Applied Intelligence (2023) 53:18821–18839
https://doi.org/10.1007/s10489-022-04433-z

Adaptive model training strategy for continuous classification of time series

Chenxi Sun[1,2] · Hongyan Li[1,2] · Moxian Song[1,2] · Derun Cai[1,2] · Baofeng Zhang[1,2] · Shenda Hong[3,4]

Accepted: 26 December 2022 / Published online: 11 February 2023

Abstract
The classification of time series is essential in many real-world applications like healthcare. The class of a time series is usually labeled at the final time, but more and more time-sensitive applications require classifying time series continuously. For example, the outcome of a critical patient is only determined at the end, but he should be diagnosed at all times for timely treatment. For this demand, we propose a new concept, Continuous Classification of Time Series (CCTS). Different from the existing single-shot classification, the key of CCTS is to model multiple distributions simultaneously due to the dynamic evolution of time series. But the deep learning model will encounter intertwined problems of catastrophic forgetting and over-fitting when learning multi-distribution. In this work, we found that the well-designed distribution division and replay strategies in the model training process can help to solve the problems. We propose a novel Adaptive model training strategy for CCTS (ACCTS). Its adaptability represents two aspects: (1) Adaptive multi-distribution extraction policy. Instead of the fixed rules and the prior knowledge, ACCTS extracts data distributions adaptive to the time series evolution and the model change; (2) Adaptive importance-based replay policy. Instead of reviewing all old distributions, ACCTS only replays important samples adaptive to their contribution to the model. Experiments on four real-world datasets show that our method outperforms all baselines.

Keywords Continuous classification of time series · Model training strategy · Medical applications

1 Introduction

The classification of time series has attracted increasing attention in many practical fields [1]. The class of a time series is usually labeled at the final time. For example, patients' outcomes will come at the end. Most deep learning (DL) models are good at single-shot classification, classifying data at a fixed time after learning time series within a fixed period [2, 3]. Because DL methods assume that the observed data is independent and identically distributed (i.i.d) and subsequences in the same period maintain one distribution [4].

However, in the real world, more and more time-sensitive applications need to classify time series continuously before the final labeled time [3]. For example, in the intensive care unit (ICU), diagnosis and prognosis are needed at any time to provide more opportunities for doctors to rescue lives [5]. Each hour of delay has been associated with roughly a 4-8% increase in sepsis mortality [6]. But patient labels, e.g. mortality or morbidity, are only available at the onset time but unknown in the early stages. In response to the current demand, we propose a new concept – Continuous Classification of Time Series (CCTS), to classify time series at every time point before the labeled time. For example, using vital signs like blood pressure to diagnose patients continuously as shown in Fig. 1.

The main requirement of CCTS is to model multi-distributed data. Most real-world time series develop dynamically, leading to the evolved data distribution, and finally producing the multi-distribution form. For example, in Fig. 2, the data distribution of blood pressure of 2,000 sepsis patients varies among early, middle, and late time stages during hospitalization, bringing a triple-distribution. Because these three distributions have the same sepsis label, the model needs to learn them simultaneously to achieve continuous classification: When

✉ Hongyan Li
leehy@pku.edu.cn

✉ Shenda Hong
hongshenda@pku.edu.cn

Extended author information available on the last page of the article.

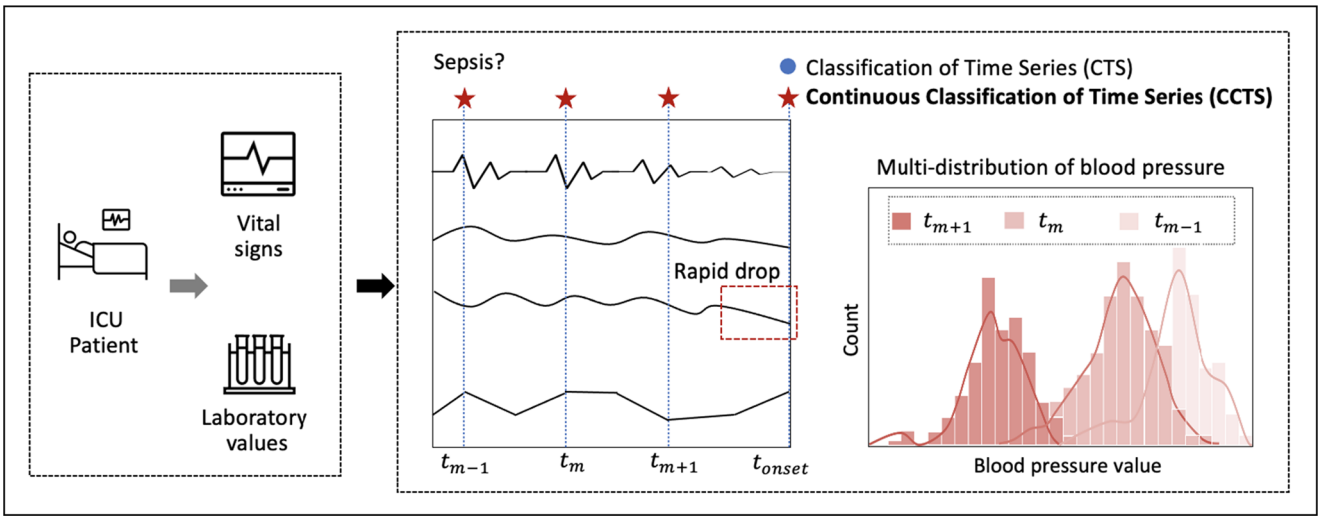


Fig. 1 A Medical Case of Continuous Classification of Time Series (CCTS): Continuous diagnosis and prognosis, where the vital signs is modeled to classify patients' health status continuously. For sepsis, the rapid drop of blood pressure (a major symptom of sepsis shock, the red dashed box) always occurs just before the shock, but its too late. The continuous mode (red stars) can achieve earlier and more accurate results than the single-shot mode (blue dot). If the model simply learns the full-length time series, it can only give the single-shot result at the onset time. If it is expected to diagnose continuously, it needs to learn data from different advanced stages, where the blood pressure has a triple-distribution at t_{m-1}, t_m, t_{m+1}

the data distribution changes, the model performance cannot decrease. However, limited by the premise of i.i.d data, if a model learns a new distribution, it will negatively affect its performance on old ones. That is the catastrophic forgetting problem [7].

Some studies, including our previous work, have proposed some solutions to this problem [8–10]. However, they are based on the known multiple data distributions, yet the distribution division in CCTS is not clear. In the context of CCTS, a time series is not one sample but can be divided into multiple samples. Different division rules will produce different distributions and also affect the final model performance. Less distributions may worsen the catastrophic forgetting problem and omit important features. More distributions may cause the over-fitting problem and have low training efficiency. For example, if the model learns distributions in each time point, it will encounter intertwined problems of catastrophic forgetting and over-fitting: A time series usually has a large number of time points. The blood pressure of a critical patient could be sampled hundreds of times. If the model frequently learns hundreds of new distributions, it will inevitably forget old ones. Meanwhile, as the development of time series needs a process, the data distributions in adjacent time are always similar. Over-learning of similar distributions will cause strict function and poor generalization [11].

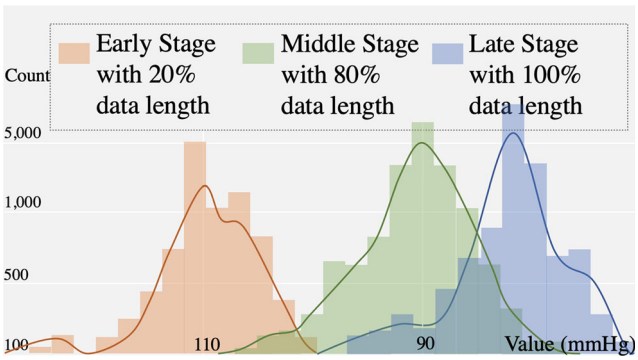


Fig. 2 Multi-distribution in Time Series dataset. The statistics of blood pressure of 2,000 sepsis represent three distributions [14]

The optimal multi-distribution is hard to obtain. Unlike images, the time series is more abstract and its characteristics are not explicit [12]. Although some methods can describe time series like Shaplets [13], they still need prior knowledge. Most importantly, the artificial rule needs to be determined before training the model and remains the same over time. But because the time series has been evolving dynamically, a fixed rule is likely to be outdated.

In this work, instead of the static division rule, we design an Adaptive model training strategy for CCTS (ACCTS). It has two adaptive policies:

- Adaptive multi-distribution extraction policy. It explores the policy space according to the reward based on distribution difference and classification accuracy, and finally extracts data distributions adaptive to the time series evaluation and the model change;
- Adaptive importance-based replay policy. It leans the impact of each sample on the model, applying partial replay to balance the problems of catastrophic forgetting and over-fitting. The important samples in each distribution are determined adaptive to their dynamic importance parameters.

Experimental results on real-world datasets show that ACCTS is more accurate than all baselines in CCTS task.

2 Related work

We summarize the classification tasks for time series data into two categories: single-shot classification and continuous classification. (See Appendix A for more related work and concepts.)

2.1 Single-shot classification

Definition 1 (Single-shot Classification of Time Series, SCTS) A time series $X = \{x_1, ...x_T\}$ is labeled with a class $C \in \mathcal{C}$ at the final time T. SCTS classifies X at a fixed time t with a single minimum loss $\mathcal{L}(f(X_{1:t}), C)$. If $t = T$, the task is CTS; If $t < T$, the task is ECTS.

Single-shot classification methods classify at a fixed time. The classical Classification of Time Series (CTS) gives results based on the full-length data [2]. But in time-sensitive applications, Early Classification of Time Series (ECTS) is more critical, making classification at an early time [3]. For example, early diagnosis helps for sepsis outcomes [15].

Many methods have been proposed and have good results in CTS and ECTS [16–22]. Because DL methods assume that the observed data is i.i.d and subsequences in the same period maintain one distribution. However, in the real world, more and more time-sensitive applications need to classify time series continuously before the final labeled time. As shown in Fig. 3, SCTS can only give the single-shot result: once the classification is complete, the action will not continue (Table 1).

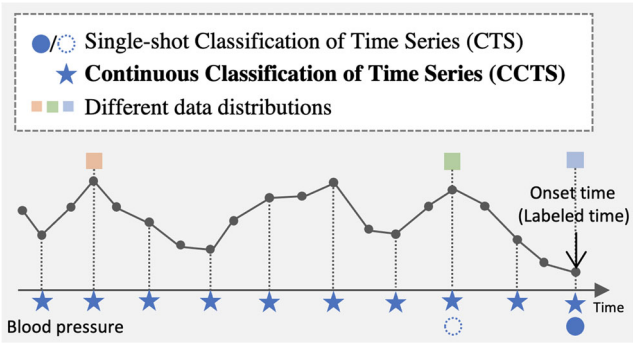


Fig. 3 Continuous Classification. CCTS is continuous mode (star) with multi-distribution (square) rater than single-shot mode (circle)

Table 1 Notations and the corresponding definitions

Notation	Definition	Notations	Definition
$\mathcal{X}$	time series dataset	f	model
$\mathcal{D}$	distribution set	μ, Q	actor-critic nets
$\mathcal{M}$	task set	$\mathcal{S}, s$	state
X, x	time series sample	$\mathcal{A}, a$	action
C, c	class label	$\mathcal{R}, r$	reward
T, t	time stamp	θ, W, b	model parameters
$\mathcal{B}$	data buffer	$\alpha, \epsilon, \lambda$	hyper-parameter
$\mathcal{L}$	loss function	g	gradient

2.2 Continuous classification

Definition 2 (Continuous Classification of Time Series, CCTS) A time series $X = \{x_1, ...x_T\}$ is labeled with a class $C \in \mathcal{C}$ at the final time T. CCTS classifies X at every t with the additive loss $\sum_{t=1}^{T} \mathcal{L}(f(X_{1:t}), C)$.

