64.83	75.78	**82.83**
C	10%	80.06	92.13	93.06	98.83	98.92	99.17	**99.78**
	20%	76.94	75.56	87.56	96.11	96.94	97.78	**98.56**
	30%	77.69	70.00	70.78	92.33	92.04	96.00	**96.33**
	40%	65.94	47.56	65.07	85.78	81.02	88.28	**90.00**
	50%	58.85	27.94	67.11	68.89	67.23	80.72	**87.67**
D	10%	92.00	82.38	96.33	99.78	99.82	99.61	**99.94**
	20%	86.50	83.50	94.44	99.11	99.23	99.28	**99.78**
	30%	73.38	73.69	91.83	96.61	96.37	97.89	**98.89**
	40%	75.00	56.44	84.00	93.22	90.23	93.94	**95.83**
	50%	71.38	43.69	80.22	82.22	81.00	86.67	**92.67**
E	10%	75.11	85.84	81.97	94.82	94.68	96.23	**97.73**
	20%	65.91	86.80	82.81	92.05	91.22	93.82	**95.78**
	30%	64.92	65.17	69.69	85.32	82.72	90.67	**92.83**
	40%	65.11	56.73	63.28	76.88	74.01	83.14	**86.54**
	50%	55.63	34.94	61.60	65.13	61.61	70.09	**78.27**

Table 3: Accuracy(%) of Better-performed Methods with Different Missing Rates on CWRU Datasets

average training times of these methods across different datasets. GRU-Δt, GRU-Int, and Warpformer show longer training times, mainly due to the high computing complexity during iterative training, especially when handling long sequences. Conversely, the proposed methods, especially Proposed(Decay), achieve significantly shorter times. This stems from the efficient fitting process with Ct-Echo, accomplished only through ridge regression instead of iterative training (like gradient descent), as well as the readily available distance metric between models. The efficiency, combined with the superior classification accuracy given in Table 1, underscores the effectiveness and practicality of the proposed methods for real-world applications.

Discussion with Different Missing Rates Given different missing rates, we evaluated the above five better-performed methods and the proposed ones. The experiments are carried out on five sub-datasets of CWRU, where we systematically introduced missing values at rates ranging from 10% to 50%, with increments of 10%.

Table 3 and Figure 6 reveal the strengths of the proposed methods, particularly Proposed(Leaky), maintaining higher accuracy across increasing missing rates, even under challenging conditions with missing rates $\geq$ 30%. The primary reason is the capture of continuous-time dynamic characteristics via Ct-Echo as aforementioned, resulting in a category-discriminative Ct-Echo model space (intuitively exampled like Figure 5). In this model space, the fitted models are positioned closer within the same category and more distinct across different categories. Consequently, as Figure 6while the difference from some methods is minimal at lower missing rates, the proposed methods demonstrate a more pronounced advantage as the missing rate increases.

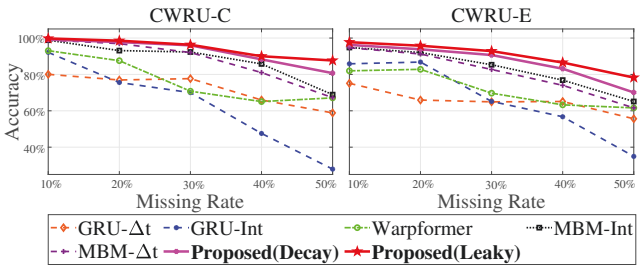


Figure 6: Comparison on better-performed methods on CWRU-C and CWRU-E datasets with different missing rates, from 10% to 50%. The proposed methods (bold font) maintain higher accuracy across increasing missing rates.

	SVM		KNN	
	Proposed (Decay)	Proposed (Leaky)	Proposed (Decay)	Proposed (Leaky)
G20	94.31 ± 0.96	91.25 ± 0.58	87.23 ± 1.81	86.17 ± 2.22
G30	98.15 ± 0.55	97.10 ± 0.58	96.92 ± 0.64	96.54 ± 1.03
B20	99.21 ± 0.28	99.50 ± 0.11	98.47 ± 0.42	98.90 ± 0.44
B30	99.69 ± 0.13	99.69 ± 0.09	99.08 ± 0.26	99.54 ± 0.16

Table 4: Accuracy(%) (mean$\pm$std) of Classification in Ct-Echo Model Space on Su Datasets with 50% Missing Rate

Classification in Ct-Echo Model Space The comparison results between classifiers SVM and K-Nearest Neighbors (KNN) (Altman 1992) in the Ct-Echo model space are given in Table 4. From the results, distance-based classifiers like SVM and KNN are both applicable in the Ct-Echo model space. SVM exhibits higher and more stable accuracy, primarily due to its ability to find an optimal hyperplane that maximizes the margin between categories. Thus SVM is chosen as the classification method in this paper.

