ABSTRACT
In the real world, the class of a time series is usually labeled at the final time, but many applications require to classify time series at every time point. e.g. the outcome of a critical patient is only determined at the end, but he should be diagnosed at all times for timely treatment. Thus, we propose a new concept: Continuous Classification of Time Series (CCTS). It requires the model to learn data in different time stages. But the time series evolves dynamically, leading to different data distributions. When a model learns multi-distribution, it always forgets or overfits. We suggest that meaningful learning scheduling is potential due to an interesting observation: Measured by confidence, the process of model learning multiple distributions is similar to the process of human learning multiple knowledge. Thus, we propose a novel Confidence-guided method for CCTS (C³TS). It can imitate the alternating human confidence described by the Dunning-Kruger Effect. We define the objective-confidence to arrange data, and the self-confidence to control the learning duration. Experiments on four real-world datasets show that C³TS is more accurate than all baselines for CCTS.

KEYWORDS
classification, time series data, confidence, deep learning

1 INTRODUCTION
Time series classification is critical in many practical fields [25, 32]. Most work gives one-shot classification, classifying at a fixed time, such as Classification of Time Series (CTS) [7] and Early Classification of Time Series (ECTS) [9]. However, in the real world, many time-sensitive applications need to classify time series continuously. For example, in ICU, most detected vital signs change dynamically. The status perception is needed at any time as the real-time diagnosis provides more opportunities to rescue lives [4, 16]. But patient labels, e.g., mortality or morbidity, are only available at the onset time but unknown in early stages. In response to the current demand, we propose a new concept – Continuous Classification of Time Series (CCTS) as shown in Figure 1.

CCTS requires the model to learn data from different stages. Most practical time series has evolved data distribution. For example, distributions of blood pressure of sepsis patients vary among early, middle, and later stages, bringing a triple-distribution. However, the deep learning model is lack of ability to learn all distributions simultaneously as they are restricted by the premise of independently and identically distribution [27]. Frequent learning of new knowledge will inevitably lead to the forgetting of old ones [21], and too much training on one distribution may make the parameter fall into the local solution, resulting in poor generalization [24].
Recently, continual learning [14] aims to model multi-distribution. Its scenario is multiple tasks, where the old and new tasks are clear so that the multi-distribution is fixed and distinctive. But CCTS aims at one task, the distributions are not determined and need to be defined firstly, leading to intertwined problems of forgetting and overfitting. Some methods use multi-model to model multi-distribution [17]. But different division criteria have a great impact on the results. And the operation of classifier selection will result in additional losses. To address the above issues, we suggest that meaningful learning scheduling is potential: A suitable learning order and moderate data replay may avoid forgetting and overfitting. Curriculum learning [3] presents an easy-to-hard order. But time series data is too abstract to define its difficulty and the difficulty of human definition may not match the machine. For example, stable vital signs are more likely to confuse the model and lead to errors.

In fact, the process of model learning multiple distributions is similar to the process of human learning multiple knowledge. People control the learning progress according to their mastery, which is usually assessed by human confidence. Based on this, we use confidence to guide the learning. Different from the confidence in Statistics, the confidence in this paper represents the human cognition of their ability. The human confidence is described by Dunning-Kruger effect curve [12, 29] as shown in Figure 2. It is an alternating process with experiences of ignorance, overconfidence, disappointment and development: When a human learns new knowledge, he will first scratch and grasp the overall framework and therefore has a lot of confidence. Then, he begins to find that he had little in-depth knowledge and loses his confidence. Over time, he studies deeply, becoming more and more experienced and confident.

We propose a novel Confidence-guided method for CCTS (C³TS) to imitate the human learning process: The objective-confidence is based on the importance-based coefficient. It arranges data and makes the model relearn important samples to consolidate memory loosely; The self-confidence is based on the total uncertainty. It controls the learning duration under the current distribution and schedules the learning order among distributions.

