CTW: Confident Time-Warping for Time-Series Label-Noise Learning

Peitian Ma[1], **Zhen Liu**[1], **Junhao Zheng**[1], **Linghao Wang**[1] and **Qianli Ma**[1,2*]
[1]School of Computer Science and Engineering,
South China University of Technology, Guangzhou, China
[2]Key Laboratory of Big Data and Intelligent Robot
(South China University of Technology), Ministry of Education
ma_scuter@163.com, cszhenliu@mail.scut.edu.cn, junhaozheng47@outlook.com, wlhsama@gmail.com,
qianlima@scut.edu.cn

Abstract

Noisy labels seriously degrade the generalization ability of Deep Neural Networks (DNNs) in various classification tasks. Existing studies on label-noise learning mainly focus on computer vision, while time series also suffer from the same issue. Directly applying the methods from computer vision to time series may reduce the temporal dependency due to different data characteristics. How to make use of the properties of time series to enable DNNs to learn robust representations in the presence of noisy labels has not been fully explored. To this end, this paper proposes a method that expands the distribution of **C**onfident instances by **T**ime-**W**arping (CTW) to learn robust representations of time series. Specifically, since applying the augmentation method to all data may introduce extra mislabeled data, we select confident instances to implement Time-Warping. In addition, we normalize the distribution of the training loss of each class to eliminate the model's selection preference for instances of different classes, alleviating the class imbalance caused by sample selection. Extensive experimental results show that CTW achieves state-of-the-art performance on the UCR datasets when dealing with different types of noise. Besides, the t-SNE visualization of our method verifies that augmenting confident data improves the generalization ability. Our code is available at https://github.com/qianlima-lab/CTW.

1 Introduction

The excellent performance of Deep Neural Networks (DNNs) in the classification task is primarily attributed to the collection of large datasets with high-quality annotations [Deng *et al.*, 2009]. However, the process of obtaining abundant labeled data inevitably led to the issue of noisy labels due to sensor errors and manual labeling errors, etc. [Atkinson and Metsis, 2020; Castellani *et al.*, 2021]. As a result, DNNs can easily overfit noisy instances during training, and the generalization performance is seriously impaired [Song *et al.*, 2022; Reed *et al.*, 2014]. Hence, Label-Noise Learning (LNL) has been drawing increasing attention in recent years.

Although recent advances in LNL have mitigated the noisy label problem in computer vision to some extent, how to deal with noisy labels for time-series data has rarely been investigated. In this study, we empirically find that simply applying some existing methods of LNL in the field of vision to time-series data does not lead to satisfactory performance, which necessitates the study of LNL for time-series.

More specifically, we argue that it is crucial to capture the temporal dependency of time series in label-noise learning of time series. Time-Warping is employed to address the variability of temporal locations of events in a window by simulating sampling from different temporal locations. This data augmentation method has been initially confirmed to improve the generalization performance of DNNs on time series [Um *et al.*, 2017]. However, simply applying data augmentation to all data may result in the augmentation of noisy instances (*i.e.*, mislabeled instances). As a result, extra noisy instances are introduced, and the generalization performance of DNNs is degraded. Following the recent studies in LNL, we can separate clean instances from noisy ones with a loss threshold according to the small-loss criterion [Gui *et al.*, 2021]. Specifically, DNNs tend to first learn clean instances and then noisy ones. In other words, the loss of clean instances is usually smaller than that of noisy ones. Therefore, we construct a confident set with the small-loss criterion and apply Time-Warping only on the confident set, avoiding introducing extra noisy instances. In this way, DNNs learn better representations of the clean instances filtered by the small-loss criterion.

However, when constructing the confident set, using a fixed loss threshold for all classes may lead to the class imbalance. Specifically, the loss will be smaller when the class is easier to learn. Then more instances are selected from easy classes while fewer are from hard classes [Karim *et al.*, 2022]. Consequently, DNNs suffer from class imbalance and has poor generalization performance. To address this issue, we propose to normalize the training loss of each class separately to encourage that the constructed confident set has a relatively balanced class distribution.