Continuous classification methods classify at every time point before the labeled time. In fact, CCTS is a combination of multiple SCTS tasks. As we analyzed in Section 1, the premise of realizing continuous classification is to model multi-distribution. We summarize two strategy categories.

2.2.1 Multi-model for multi-distribution

The first strategy applies multiple models to model multiple distributions, like SR [23] and ECEC [24]. They divide data distribution according to time stages and design a classifier for each distribution. But they only consider the data division, ignoring the strategic training method. Besides, the operation of classifier selection in a multi-model framework will result in additional losses.

2.2.2 Single-model for multi-distribution

The second strategy uses a single model to learn multiple distributions and solves the problem of catastrophic forgetting in this process. They are usually based on a Continual Learning (CL) framework, which enables the model to learn new tasks over time without forgetting the old tasks. For example, replay-based methods re-train the model by old data to consolidate memory [14, 25–27]; Regularization-based methods restrain parameter update of neural networks to limit forgetting [28–31]; Model-based methods change network structure or apply multiple models to response to different tasks [32, 33]. But most methods have the problems of storage limitation, distribution drifts, and model overfitting. In CL, the definition of old and new

tasks is clear and the division of distribution is fixed. But in CCTS, the distributions are not determined and need to be defined. Besides, two sub-disciplines, Online Learning (OL) [34] and Anomaly Detection (AD) [35], also study the mode of continuous learning or continuous classification. But they mainly maintain one data distribution. When they are directly applied in CCTS, they perform poorly at early time points.

In most methods, either all samples are assumed to be in the same distribution, or the multi-distribution is defined in advance, or the distribution division is based on the full-length time series data [36, 37]. But CCTS task need to divide time series dynamically during their evolution as a fixed rule is likely to be outdated.

3 Continuous classification of time series

As shown in Fig. 3, CCTS aims to give classification results at each time point of the time series. Based on Definition 2, in CCTS, the model need to learn multiple distributions. Without the loss of generality, we use the univariate time series to present this task. Multivariate time series can be described by changing x_t to x_t^i. i is the i-th dimension.

Definition 3 (CCTS with Multi-distribution) A dataset $\mathcal{X}$ contains many time series data. Each time series $X = \{x_1, ...x_T\}$ is labeled with a class $C \in \mathcal{C}$ at the final time T. As time series varies among time, it has a subsequence series with N different distributions $\mathcal{D} = \{\mathcal{D}^1, ..., \mathcal{D}^N\}$, each $\mathcal{D}^n$ has subsequence $X_{1:t^n}$. CCTS learns every $\mathcal{D}^n$ and introduces a task sequence $\mathcal{M} = \{\mathcal{M}^1, ..., \mathcal{M}^N\}$ to minimize the additive risk $\sum_{n=1}^{N} \mathbb{E}_{\mathcal{M}^n}[\mathcal{L}(f^n(\mathcal{D}^n; \theta), C)]$ with model f and parameter θ. f^n is the model f after being trained for $\mathcal{M}^n$. When the model is trained for $\mathcal{M}^n$, its performance on all observed data cannot degrade:

$$\begin{aligned} &\min \mathcal{L}(f^n, \mathcal{M}^n) \\ &\text{subject to } \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f^n(X_{1:t^i}; \theta^n), C) \\ &\quad \le \frac{1}{n-1}\sum_{i=1}^{n-1} \mathcal{L}(f^{n-1}(X_{1:t^i}; \theta^{n-1}), C) \end{aligned} \tag{1}$$

4 Adaptive model training strategy

To achieve the CCTS task defined in Definition 3, we first divide the time series dataset $\mathcal{X}$ based on the distribution set $\mathcal{D}$ and create the task set $\mathcal{M}$, then learn new and old tasks, avoiding catastrophic forgetting and over-fitting.

In this work, we propose an adaptive model training strategy ACCTS as shown in Fig. 4. When a model is trained by time series from the initial to the final time, ACCTS gives two decisions:

- Whether the current time series segment forms a new distribution? If yes, train the model by the current time series; Otherwise, do not train and continue to get new data points;
- Which old samples need to be replayed and learned again? If the previous decision is yes, train the model with the obtained old samples again after training it by the current time series.

4.1 Adaptive multi-distribution extraction

The first decision is got by the adaptive multi-distribution extraction polity. It is an agent that decides whether to

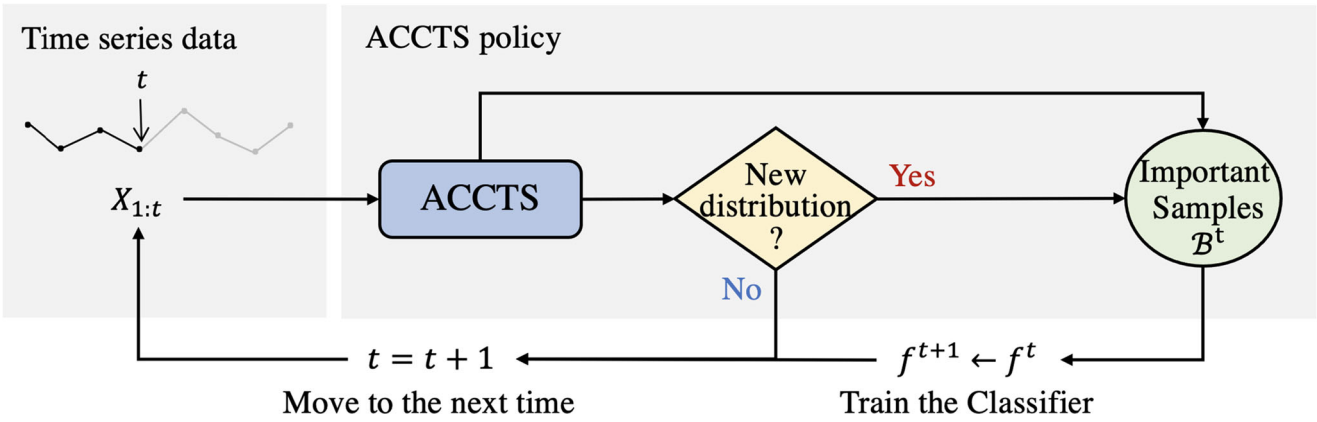


Fig. 4 Adaptive model training process for continuous classification of time series

extract the current time series sequence to train the model. It solves a 3-triple partially-observable Markov decision process $\{\mathcal{S}, \mathcal{A}, \mathcal{R}\}$ [38], where the observation arrive from a state s at each time, an action a is sampled using a learned policy, and a reward r is observed according to the selected action's quality. The objective is to optimize long-term rewards.

State $\mathcal{S}$. It is represented by the characteristics of the current data and the adaptability of the old model to the current data. It is intuitive: First, the model needs to be trained by the dataset with different features from the previous data for the comprehensive modeling; Second, the model must be trained again when it performs poorly on the current data for overall accuracy. At the current time t, we use the Long Short-Term Memory (LSTM) network as the base model to learn the hidden characteristics of a time series $X_{1:t}$, generating low-dimensional vector representation h_t. We also propose the Model Gradient (MG) g_t to evaluate the adaptability of the model to the current time series. The model gradient can help for the interpretation of the DL model by explaining the response of the neural network to input data [39]. Large gradient fluctuation reflects the low adaptability of the model to the input data. Thus, the state s_t of the current time series is:

$$s_t = \text{concatenate}(\text{LSTM}(x_t), \text{MG}(X_{1:t})) \tag{2}$$

$$\begin{aligned} \text{LSTM}(x_t) &= h_t = o_t \cdot \eta(c_t) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot \eta(W_c[h_{t-1}, x_t] + b_c) \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \end{aligned} \tag{3}$$

$$\text{MG}(X_{1:t}) = g_t = \frac{\partial}{\partial \theta_{f^n}} \mathcal{L}(f^n(X_{1:t}, \theta_f), C) \tag{4}$$

Action $\mathcal{A}$ At the current time t, the action a_t dictates the decisions of ACCTS agent: If $a_t = 0$, continue to accept the value point of time series and let LSTM move forward one time step; If $a_t = 1$, extract the current time series $X_{1:t}$ as a new distribution to be learned. For the action selection, we use ε-greedy selection to avoid abundant exploitation. a_t is replaced with a random action with the probability ε of exponentially decreasing from 1 to 0 during the training process.

$$a_t = \begin{cases} a_t, & \text{with probability } 1-\varepsilon \\ random, & \text{with probability } \varepsilon \end{cases}, \quad a_t \in \{0, 1\} \tag{5}$$

Reward $\mathcal{R}$ The agent observes the return which can qualify the parameters of the current policy. The goal of CCTS is the high accurate classification by solving the problems

Input: Data $\mathcal{X} = \{(X, C)\}, X = \{x_1, ..., x_T\}$; Classifier net f with parameter θ_f; Actor-Critic net μ, Q of ACCTS with parameter θ.
Output: Extraction policy (μ, Q).

```
for t = 1 to T do
    s_t ← (2)
    a_t ← μ(s_t, θ_Actor)
    if a_t = 1 then
        Train f by (X_{1:t}, C)
        r_t ← (6)
        Update Q, μ by:
        ∇_{θ_Q} O_critic Σ_t (y − ∇_{θ_Q} Q(s_t, a_t|θ_Q))
        ∇_{θ_μ} O_actor = Σ_t ∇_{μ(s_t)} Q(s_t, μ(s_t)) ∇_{θ^μ} μ(s_t|θ_μ)
        Soft update Q', μ' by:
        θ_{Q'} ← τθ_Q + (1 − τ)θ_{Q'}
        θ_{μ'} ← τθ^μ + (1 − τ)θ_{μ'}
    else
        Continue
    end if
end for
```

Algorithm 1 Adaptive multi-distribution extraction policy.

of catastrophic forgetting and over-fitting as we analysed in Section 1. Thus, we pursue the higher accuracy of the current classifier on all potential data distributions to control the catastrophic forgetting, and we limit the number of extracted distributions by the time span between distributions to control the over-fitting. Thus, at the current time t, after applying the action a_t, the reward r_t is consisted of two components. The first term is for the high accuracy of the current model f^n on all data, the second term is for less divisions by using the time length between the current time t^n and the last data extraction time t^{n-1}.

$$r_t = -\frac{\alpha}{T} \sum_{t=1}^{T} \mathcal{L}(f^n, X^{1:t}) + \frac{(1-\alpha)}{T}(|t^n - t^{n-1}|) \tag{6}$$

When using the transition probability $P(s_{t+1}|s_t, a_t)$, the total reward of the trajectory is the is sum of the reward in each time. Thus, the objective is to maximize the total reward $R = \sum_{t=1}^{T} r_t$. The policy gradient method [40] learns the policy $\pi_\theta(s_t, a_t) = P(a_t|s_t)$ for the larger return. The objective is $J(\theta) = \mathbb{E}[r(s, a)\pi_\theta(s, a)]$. For ACCTS, we apply Actor-Critic [41] structure with two components of the main net and the target net. The main net of Actor μ use the state s to generate the action a; The main net of Critic $\mathcal{Q}$ judges the action a through reward r by Q-function [42]. The target nets of Actor and Critic μ', $\mathcal{Q}'$ put the target Q value stable for a period of time, making the algorithm performance more stable.

$$\begin{aligned} O_{Actor}(\theta_\mu) &= \mathbb{E}_{s_t \in S}\left[\mathcal{Q}\left(s_t, \mu\left(s_t|\theta_\mu\right)|\theta_\mathcal{Q}\right)\right] \\ O_{Critic}(\theta_\mathcal{Q}) &= \mathbb{E}_{s_t \in S}[(r_t + \gamma \mathcal{Q}'(s_{t+1}, \mu'(s_{t+1}|\theta_{\mu'})|\theta_{Q'}) \\ &\quad - \mathcal{Q}(s_t, a|\theta_\mathcal{Q}))^2] \end{aligned} \tag{7}$$

4.2 Adaptive importance-based replay

The replay mechanism can help to alleviate the catastrophic forgetting [43]. However, the operation of repeated replay easily causes the over-fitting problem, especially for time series with small differences between two adjacent times. In CL, many methods only replay the representative data, such as the class means [27] and the class prototype [44], each representative is fixed to its distribution. But in CCTS, we still need to consider whether all the representatives need to be learned again and whether the representative will change over time.