## 2 CONFIDENCE-GUIDED METHOD

A dataset \( S \) contains \( N \) time series. Each time series \( X = \{x_1, ..., x_T\} \) is labeled with a class \( C \subset C \) at the final time \( T \). \( X \) in different time stages forms \( M \) distributions \( D = \{D^m\}_{m=1}^M \), each \( D^m \) has subsequence \( X^m \). CCTS learns every \( D^m \) and introduces a task sequence \( M = \{M^m\}_{m=1}^M \) to minimize the additive risk \( \sum_{m=1}^M E[M^m][\mathcal{L}(f^m(D^m; \theta), C)] \). \( f^m \) is the model after learning \( M^m \). When the model learns \( M^m \), its performance on all observed data cannot degrade: \( \frac{1}{m} \sum_{i=1}^m \mathcal{L}(f^i, M^i) \leq \frac{1}{m-1} \sum_{i=1}^{m-1} \mathcal{L}(f^i, M^i) \).

### 2.1 Importance-Based Objective-Confidence

People gain the objective-confidence through regular examinations, where they can know their weak knowledge through test results and learn it again. We introduce an importance-based replay method. In each round, it only re-trained the model by some important samples. The importance of each sample \( X_i \) is learned from the importance coefficient \( \beta_i \) in the objective of an additive loss \( \mathcal{L}_j \).

\[
\mathcal{L} = -\frac{1}{|D|} \sum_{X_i \in D} \beta_i (-\mathcal{L}_i) + \lambda(\beta_i - 1)^2 \tag{1}
\]

\( \beta_i \) is updated by the gradient descent \( \beta_i \leftarrow \beta_i - \frac{\partial \mathcal{L}}{\partial \beta_i} \). If \( X_i \) is hard to classify, \( -\mathcal{L}_i \) will be smaller. In order to minimize the overall loss, its \( \beta_i \) will be larger. The important samples are those difficult to learn with \( \beta > \epsilon \). We introduce a regularization term \( (\beta - 1)^2 \) and initialize \( \beta = 1 \) to penalize it when rapidly decaying toward 0.

### 2.2 Uncertainty-Based Self-Confidence

People gain self-confidence through their abilities \( (U_{\text{model}}) \) and the problem difficulty \( (U_{\text{data}}) \). We approximate the self-confidence of the model in Equation 2 by defining the total uncertainty [18] \( (U_{\text{total}} = U_{\text{model}} + U_{\text{data}}) \). A small score of total uncertainty indicates the model is confident that the current data has been well learned, and the termination of the decline in scores represents the signal to shift to the next training stage.

\[
\text{Confidence} = \frac{1}{U_{\text{total}}} \tag{2}
\]

We define the model uncertainty as the variance \( \sigma \) of a distribution of \( K \) classification probabilities for data \( X \). The classification results are predicted by the model with \( K \) different parameter disturbances. For computational efficiency, we adopt widely used Monte Carlo Dropout [8] to approximate Bayesian inference and achieve the parameter disturbances. For the current distribution \( D^m = \{X_{1:t^m}, C\} \), the model uncertainty of model \( f(\theta) \) is:

\[
U_{\text{model}}(f(\theta), D^m) = \frac{1}{|D^m|} \sum_{(x_{1:t^m}, c) \in D^m} \sum_{k=1}^K \log(p(c|x_{1:t^m}, \theta_k)) \tag{3}
\]

We define the data uncertainty function based on the entropy of the predictive distribution [13]. It behaves similar to max probability, but represents the uncertainty encapsulated in the entire distribution. We inject the learned importance coefficient \( \beta \) data uncertainty. Thus, the most uncertain data for the model is that has the largest combination value of entropy and importance. The uncertainty of a data \( x \) is:

\[
U_{\text{data}}(x) = \frac{\beta}{|C|} \sum_{c \in C} p(c|x) \log(1 - p(c|x)) \tag{4}
\]
Finally, we can get the total uncertainty for the current model:

\[
U_{total}(f^m) = U_{model}(f^m, D^m) + U_{data}(D^m)
\]

\[
= \frac{1}{|C|} \sum_{c \in C, x \in D^m} U_{model}(f^m, (x, c)) + \frac{1}{|D^m|} \sum_{x \in D^m} U_{data}(x) \tag{5}
\]

### 2.3 Confidence-Guided Training Process

C³TS consists of 3 cooperative modules: (1) Initial data arrangement module (Algorithm 1 Line 1-5) gives an initial data learning order. It imitates the fact that before starting a study, people will arrange the learning order according to the knowledge difficulty. (2) Objective-confidence scheduling module (Line 6-12) controls the overall learning process for \( M \) and \( D \). It determines the learning and relearning data by importance-based objective-confidence and solves forgetting and overfitting. It imitates the fact that people will decide what to review based on their test scores. (3) Self-confidence scheduling module (Line 9-12) controls the duration of each learning stage with a task \( M^m \). It determines the evidence of model convergence by uncertainty-based self-confidence. It imitates the fact that people decide whether they have mastered the knowledge or not through their confidence in the current knowledge.