In summary, we proposed Confident Time-Warping (CTW), which implements Time-Warping on a confident set obtained with unbiased sample selection to help the model learn robust representations. Specifically, before employ-

*Qianli Ma is the corresponding author.

ing sample selection based on the small-loss criterion, we first normalize the training losses in each class separately with Z-score, and select confident instances according to the unbiased training loss. Then, Time-Warping is applied to the instances in the confident set to obtain confidently augmented instances. By augmenting confident samples, the model learns better feature representations and becomes more capable of distinguishing clean instances from noisy ones. When the model has a stronger ability to discriminate clean instances from noisy ones, more clean instances are selected to the confident set for data augmentation, which forms a virtuous circle. It is worth noticing that Time-Warping helps the model learn the temporal dependency of time series by simulating sampling from different temporal locations, which plays an important role in the virtuous circle. In addition, we enhance the learning process by introducing an auxiliary task to reconstruct all instances. We summarize the contributions as follows:

- We propose Confident Time-Warping (CTW), a robust method for time-series label-noise learning. The model implements Time-Warping on confident sets to learn robust representations and expand the distribution of clean data with an aim to avoid introducing extra noisy labels.
- When constructing the confident set, we propose to normalize the training loss of different classes separately, which eliminates the selection preference of the model to alleviate class imbalance caused by sample selection.
- Extensive experiments show that CTW achieves the SOTA performance on the UCR datasets. Additionally, we demonstrate experimentally that Time-Warping is not simply used to enlarge the training set, but to facilitate the model to learn the distribution of clean data. We also discuss the advantages of Time-Warping over other augmentation methods.

2 Related Work

2.1 Learning with Noisy Labels

Current label-noise learning techniques are developing rapidly. Among them, loss adjustment and sample selection are two popular methods. The former enables the model to adjust the loss with a transition matrix estimated adaptively by oneself. For example, Xia et al. [Xia *et al.*, 2019] propose a method to estimate the transition probability matrix without anchor points for loss correction. Yao et al. [Yao *et al.*, 2020] factorize the matrix into the product of two easy-to-estimate matrices to avoid directly estimating the noisy class posterior, which may acquire a more accurate transition matrix. These methods often need to prevent the accumulation of errors caused by incorrect correction.

DNNs tend to learn simple patterns first, and then gradually overfit to noisy patterns, *i.e.*, the memory effect [Arpit *et al.*, 2017; Song *et al.*, 2019]. Therefore, small-loss training samples are usually adopted as clean samples to design robust training methods. Among them, Co-teaching and Co-teaching+ [Han *et al.*, 2018; Yu *et al.*, 2019] train two networks simultaneously and update each network on the data selected by the other. Gui et al. [Gui *et al.*, 2021] provide a theoretical basis for the small-loss criterion and propose an effective method that selects instances with a small mean loss class by class. These works assume that the noise rate is possible, yet it's difficult to achieve it in the real world. Some methods filter data through cross-validation [Chen *et al.*, 2019], or auxiliary networks with the loss moving average as a dynamic threshold [Jiang *et al.*, 2018]. JoCoR [Wei *et al.*, 2020] calculates a joint loss with co-regularization for each instance to select clean ones. CORES [Cheng *et al.*, 2020] compares the samples' regularized losses with a dynamic threshold. For the class imbalance caused by sample selection, Karim et al. [Karim *et al.*, 2022] choose to select the same rate of samples for each class to avoid the model preferring to choose samples of easy-to-learn classes. Additionally, Huang et al. [Huang *et al.*, 2022] employ a Gaussian mixture model to fit the training loss class by class, which alleviates the class imbalance. In this paper, we propose a simple and effective method that normalizes the training loss of each class to eliminate the preference of selection.

For time-series label-noise learning, Atkinson et al. [Atkinson and Metsis, 2020] adapt Labelfix, an existing tool of the vision field, to process time-series data. SREA [Castellani *et al.*, 2021] aims to gradually correct the mislabeled samples in a self-supervised fashion for time-series label-noise learning. However, these methods do not take into account the properties of time series.

2.2 Augmentation for Generalization

To improve the generalization for LNL, data augmentation has been successfully combined with semi-supervised learning techniques to achieve consistency regularization [Hu *et al.*, 2021]. DivideMix [Li *et al.*, 2020] uses Gaussian mixture models to divide data into clean and noisy sets, in which noisy ones are viewed as unlabeled samples. Then it applies a semi-supervised technique MixMatch [Berthelot *et al.*, 2019]. Unlike consistency regularization training, Nishi et al. [Nishi *et al.*, 2021] propose a weak augmentation for every loss modeling and pseudo-labeling task, and a strong augmentation to allow the back-propagation step to improve generalization.