Thus, we focus on the adaptive method to explore a wider space, where the replayed data is dynamic and determined according to the current state. We introduce an importance-based replay method. In each round, it only re-trained the model with some important samples to the model. The importance of each sample is learned from the objective of an additive loss function.

We incorporate the importance parameter β_i of a time series X_i in the replay buffer $\mathcal{B}^n$ as a coefficient of its loss $\mathcal{L}_{n,i}$. The overall loss at the current time t^n is the sum of each sample's loss:

$$\begin{aligned} \mathcal{L}_n &= \frac{1}{|\mathcal{B}^n|}\sum_{i=1}^{|\mathcal{B}^n|}(\beta_{n,i}^2\mathcal{L}_{n,i} + \lambda(\beta_{n,i}-1)^2) \\ \mathcal{B}^n &= \{X_{1:t^n}, \tilde{X}_i | \beta_{n-1,i} < \epsilon\} \end{aligned} \tag{8}$$

β is learned by the gradient descent $\beta_{n,i} \leftarrow \beta_{n,i} - \frac{\partial \mathcal{L}_n}{\partial \beta_{n,i}}$. Thus, if a sample X_i is hard to classify, its loss $\mathcal{L}_{*,i}$ will be larger. In order to minimize the loss, its $\beta_{*,i}$ will be smaller. Based on this, in each learning phrase, the buffer $\mathcal{B}^n$ contains the current time series $X_{1:t^n}$ and the important old time series $\tilde{X}$, who are the first few difficult learning samples ($\beta_{n-1,i} < \epsilon$) in the last buffer $\mathcal{B}^{n-1}$. Meanwhile, as β is the confidence of loss, if $\beta = 0$, the loss is hard to optimize. Thus, inspired by [14], we introduce a regularization term $(\beta-1)^2$ and initialize $\beta = 1$ to penalize it when rapidly decaying toward 0. As β is re-obtained after each model training process, the important samples $\tilde{X}$ are changed adaptively and the buffer $\mathcal{B}$ is updated iteratively.

4.3 Overall model training process

The adaptive multi-distribution extraction policy, which is achieved by the Actor net μ, is trained before the classifier training process, as shown in Algorithm 1. First, LSTM calculates the current sate s_t (Line 2) and gives the action a_t (Line 3). Then, the reward r_t is obtained by the long-term accuracy to update the net (Line 6), where Actor and Critic are updated alternately. The main Critic net is updated by Q value, calculated from both two Critic. Main Actor is updated by the back-propagation gradient of the main Critic.

Input: Data $\mathcal{D} = \{(X, C)\}$, $X = \{x_1, ..., x_T\}$; Actor net μ of ACCTS.
Output: Final Classifier net f^T.
1: Initialize importance buffer $\mathcal{B}_1 \leftarrow \{X_{1:1}\}$
2: Initialize DL Classifier net f^1.
3: **for** $t = 2$ to T **do**
4: $\quad \mathcal{B}_t \leftarrow \mathcal{B}_t + \{X_{1:t}\}$
5: $\quad s_t \leftarrow \{h_t, g_t\}$ from (2)
6: $\quad a_t \leftarrow \mu(s_t)$
7: //ADAPTIVE EXTRACTION
8: $\quad$ **if** $a_t = 1$ **then**
9: $\quad\quad f^t \leftarrow$ Train f^{t-1} by $\mathcal{B}^t$ with (8)
10: //IMPORTANCE STORAGE
11: $\quad\quad \mathcal{B}^{t+1} \leftarrow \{X_{1:n,k} | \beta_k < \epsilon\}$
12: $\quad$ **else**
13: $\quad\quad \mathcal{B}^{t+1} \leftarrow \mathcal{B}^t$
14: $\quad$ **end if**
15: **end for**

Algorithm 2 The model training process under the strategy of ACCTS.

Target Actor and Critic are learned by the soft update (Line 7).

The adaptive importance-based replay policy is trained along with the classifier training process, as shown in Algorithm 2. First, in each time step, the Actor of ACCTS determines if a new distribution appears (Line 4,5). If yes, train the classifier from f^n to f^{n-1} by datasets in the buffer $\mathcal{B}^n$ (Line 7,8), and get the important samples according to β to form a new buffer $\mathcal{B}^{n+1}$ (Line 9); Else, continue to get new values in next time point $t+1$. At the final time, we can get the well-trained classifier f^N.

Note that the two processes of the adaptive multi-distribution extraction and the adaptive importance-based replay are relevant rather than independent. The extraction policy is based on the feature of the buffer data, and the replay policy selects the important samples based on the extracted data. Both of them are data-based, which helps to adaptive combination. That's why we design the replay-based policy rather than the regularization-based policy after the distribution extraction.

5 Experiments

5.1 Experimental setup

Datasets For each time series in the four datasets, every time point is tagged with a class label, which is the same as its outcome label, such as 'mortality', 'sepsis', 'earthquake' and 'rain'.

- COVID-19 dataset [45] has 6,877 blood samples of 485 COVID-19 patients from Tongji Hospital, Wuhan, China. It is the multivariate time series of 74

Table 2 Classification accuracy (AUC-ROC↑) of baselines at 10 time points for 4 real-world datasets

	Method	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
COVID-19	LSTM	.605±.04	.701±.03	.793±.02	.833±.01	.844±.01	.888±.01	.918±.03	.925±.01	.939±.00	.944±.01
	SR	.636±.01	.730±.02	.810±.01	.867±.01	.901±.01	.900±.01	.935±.01	.946±.00	.952±.01	.962±.00
	ECEC	.639±.01	.732±.02	.829±.01	.870±.01	.901±.02	.904±.01	.937±.00	.948±.01	.952±.00	.963±.01
	EWC	.703±.02	.769±.01	.870±.01	.888±.02	.915±.01	.923±.01	.935±.00	.940±.01	.950±.01	.954±.00
	GEM	.699±.02	.779±.01	.871±.01	.885±.02	.914±.01	.924±.01	.936±.00	.939±.01	.949±.01	.953±.00
	CLEAR	.710±.01	.785±.01	.870±.01	.879±.01	.916±.02	.926±.01	.933±.01	.941±.00	.948±.00	.952±.00
	CLOPS	.709±.01	.775±.01	.869±.01	.900±.01	.918±.02	.925±.01	.935±.01	.940±.00	.947±.00	.954±.00
	ACCTS	**.712±.02**	**.790±.02**	**.872±.01**	**.901±.02**	**.919±.01**	**.927±.00**	**.955±.00**	**.960±.01**	**.963±.00**	**.967±.00**
SEPSIS	LSTM	.576±.06	.629±.03	.735±.06	.736±.06	.745±.05	.748±.04	.773±.03	.795±.02	.813±.02	.827±.03
	SR	.626±.03	.659±.01	.768±.01	.791±.02	.803±.01	.827±.03	.835±.01	.845±.01	.859±.02	.866±.02
	ECEC	.623±.02	.669±.01	.761±.01	.793±.01	.811±.01	.815±.01	.827±.01	.849±.01	.859±.01	.863±.01
	EWC	.671±.02	.733±.02	.799±.01	.827±.03	.832±.02	.838±.02	.842±.03	.848±.01	.850±.01	.854±.01
	GEM	.670±.02	.730±.02	.802±.01	.826±.03	.834±.02	.836±.02	.841±.03	.849±.01	.851±.01	.853±.01
	CLEAR	.680±.02	.732±.02	.801±.01	.825±.03	.833±.02	.839±.02	.842±.03	.847±.01	.850±.01	.848±.01
	CLOPS	.684±.02	.733±.02	.802±.01	.824±.03	.830±.02	.838±.02	.842±.03	.850±.01	.853±.01	.857±.01
	ACCTS	**.690±.03**	**.734±.03**	**.812±.02**	**.828±.03**	**.835±.02**	**.842±.03**	**.852±.02**	**.857±.01**	**.866±.01**	**.872±.01**
UCR-EQ	LSTM	.695±.04	.711±.03	.803±.02	.843±.01	.854±.01	.874±.01	.913±.03	.909±.01	.919±.00	.924±.01
	SR	.700±.01	.736±.01	.830±.01	.863±.01	.871±.02	.888±.01	.924±.01	.928±.10	.936±.10	.941±.10
	ECEC	.703±.01	.738±.01	.828±.01	.865±.01	.873±.02	.890±.01	.923±.01	.929±.10	.936±.00	.940±.00
	EWC	.724±.01	.768±.01	.848±.01	.874±.01	.883±.02	.895±.01	.910±.01	.923±.10	.930±.00	.933±.00
	GEM	.723±.01	.767±.01	.850±.01	.876±.01	.890±.02	.900±.01	.920±.01	.929±.00	.935±.00	.934±.00
	CLEAR	.729±.01	.770±.01	.852±.01	.880±.01	.899±.02	.904±.01	.918±.01	.923±.00	.928±.00	.932±.00
	CLOPS	.728±.01	.773±.01	.855±.01	.878±.01	.896±.02	.902±.01	.915±.01	.917±.00	.921±.00	.925±.00
	ACCTS	**.730±.02**	**.774±.02**	**.856±.01**	**.882±.02**	**.900±.01**	**.906±.00**	**.928±.00**	**.933±.01**	**.940±.00**	**.946±.00**
USHCN	LSTM	.682±.01	.700±.02	.721±.01	.745±.02	.784±.02	.820±.01	.837±.02	.852±.01	.869±.02	.891±.00
	SR	.702±.01	.730±.02	.745±.01	.761±.02	.809±.02	.836±.01	.886±.02	.902±.01	.921±.02	.933±.00
	ECEC	.707±.01	.736±.02	.748±.01	.760±.02	.806±.02	.837±.01	.887±.02	.906±.01	.920±.02	.931±.00
	EWC	.727±.01	.736±.02	.768±.01	.798±.02	.805±.02	.834±.01	.867±.02	.896±.01	.906±.02	.926±.00
	GEM	.720±.01	.728±.02	.772±.01	.781±.02	.801±.02	.838±.01	.868±.02	.899±.01	.910±.02	.928±.00
	CLEAR	.728±.01	.738±.02	.773±.01	.784±.02	.802±.02	.837±.01	.867±.02	.879±.01	.899±.02	.921±.00
	CLOPS	.728±.01	.740±.02	.769±.01	.781±.02	.800±.02	.835±.01	.861±.02	.877±.01	.895±.01	.919±.01
	ACCTS	**.730±.01**	**.742±.01**	**.775±.01**	**.791±.02**	**.810±.01**	**.841±.01**	**.898±.02**	**.910±.01**	**.928±.01**	**.939±.01**

The bold font indicates the most accurate result

laboratory test features. Mortality prediction helps for the personalized treatment and resource allocation [46].