Assuming that the computational complexity of updating a model by one sample is \( O(d) \), then training a model by \( D \) time series with length \( T \) and \( E \) epochs usually costs \( O(TEDd) \). In the training process, C³TS trains a model with \( N \) distributions \( D \) with \( N \) training tasks \( M \), assuming there are \( E \) epochs and \( S \) retained sample in each \( M^m \), the complexity will be \( O(NE^d(D + S)d) \). The overall complexity is \( O((TE + NE^d)(2D + S)d) \). As \( S \ll D \) and \( N \) is a small constant with \( N < T \), the complexity of C³TS approximates \( O(TEDd) \), almost being the same as the general training strategy.

**Algorithm 1 C³TS**

**Input:** Training set \( S = \{(X^i, C^i)\}_{i=1}^N \): An untrained target model \( f^j \); An unsupervised same model as the target model \( f^j \).

**Output:** A well-trained target model \( f^M \).

1. // Data Arrangement (Pre-train)
2. Extend \( S \leftarrow \{(X^i, C^i)\}_{i=1}^N \)
3. Train \( f^j \) by \( S \) using loss in Equation 1
4. Get \( U_{data} \) for each \( x \in S \) of \( f^j \) using Equation 4
5. Split \( S \) into \( M \) datasets \( D = \{D^m\}_{m=1}^M \) by \( U_{data} \) and define \( M \) baby tasks \( M = \{M^m\}_{m=1}^M \) by \( D \) with increasing \( U_{data} \)
6. //Objective-confidence (Task scheduling)
7. Initialize current training set buffer \( B = D^1 \)
8. for \( m = 1 \) to \( M \) do
9. // Self-confidence (Duration scheduling)
10. while not early stop of Equation 2 do
11. Train \( f^m \) by \( B \) using loss \( U_{total} \) in Equation 5
12. end while
13. Get \( \beta \) for each \( X \in B \) by \( f^m \) using Equation 1
14. Update \( B = \{D^{m+1}, \{X_i|\beta_i > \epsilon\}\} \)
15. end for

### 3 EXPERIMENTS

#### 3.1 Settings

**Baselines.** (1) ECTS-based: SR [20], ECEC [17]; (2) Replay-based: CLEAR [23], CLOPS [11]; (3) CL-based: DIF [10], UNCERT [11]; (4) Confidence-based: DROPOUT [8], TCP [6], STL [2].

**Datasets.** Regular COVID-19 mortality prediction helps for treatment and resource allocation [28] (COVID-19 [31]); Continuous sepsis diagnose helps improve patient outcomes (SEPSIS [22]). Earthquake warning helps reduce casualties [1] (UCR-EQ [5]); Rainfall warning helps prevent natural disasters [15] (USHCN [19]).

**Evaluation Metrics.** (1) The classification accuracy is evaluated by AUC-ROC; (2) The performance on forgetting and overfitting is evaluated by \( BW = \frac{1}{|M|-1} \sum_{i=1}^{|M|} R_{i,i} - \hat{R}_{i,i} \) and \( FW = \frac{1}{|M|-1} \sum_{i=1}^{|M|} |\sum_{j \neq i} R_{i,j} - \hat{R}_{i,j}| \). (3) The result uncertainty is evaluated by conditional Prediction Intervals \( PI = \{CP^{-1}(\frac{a}{2}), CP^{-1}(1 - \frac{a}{2})\} \).

#### 3.2 Results and Analysis

**Settings of hyper-parameters.** The number of distribution \( M \) and the threshold of importance coefficient \( \epsilon \) are two hyper-parameters of C³TS. For \( M \), we construct \( U_{data} \) into a normal distribution \( \mathcal{N}_\mu(\mu, \sigma^2) \), then use its confidence intervals \(-\sigma < \mu < -(\sigma + 1)\epsilon \) to split the dataset and get initial \( M \) distributions. For \( \epsilon \), we construct \( \beta \) into a positive skewed distribution \( \mathcal{N}_\mu(\mu, \sigma^2) \) with \( \mu < \sigma < m \) and make \( \epsilon = m \). The important sample is the data with \( \beta > \epsilon \).