However, the above methods may reduce the temporal dependency of time series. In the field of time series, various data augmentation techniques are proposed to improve the generalization of DNNs [Le Guennec *et al.*, 2016; Um *et al.*, 2017; **?**], like some simple transformations to obtain augmented sets: adding noise, crop, drift, and so on. Among them, Time-Warping is an effective augmentation method [Um *et al.*, 2017] which simulates sampling from different temporal locations. But augmenting all data may introduce extra noisy instances. To deal with the issue, we propose a training strategy that applies Time-Warping on confident samples. In this way, it can facilitate the learning of clean time series, avoiding learning the wrong distribution of time series due to the interference of noisy data.

3 Proposed Method

In this section, we introduce the proposed CTW, as shown in Figure 1. The original time series is input to train along the solid arrow, and the training losses are calculated after the

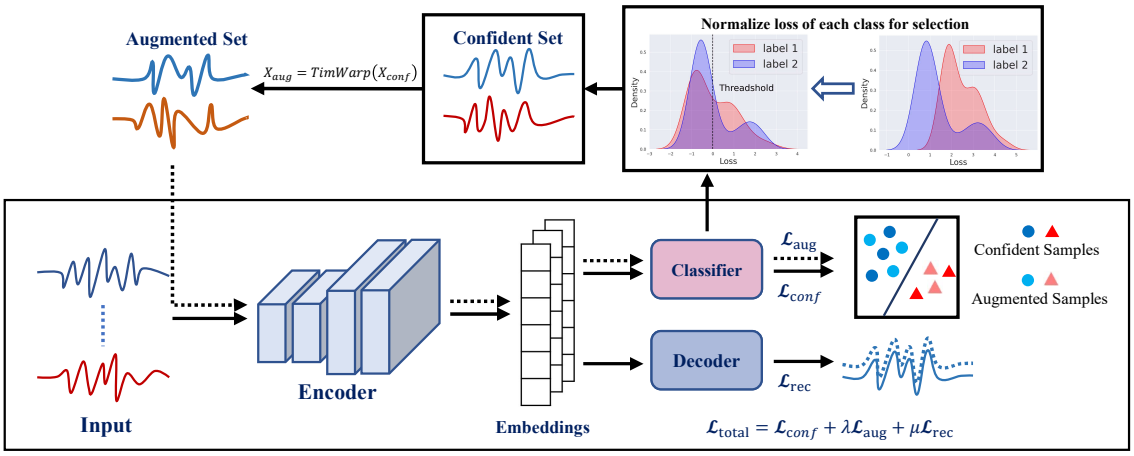


Figure 1: The illustration of the proposed CTW. The dashed arrows are the training routes of the augmented samples; the solid arrows are the training routes of the original samples. Before selecting confident samples, we align the training loss distributions of instances of each class to avoid selection preference. Augmented instances obtained by Time-Warping on the confident set help the model learn more robust representations.

classifier. The confident set is selected after eliminating the preference of selection. Confident instances augmented by time warping are trained with dashed arrows, promoting the representation learning for clean time series. Furthermore, the model further learns robust representations through reconstruction. Finally, the classification will be able to form robust decision boundaries.

3.1 Preliminary

Considering a c-class classification problem, let $\mathcal{X}$ be the input space, $\mathcal{Y} = \{0,1\}^c$ be the label space. We provide a training dataset $D = \{(x_i, y_i)\}_{i=1}^N$ drawn from distribution $\mathcal{D}_{X,Y} \subset \mathcal{X} \times \mathcal{Y}$, in which each (x_i, y_i) is independent and identically distributed. However, in the process of collecting samples, the sample may be wrongly labeled. In this paper, let $\tilde{y}_i$ be the observed label, y_i be the ground-truth label and the actual dataset we have be $\tilde{D} = \{(x_i, \tilde{y}_i)\}_{i=1}^N$. The goal of the task is to find a mapping function $f(\cdot;\Theta) : \mathcal{X} \to \{0,1\}^c$ of the DNN parameterized with Θ, such that $f(x_i;\Theta) = y_i$ as much as possible. For convenience, $y_i = k$ is used to indicate that the i-th instance belongs to the k-th class.