- SEPSIS dataset [47] has 30,336 patients' records, including 2,359 diagnosed sepsis. It is the multivariate

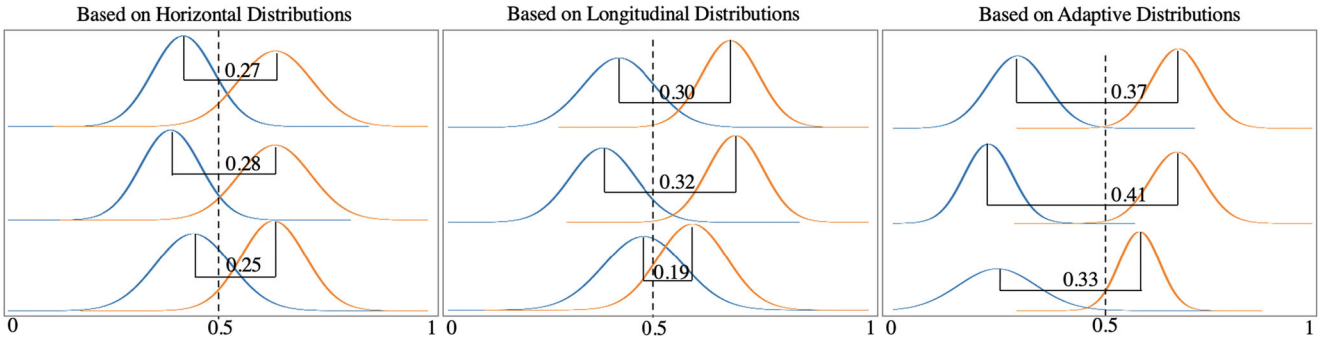


Fig. 5 Sepsis Diagnosis Based on Different Distribution Divisions. The values are the sigmoid values in binary classification task. The greater the difference between the values of the two classes, the more helpful for model classification

Table 3 Continual learning performance (Left: BWT↑, Right: FWT↑) of baselines

	EWC	GEM	CLEAR	CLOPS	ACCTS		EWC	GEM	CLEAR	CLOPS	ACCTS
UCR-EQ	+0.039	+0.041	+0.053	+0.052	**+0.058**	UCR-EQ	+0.321	+0.329	+0.312	+0.301	**+0.345**
USHCN	+0.058	+0.054	+0.063	+0.074	**+0.084**	USHCN	+0.312	+0.328	+0.335	+0.301	**+0.342**
COVID-19	+0.011	+0.012	+0.009	+0.014	**+0.020**	COVID-19	+0.426	+0.421	+0.427	+0.439	**+0.455**
SEPSIS	+0.019	+0.017	+0.030	+0.032	**+0.035**	SEPSIS	+0.295	+0.265	+0.401	+0.397	**+0.410**

The bold font indicates the best continual learning performance

time series of 40 related patient features. Early diagnose of sepsis is critical to improve the outcome of ICU patients [48].

- UCR-EQ dataset [49] has 471 earthquake records from UCR time series database archive. It is the univariate time series of seismic feature value. Natural disaster early warning, like earthquake warning, helps to reduce casualties and property losses [50].
- USHCN dataset [51] has the daily meteorological data of 48 states in U.S. from 1887 to 2014. It is the multivariate time series of 5 weather features. Rainfall warning is not only the demand of daily life, but also can help prevent natural disasters [52].

Baselines LSTM is the base model. The baselines are mainly composed of two categories as we introduced in Section 2. SR and ECEC are multi-model structures; EWC, GEM, CLEAR, CLOP have CL strategies.

- LSTM [17, 53]. It contains a single classification model LSTM. For one time series, the classification model is trained by all subsequences from time 1 to time t, where $t = 2, ..., T$.
- SR [23]. It has multiple basic classification models. All models are trained by the full-length time series. The final classification is the fusion result. It also has a stop rule of classification stop time.
- ECEC [24]. It has a set of basic classification models. Each model is trained by time series in different time stages. When classifying, the data selects the classifier based on its time stages.
- EWC [28]. It is a regularization-based strategy in continual learning field. The strategy trains a model to remember the old tasks by constraining important parameters to stay close to their old values.
- GEM [29]. It is a regularization-based strategy in continual learning field. The strategy trains a model to remember the old tasks by finding the new gradients which are at acute angles to the old gradients.
- CLEAR [25]. It is a replay-based strategy in continual learning field. The strategy uses the reservoir sampling to limit the number of stored samples to a fixed budget assuming an i.i.d. data stream.
- CLOPS [14]. It is a replay-based strategy in continual learning field. The strategy trains a base model by replaying old tasks with importance-guided buffer storage and uncertainty-based buffer acquisition.

Evaluation metrics Results are got by 5-fold cross validation, expressed as the mean and standard deviation mean±std. The accuracy is evaluated by Area Under Curve of Receiver Operating Characteristic (AUC-ROC). The performance of continuous mode is evaluated by Backward Transfer (BWT) and Forward Transfer (FWT), the influence that learning a current has on the old/future. $R \in \mathbb{R}^{|\mathcal{M}|\times|\mathcal{M}|}$ is an accuracy matrix, $R_{i,j}$ is the accuracy on $\mathcal{M}^j$ after learning $\mathcal{M}^i$. $\bar{b}$ is the accuracy with random initialization.

$$\text{BWT} = \frac{1}{|\mathcal{M}|-1}\sum_{i=1}^{|\mathcal{M}|-1} R_{|\mathcal{M}|,i} - R_{i,i} \tag{9}$$

$$\text{FWT} = \frac{1}{|\mathcal{M}|-1}\sum_{i=2}^{|\mathcal{M}|} R_{i-1,i} - \bar{b}_{i,i} \tag{10}$$

Table 4 Classification accuracy of baselines with non-uniform training sets and validation sets of COVID-19 dataset

Subset	LSTM	SR	ECEC	EWC	GEM	CLEAR	CLOPS	ACCTS
Male	.955±.01	.968±.01	.969±.01	.965±.01	.965±.00	.978±.00	.978±.01	.971±.01
Female	.924±.01	.945±.00	.947±.01	.939±.01	.938±.00	.919±.00 ↓	.921±.00 ↓	.947±.00
Age 30-	.954±.01	.965±.01	.967±.01	.967±.01	.964±.00	.977±.00	.979±.01	.972±.01
Age 30+	.923±.01	.941±.00	.943±.01	.931±.00 ↓	.923±.00 ↓	.902±.00 ↓	.914±.00 ↓	.945±.00
Test	.950±.01	.964±.01	.968±.01	.966±.01	.962±.00	.979±.00	.978±.01	.970±.00
Validation	.944±.01	.962±.00	.963±.01	.954±.00	.953±.00	.952±.00 ↓	.954±.00 ↓	.967±.00

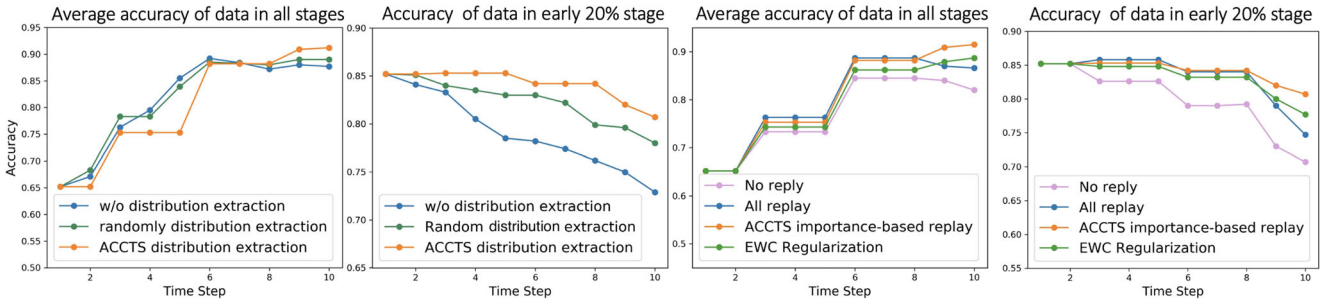


Fig. 6 Ablation study of two policies of ACCTS with the case study of COVID-19

5.2 Results and analysis

We test the baselines from the classification accuracy and performance of solving problems of catastrophic forgetting and over-fitting problems, analyze our ACCTS from the ablation study and coefficient test, and show the representation of time series in continuous classification.

Before discussing the method performance, we show the basic scenario of CCTS – multi-distribution. As shown in Fig. 2, the data in different time stages (20%-, 50%-, 100%-length) have distinct statistical characteristics and finally form multiple distributions. The fundamental goal of the following experiment is to model them.

5.2.1 Continuous classification

ACCTS has the best performance on continuous classification. As shown in Table 2, it can classify time series more accurately than all baselines at every time. The average accuracy is about 2% higher. Specifically, ACCTS is significantly better than baselines in Bonferroni-Dunn tests: Rank(baselines) $= 4.5 > 1.80+1(k = 7, n = 4, m = 5)$. k, n, m are the number of methods, datasets, cross-validation fold, $CD = q_\alpha\sqrt{\frac{k(k+1)}{6(nm)}}$, if the average rank of baselines higher $CD + 1$, the result is significantly improved.

The accurate continuous classification is important for time-sensitive applications. Take continuous sepsis diagnosis and prognosis in ICU as an example, compared with the best baseline, our method improves the accuracy by 1.32% on average, 2.19% in the early 50% time stage when the key features are unobvious. Each hour of delayed treatment increases sepsis mortality by 4-8% [48]. With the same accuracy, we can predict 0.951 hours in advance.

The adaptive division strategy is better than the static division strategy. As shown in Fig. 5, the distance between the sigmoid values of the two prediction classes is relatively large. It demonstrates the necessity of the combination of data division and model generation. Using the horizontal distributions based on clustering, the effect difference among models is relatively small. Using the longitudinal distributions, the model effect becomes better with the development of the time stage.

5.2.2 Catastrophic forgetting and over-fitting

ACCTS is the best when solving these two problems with the highest BWT and FWT as shown in Table 3. Not like the strategies of EWC, GEM, CLEAR, and CLOPS, where they train and review time series at all time points, ACCTS trains and reviews time series at adaptively selected time points. The results in Table 3 show the benefits of the adaptive strategy: It has the lowest negative influence that learning the new tasks has on the old tasks and has the highest positive influence that learning the former data distributions has on the task. Meanwhile, ACCTS can avoid model overfitting and guarantee certain model generalizations. In Table 4, for most baselines, the accuracy on the validation

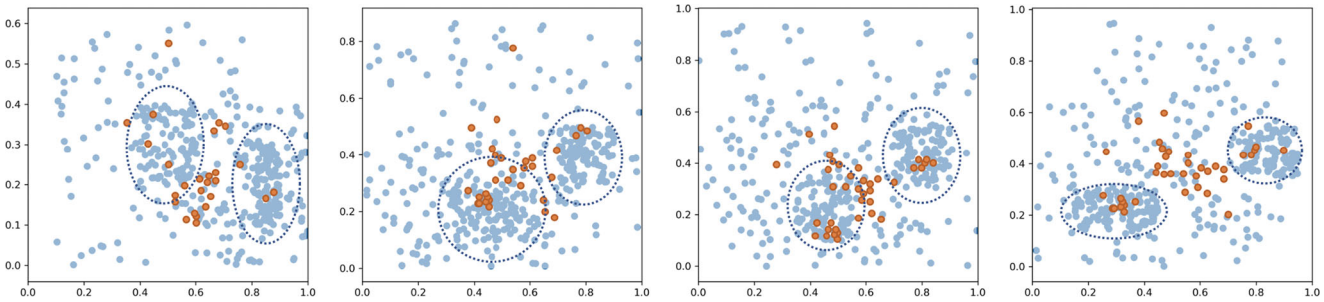


Fig. 7 The important samples in four sepsis distribution buffers (2,3,4,5 in Fig. 8)

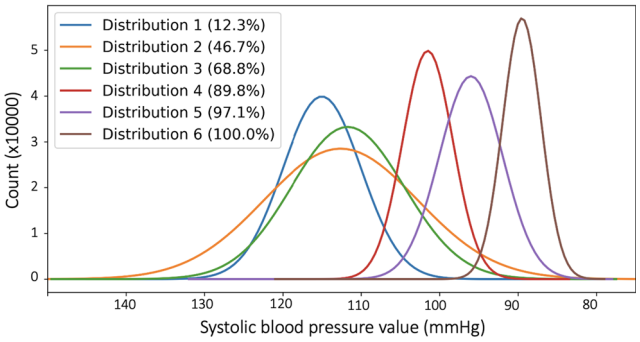


Fig. 8 Extracted six distributions in SEPSIS dataset

set is much lower than that on the training set. Mark ↓ means the accuracy is greatly reduced over 5%.