**Accuracy for continuous classification.** C³TS is significantly better than all baselines as shown in Figure 3 (1). In Bonferroni-Dunn test, \( k=10, n=3, m=5 \) are the number of methods, datasets, cross-validation fold, then \( CD = q_{0.05} \frac{\sum_{k=1}^{k+1} m_{kk}}{m_{kk}} = 2.655, rank(CCTS) = 1 < CD \). Thus, the accuracy is significantly improved. Take sepsis diagnosis as an example, compared with the best baseline, C³TS improves the accuracy by 1.2% on average, 2.1% in the early 50% time stage when the key features are unobvious. Each hour of delayed treatment increases sepsis mortality by 4-8% [26]. With the same accuracy, C³TS can predict 0.965 hour in advance.

**Performance on forgetting and overfitting.** C³TS can alleviate these two problems with the highest \( BW \) and \( FW \) in Table 1, meaning it has the lowest negative influence that learning the new tasks has on the old tasks, especially for SEPSIS and COVID-19. Figure 3 (2) shows the case study that the partial replay of importance samples trades off forgetting and overfitting. Confidence-based methods have relatively better generalization performance.

**Analysis of learning order.** The learning order based on confidence decline makes LSTM perform best as shown in Table 2. Most existing methods train the model randomly. Some work [30] has paid attention to data learning order by using difficulty and uncertainty to measure data. Experiments show that it has greater potential to define the order by imitating human confidence.

**Analysis of replayed samples.** The important samples include not only the data hard to learn but also the representative data in each class as shown in Figure 3 (3). It might be because the representative data is similar to the most common data, resulting in a greater additive loss, therefore leading to smaller coefficients in Equation 1. Thus, we can redefine the important sample to the data that can represent a class and the data that is difficult to distinguish.

**The difference among baselines.** Baselines divide data into different distributions / baby steps. Their divisions are considerably different. As shown in Table 3, the difference in the middle step, steps 2-4, is greater than that in steps 1 and 5. It shows that the most simple and complex time series quantified by different measures.
Table 1: The Performances of Solving Catastrophic Forgetting and Over Fitting Problem (BWT↑, FWT↑).

| Dataset   | SR      | ECEC    | CLEAR   | CLOPS   | DIF     | STL     | C⁢³⁴TS  | SR      | ECEC    | CLEAR   | CLOPS   | DIF     | STL     | C⁢³⁴TS  |
|-----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| UCR-EQ    | +0.019  | +0.021  | +0.053  | +0.052  | +0.022  | +0.020  | +0.058  | +0.121  | +0.129  | +0.312  | +0.301  | +0.124  | +0.120  | +0.345  |
| USHCN     | +0.028  | +0.034  | +0.063  | +0.074  | +0.017  | +0.023  | +0.084  | +0.212  | +0.128  | +0.335  | +0.301  | +0.205  | +0.216  | +0.342  |
| COVID-19  | -0.001  | +0.010  | +0.009  | +0.014  | +0.002  | +0.006  | +0.020  | +0.126  | +0.221  | +0.427  | +0.439  | +0.220  | +0.232  | +0.455  |
| SEPSIS    | -0.019  | -0.017  | +0.030  | +0.032  | -0.010  | -0.008  | +0.035  | +0.095  | +0.165  | +0.401  | +0.397  | +0.205  | +0.216  | +0.410  |

4 CONCLUSION

We propose a new concept of CCTS to meet real needs. To achieve it, we design a novel confidence-guided approach C⁢³⁴TS. It can imitate the human behavior of object-confidence and self-confidence. We test the method on four real-world datasets based on classification accuracy, forgetting, and overfitting, showing that C⁢³⁴TS is better than all baselines. We also analyze hyper-parameters, distributions, and learning orders, showing the mechanism of C⁢³⁴TS and the matching of confidence between C⁢³⁴TS and human. We prove that the confidence-guided strategy is an effective and self-adaptive indicator to guide the learning process.

ACKNOWLEDGMENTS

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