3.2 Eliminate the Preference of Selection for Different Classes

As DNNs tend to learn simple and generalized patterns first and then gradually overfit to noisy ones [Arpit *et al.*, 2017; Song *et al.*, 2019], sample selection methods based on small-loss criteria are commonly used. However, the distribution of training losses for different classes may be inconsistent during training. For example, when the loss of class A is smaller, and the loss of class B is generally larger, instances of class A are favored as the confident ones due to their small loss. Then it leads to the class imbalance in the confident set (A practical example is given in Section H of the supplementary material.). The model will tend to learn the pattern of class A. With this preference, cognitive biases will accumulate and then degrade the generalization performance of the model.

To eliminate the preference of selection mentioned above, we propose to normalize training losses of instances class by class by Eq. (2). Specifically, we first calculate the weighted mean loss as the selection criterion as follows:

$$\tilde{\ell}_i := \gamma \ell_i^{(t)} \times (1-\gamma) \sum_{j=t-\beta-1}^{t-1} \ell_i^{(j)}, \tag{1}$$

where γ and β are hyperparameters, $\ell_i^{(t)}$ is the training loss of the i-th instance in the t-th epoch. In Eq. (1), γ controls the weight of the current loss, β controls the historical epochs. It takes into account the historical losses of instances, since the losses of instances may fluctuate in different epochs as DNNs are optimized by stochastic gradient descent [Gui *et al.*, 2021]. This can further increase the probability of locating clean ones. Then, we normalize training losses in each class separately as follows:

$$\forall k \in [c],\ Normalize(\{\tilde{\ell}_i \mid \tilde{y}_i = k, \forall i \in [B]\}), \tag{2}$$

where $Normalize(\cdot)$ indicates normalizing the training loss with z-score class by class, B is the batch size.

Then the distribution of loss from different classes has the same mean (zero) and variance (one), indicating no selection preference between different classes. After that, we look for confident samples with small-loss criteria:

$$\mathcal{D}_{conf} = \{(x_i, \tilde{y}_i) \mid \bar{\ell}_i \leq \ell_{thred}, \forall i \in [B]\}, \tag{3}$$

where l_{thred} is the dynamic threshold for selection. We take the average loss as the dynamic threshold, and all samples with a loss below the threshold are viewed as confident instances. Given the confident set, classification learning is

conducted as

$$\mathcal{L}_{\text{conf}} = \sum_{(\boldsymbol{x}_i, \tilde{y}_i) \in \mathcal{D}_{conf}} \ell\left(\boldsymbol{f}\left(\boldsymbol{x}_i\right), \tilde{y}_i\right), \tag{4}$$

where $\ell(\cdot, \cdot)$ is the cross entropy loss function.

3.3 Time-Warping on Confident Set

Data augmentation methods are proposed to improve the generalization of the model. Among them, Time-Warping (TW) is an excellent way to perturb the temporal location [Um *et al.*, 2017]. It distorts the time intervals between samples by linearly interpolating the original sequence and re-selecting the sample points of the series. As a consequence, the temporal locations of the samples are changed. It is employed to address the variability of temporal locations of events. We show that the backbone we adopted with Time-Warping performs better than Vanilla does on 82 clean UCR datasets in Section G of the supplementary material.

For LNL, existing research usually uses data augmentation as a method to achieve consistency regularization [Li *et al.*, 2020]. However, applying the augmentation method to all data may introduce extra noisy labels, which affects the model's learning of the pattern of clean time series. In this work, we emphasize that applying Time-Warping to confident samples rather than all samples can enhance the model's learning of the distribution of clean ones, thereby greatly improving the generalization of the model. Specifically, given the confident set $\mathcal{D}_{conf}$, we let:

$$\mathcal{D}_{aug} = \{(TimeWarp(x_i), \tilde{y}_i) \,|\forall (x_i, \tilde{y}_i) \in \mathcal{D}_{conf}\}, \tag{5}$$

Then we start over to train the augmented set ($\mathcal{D}_{aug}$) along the encoder and the classifier, see Figure 1. The added loss with augmentation is calculated:

$$\mathcal{L}_{aug} = \sum_{\left(\boldsymbol{x}_i^{aug}, \tilde{y}_i\right) \in \mathcal{D}_{aug}} \ell\left(\boldsymbol{f}\left(\boldsymbol{x}_i^{aug}\right), \tilde{y}_i\right). \tag{6}$$

Note that each augmented sample keeps the original label, which is different from computing the loss of consistency regularization.

In addition, we introduce an auxiliary task to reconstruct all instances, which allows the model to learn more robust representations of time series as well as avoiding wasting the information of noisy samples. We do not reconstruct augmented samples. It will be discussed further in the subsequent experimental section. The reconstruction loss ($\mathcal{L}_{rec}$) could be computed by Mean Squared Error (MSE).