Conclusion

This paper proposes learning in the "Ct-Echo Model Space" for anomaly detection of irregular sequences. Ct-Echo fits sequences through continuous-time integration in the hidden layer, capturing their dynamic characteristics. Irregular sequences are represented by the fitted Ct-Echo readout models. Anomaly detection is then performed on these fitted models rather than the original data. Experimental results on several datasets highlight the improved efficiency and effectiveness of our method, especially with diverse missing rates and limited training data. In future work, we plan to extend its applicability to broader domains, focusing on refining and configuring Ct-Echo for more complex data analysis.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China (No. 62206261, 62176245, 62137002).

References

Altman, N. S. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3): 175–185.

Che, Z.; Purushotham, S.; Cho, K.; Sontag, D.; and Liu, Y. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1): 6085.

Chen, A.; Zhou, X.; Fan, Y.; and Chen, H. 2023. Underground Diagnosis Based on GPR and Learning in the Model Space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–14.

Chen, H.; Tang, F.; Tino, P.; Cohn, A. G.; and Yao, X. 2015. Model metric co-learning for time series classification. In *Twenty-fourth international joint conference on artificial intelligence*.

Chen, H.; Tiňo, P.; Rodan, A.; and Yao, X. 2013. Learning in the model space for cognitive fault diagnosis. *IEEE transactions on neural networks and learning systems*, 25(1): 124–136.

Chen, L.; Zhou, X.; and Chen, H. 2024. Audio Scanning Network: Bridging Time and Frequency Domains for Audio Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10): 11355–11363.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Chowdhury, R. R.; Li, J.; Zhang, X.; Hong, D.; Gupta, R. K.; and Shang, J. 2023. PrimeNet: Pre-Training for Irregular Multivariate Time Series. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Cortes, C.; and Vapnik, V. 1995. Support-vector networks. *Machine learning*, 20(3): 273–297.

De Boor, C. 1978. *A practical guide to splines*, volume 27. Springer-Verlag New York.

Dormand, J. R.; and Prince, P. J. 1980. A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1): 19–26.

Gong, Z.; Chen, H.; Yuan, B.; and Yao, X. 2018. Multiobjective learning in the model space for time series classification. *IEEE Transactions on Cybernetics*, 49(3): 918–932.

Jaeger, H. 2001. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34): 13.

Jaeger, H.; Lukoševičius, M.; Popovici, D.; and Siewert, U. 2007. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural networks*, 20(3): 335–352.

Jhin, S. Y.; Lee, J.; and Park, N. 2023. Precursor-of-Anomaly Detection for Irregular Time Series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 917–929.

Keiner, J.; Kunis, S.; and Potts, D. 2009. Using NFFT 3—A Software Library for Various Nonequispaced Fast Fourier Transforms. *ACM Trans. Math. Softw.*, 36(4).

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. 2020. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33: 6696–6707.

Liu, D.; Xiao, Z.; Hu, X.; Zhang, C.; and Malik, O. 2019. Feature extraction of rotor fault based on EEMD and curve code. *Measurement*, 135: 712–724.

Liu, S.; Zhou, X.; and Chen, H. 2024. From Data to D3 Model: Adaptive Subsurface Anomaly Detection in GPR Data. *IEEE Transactions on Geoscience and Remote Sensing*, 62: 1–12.

Loparo, K. 2012. Case western reserve university bearing data center. *Bearings Vibration Data Sets, Case Western Reserve University*, 22–28.

Ma, Q.; Li, S.; Zhuang, W.; Wang, J.; and Zeng, D. 2020. Self-supervised time series clustering with model-based dynamics. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9): 3942–3955.

Quevedo, J.; Chen, H.; Cugueró, M. À.; Tino, P.; Puig, V.; Garciá, D.; Sarrate, R.; and Yao, X. 2014. Combining learning in model space fault diagnosis with data validation/reconstruction: Application to the Barcelona water network. *Engineering Applications of Artificial Intelligence*, 30: 18–29.

Rodan, A.; and Tiňo, P. 2012. Simple deterministically constructed cycle reservoirs with regular jumps. *Neural computation*, 24(7): 1822–1852.

Rubanova, Y.; Chen, R. T.; and Duvenaud, D. K. 2019. Latent ordinary differential equations for irregularly-sampled time series. *Advances in neural information processing systems*, 32.

Shao, S.; McAleer, S.; Yan, R.; and Baldi, P. 2019. Highly Accurate Machine Fault Diagnosis Using Deep Transfer Learning. *IEEE Transactions on Industrial Informatics*, 15(4): 2446–2455.

Shukla, S. N.; and Marlin, B. 2021. Multi-Time Attention Networks for Irregularly Sampled Time Series. In *International Conference on Learning Representations*.

Sterne, J. A.; White, I. R.; Carlin, J. B.; Spratt, M.; Royston, P.; Kenward, M. G.; Wood, A. M.; and Carpenter, J. R. 2009. Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. *Bmj*, 338.

Yuan, Z.; Ban, X.; Zhang, Z.; Li, X.; and Dai, H.-N. 2023. ODE-RSSM: Learning Stochastic Recurrent State Space Model from Irregularly Sampled Data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9): 11060–11068.

Zhang, J.; Zheng, S.; Cao, W.; Bian, J.; and Li, J. 2023. Warpformer: A Multi-Scale Modeling Approach for Irregular Clinical Time Series. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3273–3285. Association for Computing Machinery.