5.2.3 Ablation study

Both adaptive multi-distribution extraction policy and adaptive importance-based replay policy are necessary as shown in Fig. 6. The adaptive multi-distribution extraction performs best in overall data and early distribution. It can avoid the catastrophic forgetting of the method that trains the model at every time step; The adaptive importance-based replay also has the best performance, it can avoid the overfitting of all relays. Besides, the accuracy of importance-based replay is higher than regularization, which demonstrates a good fit between the two policies of ACCTS (Figs. 7 and 8).

ACCTS has two definable coefficients α and ϵ, which belong to two policies separately. Larger α review more distribution to learn. Larger ϵ causes more samples to review. As shown in Fig. 9, the practice is to set them in the direct ratio: Within a reasonable range, more distributions need more review.

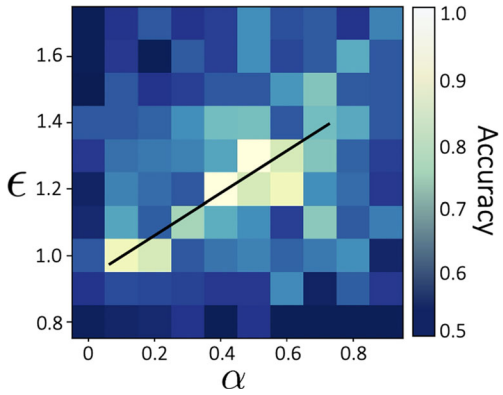


Fig. 9 Classification accuracy with setting different α, ϵ

5.2.4 Multi-distribution and important samples

The case study of sepsis dataset in Fig. 8 shows that ACCTS only extracts six distributions and the difference among distributions is relatively large. The extraction is concentrated in 85%-length late stage, which may be because the patient's vital signs change significantly near the outcome time.

The important samples include not only the data hard to learn but also the representative data as shown in Fig. 7. It might be because that, the representative data is similar to the most common data, resulting in a greater additive loss, therefore leading to smaller coefficients in (8).

5.2.5 Case study

Figure 10(a) shows that the noise ratio is positively correlated with the gradient fluctuation, and the fluctuation is negatively correlated with classification accuracy. Thus, the dynamic change of gradient can reflect the adaptability of the model to the training data. Figure 10(b) and (d) shows that ACCTS can divide the original distribution into multiple distributions with less intersection. For example, in the COVID-19 dataset of Fig. 10(d), ACCTS adaptively divides original data into 4 subsets with a smaller cross-section. Figure 10(b) shows that the distribution differences between early non-sepsis and later sepsis, and between early sepsis and later non-sepsis, are larger than the original difference. Meanwhile, the data revolution is different in different distributions. Distributions in Fig. 10(c) focus on the rise, fall, up-turn, and down-turn of systolic blood pressure, respectively.

6 Conclusion

In this paper, we propose a new concept of Continuous Classification of Time Series (CCTS) to meet real needs. It has two major difficulties of catastrophic forgetting and over-fitting. In CCTS, the multi-distribution of time series is not clearly defined, and the distribution division directly affects the above two difficulties. Thus, we design an adaptive model training strategy named ACCTS. It contains a multi-distribution extraction policy adaptive to the time series evaluation and the model change, and an importance-based replay policy adaptive to the data features and final accuracy. We test the methods on four real-world datasets and analyze them from perspectives of accuracy, continuous learning, ablation study, parameter setting, and case study. The future work will deeply explore the relations between different data distributions, study the safety requirements of medical scenarios and other applications.

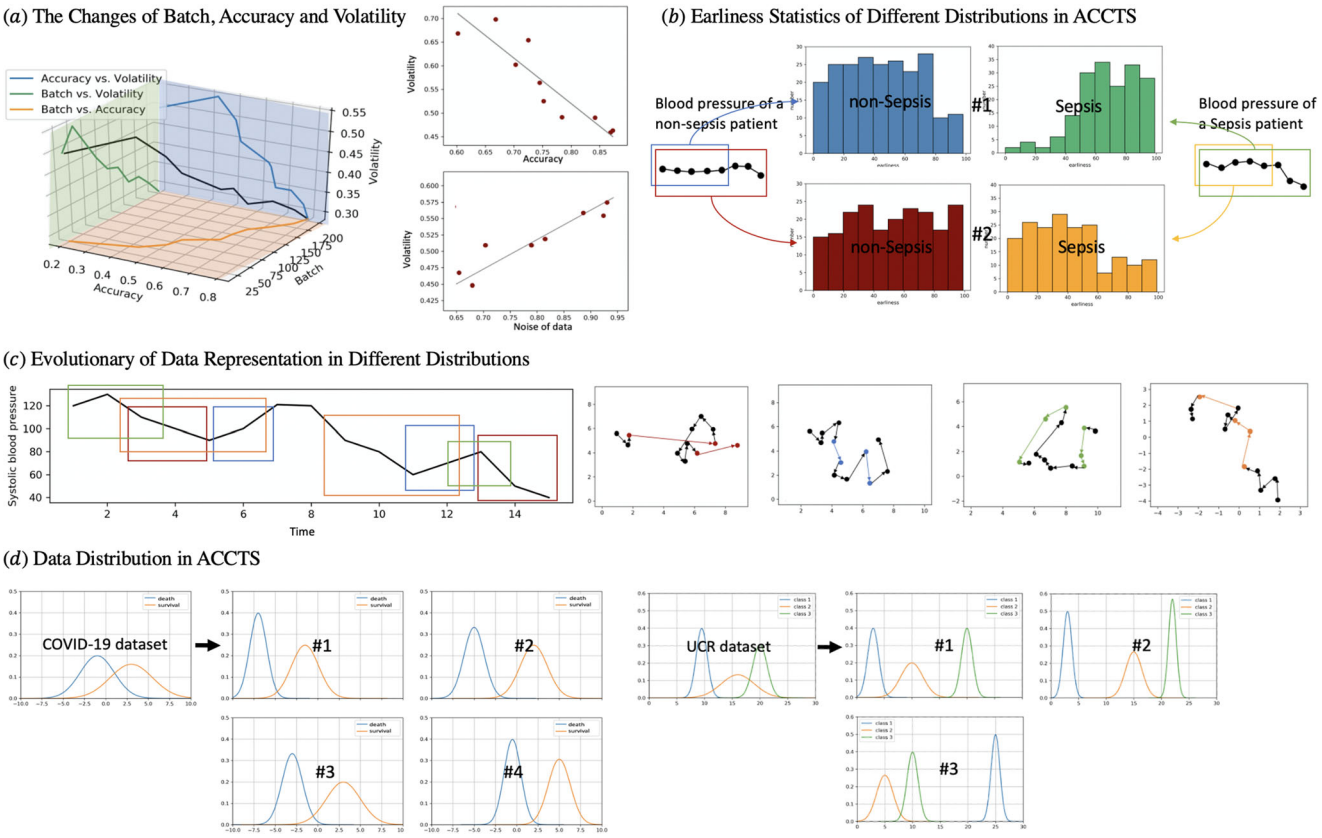


Fig. 10 Cases of Model Change and Data Distribution Change during Model Training Process by ACCTS. (a) shows the changes in classification accuracy, and model stability with different batch sizes during continuous training; (b) shows time and value characteristics in different distributions in the sepsis dataset. For example, in the late distribution, the blood pressure statistic of sepsis is lower. (c) shows the changes in the representation of different characteristics (rise, fall, up-turn, and down-turn) of time series during model training. For example, there is a large change between the early and late representation of the fall in blood pressure. (d) shows the degree to which the model distinguishes categories in different distributions. The value here are the same as those in Fig. 5

Appendix A: Related work and concepts

Time series is one of the most common data forms, the popularity of time series classification has attracted increasing attention in many practical fields, such as healthcare and industry. In the real world, many applications require classification at every time. For example, in the Intensive Care Unit (ICU), critical patients' vital signs develop dynamically, the status perception and disease diagnosis are needed at any time. Timely diagnosis provides more opportunities to rescue lives. In response to the current demand, we propose a new task – Continuous Classification of Time Series (CCTS). It aims to classify as accurately as possible at every time in time series.

Currently, some sub-disciplines also study the mode of continuous learning or continuous classification. But their setting does not match our needs and their methods can't address our issues. As shown in Fig. 11, Online Learning (OL) [34] models the incoming data steam continuously to solve an overall optimization problem with the partially observed data. It focuses more on issues in data steam, rather than the dynamics of time series. OL cannot meet the Requirement 1, 2, 4; Continual Learning (CL) [8] enables the model to learn new tasks over time without forgetting the old tasks. In its setting, the model learns a new task at every moment. The old task and new task are clear so that the multi-distribution is fixed. While the dynamic time series has data correlation over time, which easily further causes the overfitting problem. CL cannot meet the Requirement 2 and partial Requirement 1; Anomaly Detection (AD) [35] identifies data that does not conform to the expected pattern. It mainly maintains one data distribution and gives an alarm when an exception occurs. AD cannot meet Requirement 1 and partial Requirement 2. Because the existing research can not meet the current demand, we propose a new task CCTS. The existing work can be summarized into two categories: Single-shot Classification, Continuous Classification.

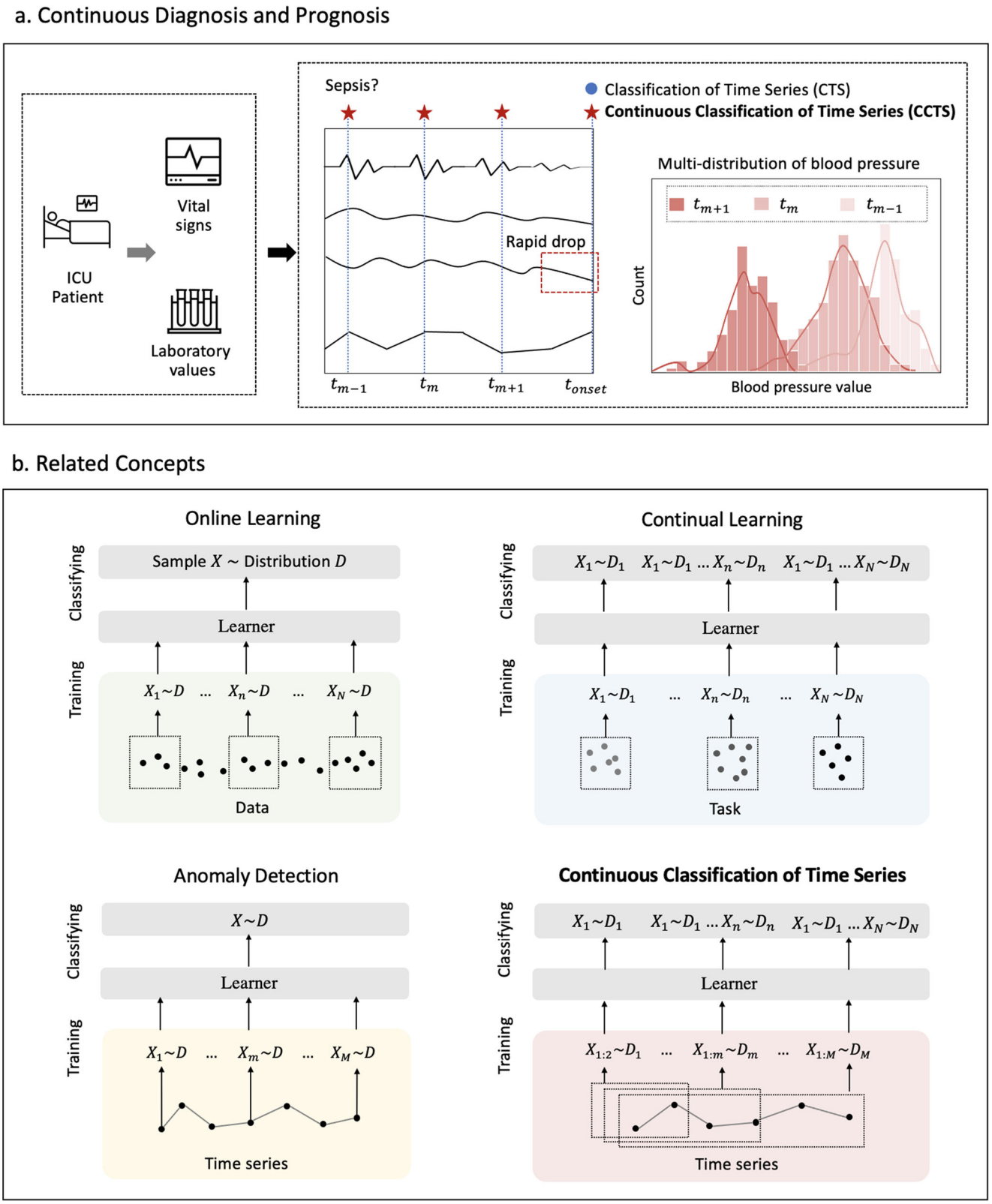


Fig. 11 Continuous Classification of Time Series (CCTS) differences and similarities between CCTS and other concepts

A.1: Single-shot classification

Classifying at a fixed time. *A time series* $X = \{x_1, ...x_T\}$ *is labeled with classes* C. *Single-shot classification aims to classify* X *at a time* $t, t \leq T$ *with the minimum loss* $\mathcal{L}(f(X_{1:t}), C)$.