In fact, to further improve the generalization performance of the model, semi-supervised learning can be also combined to exploit noisy samples. Although it will not be discussed in this paper, we demonstrate in experiments that CTW is sufficient to achieve state-of-the-art performance for LNL on time series datasets. Lastly, according to the above analyses, the parameters of the entire model are updated by

$$\mathcal{L}_{total} = \mathcal{L}_{conf} + \lambda \mathcal{L}_{aug} + \mu \mathcal{L}_{rec}, \tag{7}$$

where λ and μ are loss weights, which we set $\lambda = 1$ and $\mu = 1$ in following experiments unless otherwise specified.

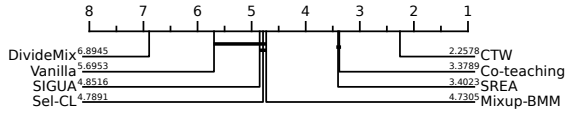


Figure 2: Critical difference diagram of the comparison with baselines on 128 UCR datasets with 30% symmetric noise

.

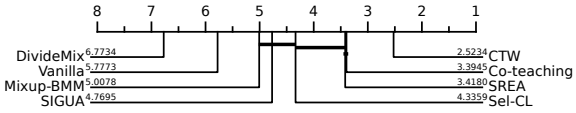


Figure 3: Critical difference diagram of the comparison with baselines on 128 UCR datasets with 40% asymmetric noise

.

Methods	Sym 30%		Asym 40%	
	Avw_F1	#Best	Avw_F1	#Best
Vanilla	0.608	2/128	0.499	4/128
SIGUA	0.652	4/128	0.544	3/128
Co-teaching	0.696	12/128	0.590	15/128
Mixup-BMM	0.630	10/128	0.530	6/128
DivideMix	0.449	2/128	0.399	7/128
Sel-CL	0.601	19/128	0.542	22/128
SREA	<u>0.700</u>	<u>27/128</u>	<u>0.589</u>	<u>26/128</u>
CTW	**0.733**	**55/128**	**0.621**	**45/128**

Table 1: Comparison with baseline methods on 128 UCR datasets. *Avw_F1*: the average of weighted F1-score (the average of standard deviation). *#Best*: the number of best results. The best results are in **bold**. The second largest results are <u>underlined</u>.

It's worth noticing that in our framework, the model learns better feature representations and becomes more capable of distinguishing clean instances from noisy ones by Time-Warping on confident sets. When the model has a stronger ability to discriminate clean time series from noisy ones, more clean instances are selected to the confident set for data augmentation, which forms a virtuous circle.

4 Experiments

4.1 Experiments Setup

In this section, we describe our experiments. Full experimental results are shown in the supplementary material.

Datasets. We evaluate our model on publicly available time-series classification datasets from the UCR and UEA repositories [Dau *et al.*, 2019; Bagnall *et al.*, 2018]. Among them, 13 datasets (8 from UCR and 5 from UEA) serve as our benchmarks. The information about them can be found in Section A of the supplementary material. In addition, to make the case more convincing, we evaluate the methods on all 128 datasets in the UCR repository with 30% symmetric noise and 40% asymmetric noise, respectively. Note the noise rate as η. For symmetric noise (Sym), the probability of each sample in the dataset being mislabeled as another class is $\frac{\eta}{c-1}$; the asymmetric noise (Asym) considered in this paper

Methods	Sym				Asym				IDN	
	15%	30%	45%	60%	10%	20%	30%	40%	30%	40%
Vanilla	0.768	0.667	0.543	0.398	0.793	0.730	0.651	0.547	0.646	0.550
SIGUA	0.777	0.709	0.589	0.444	0.795	0.751	0.691	0.584	0.657	0.597
Co-teaching	0.809	0.749	0.673	0.509	0.814	0.779	0.738	0.635	0.722	0.653
Mixup-BMM	0.762	0.718	0.616	0.494	0.761	0.743	0.706	0.611	0.681	0.611
Dividemix	0.413	0.420	0.412	0.345	0.440	0.414	0.430	0.422	0.423	0.378
Sel-CL	0.708	0.700	0.645	**0.580**	0.728	0.705	0.687	0.623	0.685	0.659
SREA	0.802	0.747	0.638	0.495	0.803	0.764	0.712	0.610	0.708	0.647
CTW	**0.827**	**0.786**	**0.690**	0.522	**0.836**	**0.819**	**0.771**	**0.692**	**0.758**	**0.677**

Table 2: Comparison with baseline methods in the average of weighted F1-score on benchmark datasets. The best results are in **bold**. The second largest results are underlined. More results are shown in Section B of the supplementary material.