The foundation is the Classification of Time Series (CTS), making classification based on the full-length data [2]. But in time-sensitive applications, Early Classification of Time Series (ECTS), classifying at an early time, is more critical [3]. For example, early diagnosis helps for sepsis outcomes [15]. Nowadays, Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have shown good performances for CTS and ECTS by modeling long-term dependencies [17], addressing data irregularities [19], learning frequency features [22], etc.

Definition 4 (Classification of Time Series (CTS)) A dataset of time series $\mathcal{D} = \{(X^n, C^n)\}_{n=1}^N$ has N samples. Each time series X^n is labeled with a class C^n, CTS classifies time series using the full-length data by model $f : f(X) \rightarrow C$

Definition 5 (Early Classification of Time Series (ECTS)) A dataset of time series $\mathcal{D} = \{(X^n, C^n)\}_{n=1}^N$ has N samples. Each time series $X^n = X_t{}_{t=1}^T$ is labeled with a class C^n. ECTS classifies time series in an advanced time t by model $f : f(\{x_1, x_2, ..., x_t\}) \rightarrow C$, where $t < T$.

The existing (early) classification of time series is the single-shot classification, where the classification is performed only once at the final or an early time. However, many real-world applications require continuous classification. For example, intensive care patients should be detected and diagnosed at all times to facilitate timely life-saving. The above methods only classify once and just lean a single data distribution. They have good performances on i.i.d data at a fixed time, like early 6 hours sepsis diagnosis [21], but fail for multi-distribution. In fact, continuous classification is composed of multiple single-shot classifications as shown in Fig. 11.

A.2: Continuous classification

Classifying at every time. *A time series is* $X = \{x_1, ...x_T\}$. *At time* t, $x_{1:t}$ *is labeled with class* c_t. *Continuous Classification classifies* $x_{1:t}$ *at every time* $t = 1, ..., T$ *with the minimum loss* $\sum_{t=1}^T \mathcal{L}(f(x_{1:t}), c^t)$.

Most methods use multi-model to learn multi-distribution, like SR [23] and ECEC [24]. They divide data by time stages and design different classifiers for different distributions. But the operation of data division and classifier selection will cause additional losses.

In fact, CCTS is composed of multiple ECTS and the continuous classification is composed of multiple single-shot classification.

Definition 6 (Continuous Classification of Time Series (CCTS)) A dataset of time series $\mathcal{D} = \{(X^n, C^n)\}_{n=1}^N$ has N samples. Each time series $X^n = X_t{}_{t=1}^T$ is labeled with a class C^n. CCTS classifies time series in every time t by model $f : f(\{x_1, x_2, ..., x_t\}) \rightarrow C$, where $t = 1, ..., T$.

Currently, some sub-disciplines also study the mode of continuous learning or continuous classification. But their setting does not match our needs and their methods can't address our issues. As shown in Fig. 11, Online Learning (OL) [34] models the incoming data steam continuously to solve an overall optimization problem with the partially observed data. It focuses more on issues in data steam, rather than the dynamics of time series. Thus, OL cannot meet the Requirement 1, 2, 3; Continual Learning (CL) [8] enables the model to learn new tasks over time without forgetting the old tasks. In its setting, the model learns a new task at every moment. The old task and new task are clear so that the multi-distribution is fixed. While the dynamic time series has data correlation over time, which easily further causes the overfitting problem. Thus, CL cannot meet the Requirement 2 and partial Requirement 1; Anomaly Detection (AD) [35] identifies data that does not conform to the expected pattern. It mainly maintains one data distribution and gives an alarm when an exception occurs. Thus, AD cannot meet Requirement 1 and partial Requirement 2. Because the existing research can not meet the current demand, we propose a new concept CCTS.

Definition 7 (Online Learning (OL)) A OL issue has a sequence of dataset $\mathcal{X} = \{X^1, X^2, ..., X^N\}$ for one task $\mathcal{T}$. Each dataset X^t has a distribution D^t. CL learns a new D^t at every time t. The goal is to find the optimal solution of $\mathcal{T}$ after N iterations by minimize the regret $\mathcal{R} := \sum_{t=1}^N (f^t(X^t) - \min f^t(X^t))$.

Definition 8 (Continual Learning (CL)) A CL issue $\mathcal{T} = \{T^1, T^2, ..., T^N\}$ has a sequence of N tasks. Each task $T^n = (X^n, C^n)$ is represented by the training sample X^n with classes C^n. CL learns a new task at every moment. The goal is to control the statistical risk of all seen tasks $\sum_{n=1}^N \mathbb{E}_{(X^n, C^n)}[\mathcal{L}(f_n((X^n; \theta), C^n)]$ with loss $\mathcal{L}$, network function f_n and parameters θ.

Appendix B: Using different DL models as backbone networks for ACCTS

In ACCTS, the State $\mathcal{S}$ and the classifier are based on LSTM as we are dealing with time series with unequal lengths. Meanwhile, RNN-based models have the embedded state representation, which can be more easily used to reinforcement learning strategies, as shown in (2).

CNN-based models and Transformer-based models can also model time series data. But they prefer to deal with sequence with equal length. And there is no explicit hidden state of data.

But in order to verify the effectiveness of the dynamic data division strategy for CCTS, we have tested ACCTS by using LSTM, CNN, and Transformer as Backbone Networks.

B.1: Backbone networks

$$s_t = \text{concatenate}(\text{NN}(x_t), \text{MG}(X_{1:t})) \tag{11}$$

- LSTM: $\text{NN}(x_t) = \text{LSTM}(x_t)$; Classifier net f = LSTM with parameter θ_f.

$$\begin{aligned} \text{LSTM}(x_t) &= h_t = o_t \cdot \eta(c_t) \\ c_t &= f_t \cdot c_{t-1} + i_t \cdot \eta(W_c[h_{t-1}, x_t] + b_c) \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \\ i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \end{aligned} \tag{12}$$

- CNN: $\text{NN}(x_t) = \text{CNN}(x_{1:t})$ (feature in the last fully-connected layer); Classifier net f = CNN with parameter θ_f.

$$\begin{aligned} \text{CNN}(x_{1:t}) &= fc(h_t) \\ h_t &= (K * x_{1:t})^{\times n} \end{aligned} \tag{13}$$

- Transformer: $\text{NN}(x_t) = \text{Transformer}(x_{1:t})$ (feature in the output layer); Classifier net f = Transformer with parameter θ_f.

$$\begin{aligned} \text{Transformer}(x_{1:t}) &= fc(h_t) \\ h_t &= \text{Self-Attention}(x_{1:t}) \end{aligned} \tag{14}$$

B.2: Datasets and baselines

We use 4 real-world datasets. For each time series in the four datasets, every time point is tagged with a class label, which is the same as its outcome label. The 10 time points are the window sizes for CNN and Transformer.

- UCR Earthquake Prediction [49] UCR-EQ.
- USHCN Climate Prediction [51] USHCN.
- COVID-19 Mortality Prediction [45] COVID-19.
- Physionet 2019 Sepsis Prediction [47] SEPSIS.

The baselines are mainly composed of two categories as we introduced in Section 2. SR has multi-model structure; GEM and CLOP have CL strategies.

- SR [23]. It has multiple basic classification models. All models are trained by the full-length time series. The final classification is the fusion result.
- GEM [29]. The strategy trains a model to remember the old tasks by finding the new gradients which are at acute angles to the old gradients.
- CLOPS [14]. The strategy trains a base model by replaying old tasks with importance-guided buffer storage and uncertainty-based buffer acquisition.

B.3: Results of continuous classification

ACCTS has the best performance on classification accuracy. As shown in Table 2, it can classify time series more accurately than all baselines at every time. The average accuracy is about 2% higher. Specifically, ACCTS is significantly better than baselines in Bonferroni-Dunn tests: Rank(baselines) $= 4.5 > 1.80+1(k = 7, n = 4, m = 5)$. k, n, m are the number of methods, datasets, cross-validation fold, $CD = q_\alpha\sqrt{\frac{k(k+1)}{6(nm)}}$, if the average rank of baselines higher $CD + 1$, the result is significantly improved. The accurate continuous classification is important for time-sensitive applications. Take continuous sepsis diagnosis and prognosis in ICU as an example, compared with the best baseline, our method improves the accuracy by 1.32% on average, 2.19% in the early 50% time stage when the key features are unobvious. Each hour of delayed treatment increases sepsis mortality by 4-8% [48]. With the same accuracy, we can predict 0.951 hours in advance.

In fact, our method is actually a strategy that can be used on different basic models. We apply the method to different basic models. The dynamic data division strategy can improve the performance of RNN-based models (LSTM, GRU), CNN-based model, and Transformer-based model in CCTS task as shown in Table 5. ACCTS[1], ACCTS[1*], ACCTS[2], ACCTS[3] have more accurate results than LSTM, GRU, CNN, Transformer at every time points (Fig. 12).