Methods	TimeWarp	Gaussion Noise	Convolve	Drift	Oversample	Crop	MF_Mixup
AAug	0.784	0.748	0.752	**0.754**	0.743	**0.772**	0.753
CAug	**0.792**	**0.761**	**0.762**	0.750	**0.758**	**0.772**	**0.764**

Table 3: Comparison of *AAug* and *CAug*. Each model equips with different augmentation methods, dealing with 30% symmetric noise. *AAug*: Augment all instances. *CAug*: Augment confident instances. The average of weighted F1-score on benchmark datasets are reported. The results show that augmenting confident instances improves the generalization performance in general.

is paired noise: class A $\rightarrow$ class B, class B $\rightarrow$ class C, class C $\rightarrow$ class A, where the probability of label flipping to incorrect ones is η. For instance-dependent noise (IDN), we corrupt labels as [Xia *et al.*, 2020] do.

Architecture. We utilize FCN as the backbone in our model following SREA [Castellani *et al.*, 2021]. The encoder in our model is composed of 4 convolutional blocks. Each block is connected with a 1D-convolution layer. Batch normalization [Ioffe and Szegedy, 2015], ReLU activation, and a dropout layer are applied following the 1D-convolution layer. The decoder has a symmetric structure of the encoder. Then a linear classifier follows the encoder. The dimension of embeddings after the encoder is 32. The linear classifier has 128 hidden units.

Baselines. For a fair comparison, all experiments use the same structure mentioned above. All comparative methods are: **Vanilla** (A method that does not adopt any technology of label-noise learning.), **Co-teaching** [Han *et al.*, 2018], **Mixup-BMM** [Arazo *et al.*, 2019], **SIGUA** [Han *et al.*, 2020], **DivideMix** [Li *et al.*, 2020], **SREA** [Castellani *et al.*, 2021] and **Sel-CL** [Li *et al.*, 2022]. For Co-teaching and SIGUA, the noise rate is provided as needed. Details about baselines are described in Section A of the supplementary material.

Implementation Details. We use the Adam optimizer [Kingma and Ba, 2014] with an initial learning rate of 0.001. We merge the original training and test sets for all time series datasets, then perform five-fold cross-validation, training on four folds and testing on the remaining fold. We evaluate the model at the last epoch following [Han *et al.*, 2018] and [Castellani *et al.*, 2021]. The average of the weighted F1-scores (Avw_F1) is reported. In our model, unless otherwise specified, the corresponding hyperparameters default to: $\lambda = 1, \mu = 1, \gamma = 0.3$ and $\beta = 10$. For all experiments, the max epoch is set to 300. Other details are shown in Section A of the supplementary material.

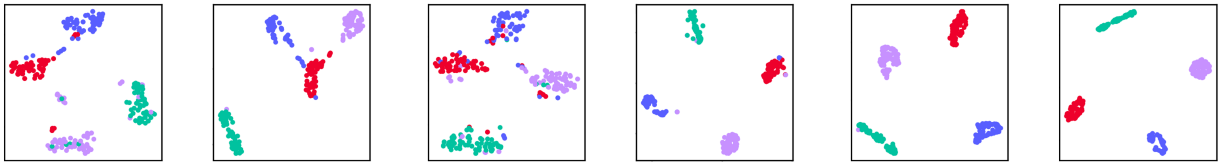
4.2 Comparison with State-of-the-Arts

As shown in Tabel 2, CTW achieves the best results on benchmarks. Except for the case of 60% symmetric noise, our model outperforms SOTA by 0.015 to 0.057 with other noise rates. Additionally, we note that the average rank (shown in Section B of the supplementary material) of our model is lower than Co-teaching in the case of 45% symmetric noise, despite our model's superiority in Avw_F1. By comparing specific results, we found that CTW performed slightly worse than Co-teaching did on 5 datasets from the UEA repository. Nonetheless, our method outperforms others in most cases according to Av_rank. Besides, an interesting fact is that DivideMix does not perform well, which is quite different from the situation in computer vision. We discuss why it fails on time series in Section B of the supplementary material.

With 30% symmetric and 40% asymmetric noise, our model achieves the best results on 55 datasets and 45 datasets among all 128 UCR datasets, which are shown in Tabel 1. To further analyze the performance, we also conduct the Nemenyi non-parametric statistical test [Demšar, 2006] and plot the critical difference diagram in Figure 2 and Figure 3. Finally, we compare all methods on all 128 UCR datasets in an average rank. The Nemenyi test shows that our model is significantly superior to Co-teaching and SREA at $p < 0.05$ level with both two kinds of noise rates.

4.3 Augmentation Works for Expanding the Distribution of Confident Instances

In this section, we demonstrate experimentally that data augmentation methods improve the generalization performance of the model by expanding the distribution of confident instances rather than simply increasing the amount of data. To get more general conclusions, we replace Time-Warping with different augmentation methods in CTW. Therefore, AAug



(a) Oversample: AAug vs CAug (b) MF_Mixup: AAug vs CAug (c) TimeWarp: AAug vs CAug

Figure 4: The t-SNE visualizations on the Trace dataset. *AAug*: Augment all instances. *CAug*: Augment confident instances. Oversample: Augmented instances are the same as original ones; MF_Mixup: Augment instances with manifold mixup; TimeWarp: Augment instances with Time-Warping. Augmenting on confident instances helps the model learn more compact representations from each class.

(Augment on all data) and CAug (Augment on confident set) are compared in this section. In detail, the augmented data in AAug will go through the sample selection stage to select confident augmented samples for updating model parameters. It is easy to find that the role of data augmentation here is merely to enlarge the training dataset. On the other hand, CAug implements the augmentation method on the confident set based on the idea of expanding the distribution of confident samples.

In the above models, we apply multiple augmentation methods for more general results: Time-Warping (TimeWarp), Add Gaussian noise (GaussNoise), Convolve, Drift, Oversample, Crop, and Manifold Mixup [Verma *et al.*, 2019]. As shown in Table 3, CAug outperforms AAug with most data augmentation methods. Among them, CAug equipped with Time-Warping achieves the best result. It is worth noting that the results of CAug equipped with Oversample show that even just repeating the training of the confident samples can significantly improve the generalization performance of the model. It can be found from t-SNE (Figure 4) that CAug with Oversample can gradually expand the proportion of clean data, while AAug with Oversample only expands the scale of training data. Although the latter also increases confident samples, it introduces mislabeled samples that are proportional to the noise rate. It further illustrates the importance of implementing data augmentation on confident samples. The same phenomenon exists in other methods. In addition, the representation learned by CAug with Time-Warping is more compact than the model equipped with other augmentation methods, which is attributed to the fact that Time-Warping helps the model learn the temporal dependency of time series by simulating sampling at different time points.

4.4 Sample Selection Analysis

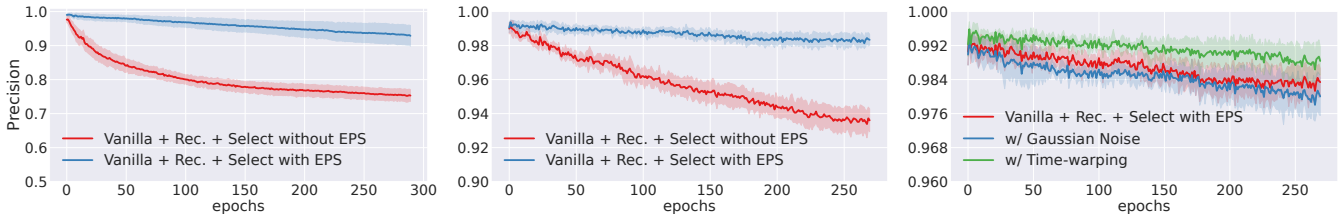
In this section, we first discuss the effects of eliminating the model's preference of selection for instances of different classes. We compare the performance of two models: one is Vanilla equipped with the common sample selection technique (selecting confident instances directly from the whole batch); the other is Vanilla with the technique of eliminating selection preference mentioned in this paper before selecting confident instances. For a fair comparison, the dynamic thresholds used for selection both take the average training loss. On the CBF dataset and the Symbols dataset (see Figure 5a and Figure 5b), the model with eliminating the preference of selection consistently has higher precision.