We are dealing with time series with unequal lengths, thus using the RNN-based model. Meanwhile, RNN-based models have the embedded state representation, which can be more easily used to reinforcement learning strategies, as shown in (2) and 3. Therefore, we can also use the GRU model. On our datasets, LSTM and GRU perform similarly, and LSTM performs relatively well as shown in Table 5. Meanwhile, CNN-based models and Transformer-based models can also model time series data. But they prefer to deal with the sequence with equal length. And there is no explicit hidden state of data. And when we use

Table 5 Classification Accuracy (AUC-ROC↑) of Baselines on 4 Real-world Datasets. Trans: Transformer; ACCTS[1] means that LSTM is the backbone network; ACCTS[1]* means that GRU is the backbone network; ACCTS[2] means that CNN is the backbone network; ACCTS[3] means that Transformer is the backbone network. * -n means The n-th last time point

	*	-9	-8	-7	-6	-5	-4	-3	-2	-1
UCR-EQ	SR	.736±.01	.830±.01	.863±.01	.871±.02	.888±.01	.924±.01	.928±.10	.936±.10	.941±.10
	GEM	.767±.01	.850±.01	.876±.01	.890±.02	.900±.01	.920±.01	.929±.00	.935±.00	.934±.00
	CLOPS	.773±.01	.855±.01	.878±.01	.896±.02	.902±.01	.915±.01	.917±.00	.921±.00	.925±.00
	LSTM	.711±.03	.803±.02	.843±.01	.854±.01	.874±.01	.913±.03	.909±.01	.919±.00	.924±.01
	ACCTS[1]	**.774±.02**	**.856±.01**	**.882±.02**	**.900±.01**	**.906±.00**	**.928±.00**	**.933±.01**	**.940±.00**	**.946±.00**
	GRU	.713±.01	.807±.04	.845±.02	.856±.02	.873±.02	.910±.02	.909±.02	.916±.01	.923±.01
	ACCTS[1]*	**.775±.02**	**.857±.02**	**.879±.03**	**.902±.02**	**.906±.01**	**.926±.02**	**.930±.03**	**.941±.03**	**.944±.02**
	CNN	.708±.01	.797±.04	.840±.02	.846±.04	.870±.03	.902±.01	.905±.00	.912±.02	.921±.01
	ACCTS[2]	**.770±.02**	**.843±.05**	**.868±.04**	**.894±.03**	**.899±.02**	**.918±.03**	**.926±.04**	**.938±.03**	**.942±.02**
	Trans	.709±.02	.794±.05	.842±.03	.843±.05	.873±.03	.910±.03	.9150±.04	.915±.02	.922±.02
	ACCTS[3]	**.770±.02**	**.843±.05**	**.860±.04**	**.8984±.05**	**.903±.05**	**.920±.03**	**.928±.04**	**.938±.05**	**.942±.03**
USHCN	SR	.730±.02	.745±.01	.761±.02	.809±.02	.836±.01	.886±.02	.902±.01	.921±.02	.933±.00
	GEM	.728±.02	.772±.01	.781±.02	.801±.02	.838±.01	.868±.02	.899±.01	.910±.02	.928±.00
	CLOPS	.740±.02	.769±.01	.781±.02	.800±.02	.835±.01	.861±.02	.877±.01	.895±.01	.919±.01
	LSTM	.700±.02	.721±.01	.745±.02	.784±.02	.820±.01	.837±.02	.852±.01	.869±.02	.891±.00
	ACCTS[1]	**.742±.01**	**.775±.01**	**.791±.02**	**.810±.01**	**.841±.01**	**.898±.02**	**.910±.01**	**.928±.01**	**.939±.01**
	GRU	.701±.02	.724±.01	.744±.01	.785±.03	.821±.02	.836±.01	.850±.02	.867±.02	.892±.00
	ACCTS[1]*	**.745±.03**	**.774±.03**	**.795±.04**	**.813±.03**	**.840±.01**	**.899±.02**	**.905±.02**	**.923±.02**	**.934±.01**
	CNN	.690±.03	.709±.03	.735±.03	.774±.003	.818±.02	.835±.01	.850±.02	.868±.02	.889±.02
	ACCTS[2]	**.740±.03**	**.764±.03**	**.793±.03**	**.810±.04**	**.838±.02**	**.895±.03**	**.902±.03**	**.920±.02**	**.932±.01**
	Trans	.692±.03	.719±.03	.736±.02	.777±.004	.820±.02	.837±.01	.848±.02	.865±.02	.888±.02
	ACCTS[3]	**.741±.03**	**.766±.05**	**.794±.03**	**.815±.04**	**.841±.03**	**.896±.03**	**.900±.04**	**.922±.01**	**.935±.01**
COVID-19	SR	.730±.02	.810±.01	.867±.01	.901±.01	.900±.01	.935±.01	.946±.00	.952±.01	.962±.00
	GEM	.779±.01	.871±.01	.885±.02	.914±.01	.924±.01	.936±.00	.939±.01	.949±.01	.953±.00
	CLOPS	.775±.01	.869±.01	.900±.01	.918±.02	.925±.01	.935±.01	.940±.00	.947±.00	.954±.00
	LSTM	.701±.03	.793±.02	.833±.01	.844±.01	.888±.01	.918±.03	.925±.01	.939±.00	.944±.01
	ACCTS[1]	**.790±.02**	**.872±.01**	**.901±.02**	**.919±.01**	**.927±.00**	**.955±.00**	**.960±.01**	**.963±.00**	**.967±.00**
	GRU	.700±.03	.794±.02	.834±.01	.845±.02	.885±.02	.915±.02	.922±.02	.935±.01	.942±.02
	ACCTS[1]*	**.791±.02**	**.875±.01**	**.900±.02**	**.915±.01**	**.924±.02**	**.953±.01**	**.959±.01**	**.961±.01**	**.965±.01**
	CNN	.690±.05	.791±.05	.830±.04	.838±.04	.882±.03	.912±.04	.920±.01	.932±.04	.939±.04
	ACCTS[2]	**.788±.04**	**.870±.04**	**.895±.05**	**.912±.02**	**.919±.04**	**.949±.05**	**.956±.03**	**.957±.04**	**.964±.03**
	Trans	.693±.05	.793±.05	.831±.04	.837±.04	.885±.04	.914±.04	.921±.02	.936±.05	.941±.04
	ACCTS[3]	**.791±.04**	**.872±.03**	**.896±.05**	**.915±.02**	**.921±.05**	**.945±.05**	**.957±.03**	**.956±.05**	**.964±.04**
SEPSIS	SR	.659±.01	.768±.01	.791±.02	.803±.01	.827±.03	.835±.01	.845±.01	.859±.02	.866±.02
	GEM	.730±.02	.802±.01	.826±.03	.834±.02	.836±.02	.841±.03	.849±.01	.851±.01	.853±.01
	CLOPS	.733±.02	.802±.01	.824±.03	.830±.02	.838±.02	.842±.03	.850±.01	.853±.01	.857±.01
	LSTM	.629±.03	.735±.06	.736±.06	.745±.05	.748±.04	.773±.03	.795±.02	.813±.02	.827±.03
	ACCTS[1]	**.734±.03**	**.812±.02**	**.828±.03**	**.835±.02**	**.842±.03**	**.852±.02**	**.857±.01**	**.866±.01**	**.872±.01**
	GRU	.631±.03	.736±.05	.737±.05	.747±.04	.751±.03	.772±.04	.793±.01	.814±.02	.826±.03
	ACCTS[1]*	**.735±.04**	**.814±.03**	**.829±.04**	**.834±.05**	**.840±.04**	**.851±.05**	**.855±.04**	**.864±.04**	**.870±.04**
	CNN	.625±.04	.734±.04	.730±.04	.743±.03	.745±.06	.770±.03	.792±.02	.812±.02	.825±.04
	ACCTS[2]	**.724±.03**	**.810±.03**	**.825±.04**	**.832±.04**	**.839±.02**	**.850±.03**	**.854±.04**	**.863±.05**	**.869±.02**
	Trans	.626±.06	.736±.05	.733±.05	.742±.05	.749±.05	.772±.04	.793±.03	.815±.06	.829±.05
	ACCTS[3]	**.726±.03**	**.812±.04**	**.829±.06**	**.835±.06**	**.842±.04**	**.853±.04**	**.855±.04**	**.865±.05**	**.871±.03**

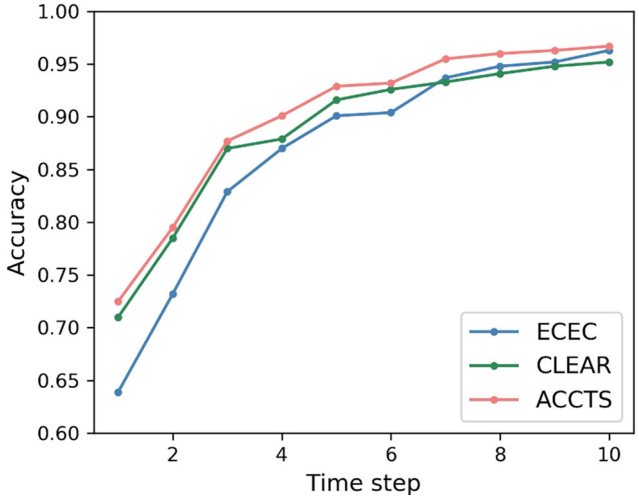


Fig. 12 Classification accuracy of SOTA ECTS, CL, and CCTS methods

CNN and Transformer, the time window has to be set in advance. We set 10 time points in experiments.

Author Contributions C.S. and H.L. conceived the project. C.S. and S.H contributed ideas, designed and conducted the experiments. S.H, H.L., M.S, D.C., B,Z evaluated the experiments. All authors co-wrote the manuscript.

Funding This work was supported by the National Natural Science Foundation of China (No.62172018, No.62102008), and the National Key Research and Development Program of China under Grant 2021YFE0205300.

Data Availability All datasets are publicly available (See references). Correspondence and requests for materials should be addressed to Chenxi Sun, Hongyan Li, and Shenda Hong.

Declarations

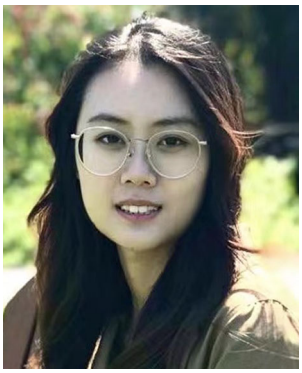
Conflict of Interests No potential conflict of interest was reported by the authors.

References

1. Santos T, Kern R (2016) A literature survey of early time series classification and deep learning. In: Proceedings of the 1st international workshop on science, application and methods in industry 4.0 Co-located with (i-KNOW 2016), Graz, Austria, October 19, 2016
2. Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller P (2019) Deep learning for time series classification: a review. Data Min Knowl Discov 334:917–963
3. Gupta A, Gupta HP, Biswas B, Dutta T (2020) Approaches and applications of early classification of time series: a review. IEEE Trans Artif Intell 11:47–61
4. Shim D, Mai Z, Jeong J, Sanner S, Kim H, Jang J (2021) Online class-incremental continual learning with adversarial shapley value. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, virtual event, february 2-9, 2021, pp 9630–9638
5. Chen W, Wang J, Fe Ng QL, Xu SC, Ba L (2014) The treatment of severe and multiple injuries in intensive care unit: report of 80 cases. European Review for Medical & Pharmacological Sciences 1824:3797
6. Seymour CW, Gesten F, Prescott HC, Friedrich ME, Iwashyna TJ, Phillips GS, Lemeshow S, Osborn T, Terry KM, Levy MM (2017) Time to treatment and mortality during mandated emergency care for sepsis. N Engl J Med 2235
7. Parisi GI, Kemker R, Part JL, Kanan C, Wermter S (2019) Continual lifelong learning with neural networks: a review. Neural Netw 113:54–71
8. Delange M, Aljundi R, Masana M, Parisot S, Jia X, Leonardis A, Slabaugh G, Tuytelaars T (2021) A continual learning survey: Defying forgetting in classification tasks. IEEE Trans Pattern Anal Mach Intell 1–1
9. Sun C, Song M, Cai D, Zhang B, Hong S, Li H (2022) Confidence-guided learning process for continuous classification of time series. In: The 31st ACM international conference on information and knowledge management (CIKM '22), october 17–21, 2022, atlanta, GA, USA. ACM, New York, NY, USA, p 5. https://doi.org/10.1145/3511808.3557565
10. Sun C, Li H, Song M, Cai D, Zhang B, Hong S (2022) Continuous diagnosis and prognosis by controlling the update process of deep neural networks. arXiv:2210.02719. https://doi.org/10.48550/arXiv.2210.02719
11. Saha G, Garg I, Roy K (2021) Gradient projection memory for continual learning. In: 9Th international conference on learning representations, ICLR 2021, virtual event, austria, may 3-7, 2021
12. Xing Z, Pei J, Yu PS, Wang K (2011) Extracting interpretable features for early classification on time series. In: Proceedings of the 2011 SIAM international conference on data mining, pp 247–258
13. Liang Z, Wang H (2021) Efficient class-specific shapelets learning for interpretable time series classification. Inf Sci 570:428–450
14. Kiyasseh D, Zhu T, Clifton D (2021) A clinical deep learning framework for continually learning from cardiac signals across diseases, time, modalities, and institutions. Nat Commun 121:4221
15. Liu B, Li Y, Sun Z, Ghosh S, Ng K (2018) Early prediction of diabetes complications from electronic health records: a multi-task survival analysis approach. In: Proceedings of the Thirty-Second AAAI conference on artificial intelligence, 2018, pp 101–108
16. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: CVPR, vol 1, p 3
17. Choi E, Schuetz A, Stewart WF, Sun J (2017) Using recurrent neural network models for early detection of heart failure onset. J Am Med Inform Assoc 242:361–370
18. Tan Q, Ye M, Yang B, Liu S, Ma AJ, Yip TC, Wong GL, Yuen PC (2020) DATA-GRU: Dual-attention time-aware gated recurrent unit for irregular multivariate time series. In: The thirty-fourth AAAI conference on artificial intelligence, new york, NY, USA, February 7-12, 2020, pp 930–937
19. Sun C, Hong S, Song M, Chou Y-H, Sun Y, Cai D, Li H (2021) Te-esn: Time encoding echo state network for prediction based on irregularly sampled time series data. In: Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI-21, pp 3010–3016, https://doi.org/10.24963/ijcai.2021/414
20. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Bonet B, Koenig S (eds) Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin. Texas, USA, pp 2267-2273
21. Reyna MA, Josef CS, Jeter R, Shashikumar SP, Sharma A (2019) Early prediction of sepsis from clinical data: The physionet/computing in cardiology challenge 2019. Crit Care Med 482:1
22. Hsu E, Liu C, Tseng VS (2019) Multivariate time series early classification with interpretability using deep learning and

attention mechanism. In: Yang Q, Zhou Z, Gong Z, Zhang M, Huang S (eds) Advances in knowledge discovery and data mining - 23rd pacific-asia conference, PAKDD 2019, macau, china, april 14-17, 2019, proceedings, Part III. Lecture notes in computer science, vol 11441, pp 541–553
23. Mori U, Mendiburu A, Dasgupta S, Lozano JA (2018) Early classification of time series by simultaneously optimizing the accuracy and earliness. IEEE Trans Neural Networks Learn Syst 2910:4569–4578
24. Lv J, Hu X, Li L, Li P (2019) An effective confidence-based early classification of time series. IEEE Access 7:96113–96124
25. Rolnick D, Ahuja A, Schwarz J, Lillicrap TP, Wayne G (2019) Experience replay for continual learning. In: Wallach HM, Larochelle H, Beygelzimer A, d'Alché-Buc F, Fox EB, Garnett R (eds) Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurIPS 2019, december 8-14, 2019, Vancouver, BC, Canada, pp 348–358
26. Isele D, Cosgun A (2018) Selective experience replay for lifelong learning. In: McIlraith SA, Weinberger KQ (eds) Proceedings of the thirty-second AAAI conference on artificial intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018, pp 3302–3309
27. Rebuffi S, Kolesnikov A, Sperl G (2017) Icarl: Incremental classifier and representation learning. In: Lampert CH (ed) 2017 IEEE Conference on computer vision and pattern recognition, CVPR 2017, honolulu, HI, USA, July 21-26, 2017, pp 5533–5542
28. Kirkpatrick J, Pascanu R, Rabinowitz NC, Veness J, Desjardins G, Rusu AA, Milan K, Quan J, Ramalho T, Grabska-Barwinska A, Hassabis D, Clopath C, Kumaran D, Hadsell R (2016) Overcoming catastrophic forgetting in neural networks. arXiv:1612.00796
29. Lopez-Paz D, Ranzato M (2017) Gradient episodic memory for continual learning. In: Advances in neural information processing systems 30: Annual conference on neural information processing systems 2017, december 4-9, 2017, long beach, CA, USA, pp 6467–6476
30. Liu X, Masana M, Herranz L, van de Weijer J, López AM, Bagdanov AD (2018) Rotate your networks: Better weight consolidation and less catastrophic forgetting. In: 24Th international conference on pattern recognition, ICPR 2018, Beijing, China, august 20-24, 2018, pp 2262–2268
31. Zhang J, Zhang J, Ghosh S, Li D, Tasci S, Heck LP, Zhang H, Kuo C-J (2020) Class-incremental learning via deep model consolidation. In: IEEE Winter conference on applications of computer vision, WACV 2020, snowmass village, CO, USA, March 1-5, 2020, pp 1120–1129
32. Fernando C, Banarse D, Blundell C, Zwols Y, Ha D, Rusu AA, Pritzel A, Wierstra D (2017) Pathnet: Evolution channels gradient descent in super neural networks. arXiv:1701.08734
33. Mallya A, Lazebnik S (2018) Packnet: Adding multiple tasks to a single network by iterative pruning. In: 2018 IEEE Conference on computer vision and pattern recognition, CVPR 2018, salt lake city, UT, USA, June 18-22, 2018, pp 7765–7773
34. Yuanyu WEA (2021) Projection-free online learning in dynamic environments. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, virtual event, february 2-9, 2021, pp 10067–10075
35. Fernando T, Gammulle H, Denman S, Sridharan S, Fookes C (2022) Deep learning for medical anomaly detection - A survey. ACM Comput Surv 547:141–114137
36. Ma Q, Chen C, Li S, Cottrell GW (2021) Learning representations for incomplete time series clustering. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, virtual event, february 2-9, 2021. AAAI Press, pp 8837–8846
37. Chen IY, Krishnan RG, Sontag DA (2022) Clustering interval-censored time-series for disease phenotyping. In: Thirty-sixth AAAI conference on artificial intelligence, AAAI 2022, thirty-fourth conference on innovative applications of artificial intelligence, IAAI 2022, the twelveth symposium on educational advances in artificial intelligence, EAAI 2022 virtual event, february 22 - march 1, 2022. AAAI Press, pp 6211–6221
38. Kaelbling LP, Littman ML, Cassandra AR (1995) Partially observable markov decision processes for artificial intelligence. In: Dorst, l., van lambalgen, m., voorbraak, f. (eds.) reasoning with uncertainty in robotics, international workshop, RUR '95, amsterdam, the netherlands, december 4-6, 1995, proceedings. Lecture notes in computer science, vol 1093, pp 146–163
39. Srinivas S, Fleuret F (2019) Full-gradient representation for neural network visualization. In: Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurIPS 2019, december 8-14, 2019, vancouver, BC, Canada, pp 4126–4135
40. Sutton RS, McAllester DA, Singh SP, Mansour Y (1999) Policy gradient methods for reinforcement learning with function approximation. In: Advances in neural information processing systems 12, [NIPS conference, denver, colorado, USA, November 29 - December 4, 1999], pp 1057–1063
41. Zhang S, Boehmer W, Whiteson S (2019) Generalized off-policy actor-critic. In: Advances in neural information processing systems 32: Annual conference on neural information processing systems 2019, neurIPS 2019, december 8-14, 2019, vancouver, BC, Canada, pp 1999–2009
42. Watkins CJ, Dayan P (1992) Q-learning. Machine learning 83-4:279–292
43. Borsos Z, Mutny M, Krause A (2020) Coresets via bilevel optimization for continual learning and streaming. In: Advances in neural information processing systems 33: Annual conference on neural information processing systems 2020, neurIPS 2020, december 6-12, 2020, virtual
44. Mazumder P, Singh P, Rai P (2021) Few-shot lifelong learning. In: Thirty-fifth AAAI conference on artificial intelligence, virtual event, february 2-9, 2021, pp 2337–2345
45. Yan L, Zhang HT, Goncalves J et al (2020) An interpretable mortality prediction model for COVID-19 patients. Nat Mach Intell 2:283–288.
46. Sun C, Hong S, Song M, Li H, Wang Z (2020) Predicting covid-19 disease progression and patient outcomes based on temporal deep learning. BMC Med Inform Decis Mak 21:45
47. Reyna MA, Josef C, Seyedi S, Jeter R, Shashikumar SP, Westover MB, Sharma A, Nemati S, Clifford GD (2019) Early prediction of sepsis from clinical data: the physionet/computing in cardiology challenge 2019. In: 46Th computing in cardiology, cinc 2019, singapore, september 8-11, 2019, pp 1–4, https://doi.org/10.23919/CinC49843.2019.9005736
48. Seymour CW, Gesten F, Prescott HC, Friedrich ME, Iwashyna TJ, Phillips GS, Lemeshow S, Osborn T, Terry KM, Levy MM (2017) Time to treatment and mortality during mandated emergency care for sepsis. N Engl J Med 37623:2235–2244
49. Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista G (2015) The UCR Time Series Classification Archive www.cs.ucr.edu/eamonn/time_series_data/
50. Ammon CJ, Velasco AA, Lay T, Wallace TC (2021) Earthquake prediction. Forecasting, & Early Warning 223–248
51. Menne WCM, R V (2016) Long-term daily and monthly climate records from stations across the contiguous United States U.S.Historical Climatology Network)

52. Lee WY, Park SK, Sung HH (2021) The optimal rainfall thresholds and probabilistic rainfall conditions for a landslide early warning system for chuncheon, republic of korea Landslides
53. Wiens J, Horvitz E, Guttag JV (2012) Patient risk stratification for hospital-associated c. diff as a time-series classification task. In: Advances in neural information processing systems, pp 467–475



Chenxi Sun is a Ph.D. candidate at the School of Intelligence Science and Technology, Peking University, Beijing, China. She received her B.S. degree from the School of Computer Science and Technology at Shandong University in 2019. Her main research interests include knowledge discovery, deep learning, and data mining on time series data.

Hongyan Li received her Ph.D. in Computer software and theory from Northwestern Polytechnical University, Xi'an, China, in 1999. She is currently a Professor at the School of Intelligence Science and Technology, Peking University, Beijing, China. Her research interests include big data analysis, knowledge discovery, deep learning, machine learning, and state perception of the complex system.

Moxian Song is a Ph.D. candidate at the School of Intelligence Science and Technology, Peking University. Before that, he obtained his B.S. degree in the School of Electronics and Information at Northwestern Polytechnical University in 2018. His research interests are machine learning and data mining, especially partial label learning and information fusion.

Cai Derun is currently working on his MSc degree at the School of Intelligence Science and Technology, Peking University, Beijing, China. His main research interests include data mining and graph representation learning.

Baofeng Zhang received his BE degree from the School of Computer and Communication Engineering, University of Science and Technology Beijing, China, in 2021. He is currently working on his Ph.D. degree at the School of Intelligence Science and Technology, Peking University, China. His research areas include machine learning, deep learning, and time series analysis.

Shenda Hong is an Assistant Professor in National Institute of Health Data Science at Peking University. Before that, he was a (Boya) Postdoctoral Researcher in National Institute of Health Data Science at Peking University from 2020 to 2022, a Postdoctoral Researcher at Georgia Institute of Technology from 2019 to 2020, a Visiting Researcher at Harvard Medical School in 2020. He obtained his Ph.D. degree from Peking University in 2019, and B.S. degree from Beijing University of Posts and Telecommunications in 2014. His research interests are data mining and artificial intelligence for real-world healthcare data, especially deep learning for temporal medical data.

Affiliations

Chenxi Sun[1,2] · Hongyan Li[1,2] · Moxian Song[1,2] · Derun Cai[1,2] · Baofeng Zhang[1,2] · Shenda Hong[3,4]

Chenxi Sun
sun_chenxi@pku.edu.cn

Moxian Song
songmoxian@pku.edu.cn

Derun Cai
cdr@stu.pku.edu.cn

Baofeng Zhang
boffinzhang@stu.pku.edu.cn

1 School of Intelligence Science and Technology, Peking University, Beijing, China

2 Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing, China

3 National Institute of Health Data Science, Peking University, Beijing, China

4 Institute of Medical Technology, Health Science Center of Peking University, Beijing, China