Methods	sym 60%	asym 40%
Vanilla + Rec. + Sel. w/o EPS	0.448	0.598
Vanilla + Rec. + Sel. w/ EPS	**0.450**	**0.613**
CTW	0.522	**0.692**
CTW (Sel. with GMM class by class)	**0.523**	0.671
CTW (Sel. with R class by class)	0.518	0.673

Table 4: Sample selection analysis. *Rec.*: Samples reconstruction. *Sel.*: Sample selection. *EPS*: Eliminating the Preference of Selection. *R*: The proportion of all losses that are less than ℓ_{thred}. The same proportion *R* is used for each class. We can see that selecting confident instances with EPS improves the generalization performance of the model. Furthermore, Time-Warping on confident instances helps the model learn more clean time series.

However, the precision curve in Figure 5 shows a downward trend as the model is trained, which may be due to the fact that the model has fully memorized some mislabeled instances at a later stage of training. It leads to a gradual accumulation of cognitive biases in the model, which makes it continuously categorize other noisy instances with similar characteristics as clean ones. The operation of eliminating selection preferences can alleviate the accumulation of this cognitive bias (see Figure 5b). Furthermore, utilizing Time-Warping on the confident set can improve the precision and also significantly alleviate the bias since the method helps the model to learn the pattern of clean time series (see Figure 5c). In contrast, adding Gaussian noise to the confident instances on the Symbols dataset cannot help the model select clean instances properly since it may damage the structure of the time series.

In addition, Huang et al. [Huang *et al.*, 2022] employs the Gaussian mixture model to select confident instances class by class and Karim et al. [Karim *et al.*, 2022] select the same proportion of instances for each class separately, which can also alleviate the class imbalance caused by sample selection. We compare their sample selection strategies with our method on 13 benchmark datasets, all methods are equipped with Time-Warping on the confident set. As shown in Table 4, CTW is slightly lower than the former (CTW Sel. with GMM class by class) with 60% symmetric noise and higher than that with 40% asymmetric noise. Nevertheless, the experimental



(a) CBF dataset (b) Symbols dataset (c) Symbols dataset

Figure 5: Sample selection analysis with precision curves. Experiments are conducted with 30% symmetric noise. The precision is calculated by $\frac{TP}{TP+FP}$, where $TP(FP)$ means the number of clean(noisy) instances in the confident set we selected. (a) and (b) show that EPS helps the model select more clean instances. (c) shows that Time-Warping on confident ones promotes the selection ability, forming a virtuous circle.

Methods	Avw_F1	Av_Rank
CTW	**0.786**	**2.00**
w/o Selection	0.727	4.54
w/o EPS	0.781	2.85
w/o Time-Warping	0.761	2.92
w/o Decoder	0.780	2.69

Table 5: Ablation experiments on benchmark datasets, dealing with 30% symmetric noise. *EPS*: Eliminating the Preference of Selection. The best results are in **bold**.

Methods	w/ Aug. Rec.	w/o Aug. Rec.	p-value
TimeWarp	0.784	**0.792**	0.280
GaussNoise	0.755	**0.761**	0.061
Convolve	0.759	**0.762**	0.319
Drift	**0.759**	0.750	0.596
Oversample	**0.761**	0.758	0.124
Crop	0.768	**0.772**	0.782
MF_Mixup	**0.772**	0.764	0.326

Table 6: Results of ablation analysis on reconstructing augmented instances, dealing with 30% symmetric noise. *Aug. Rec.*: Reconstructing augmented instances. The best results are in **bold**.

results of them are not significantly different. The results of the strategy that selects the same proportion class by class are worse than ours when dealing with both two kinds of noises.

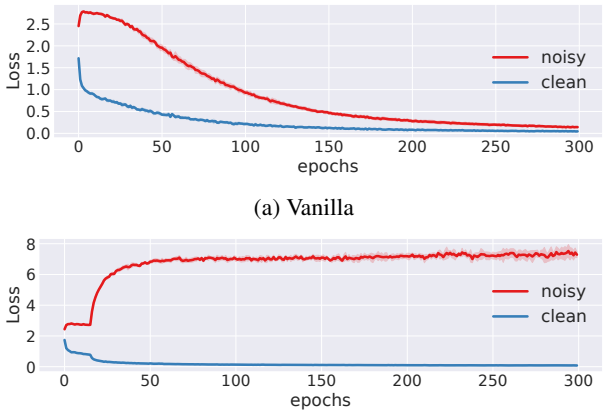
4.5 Ablation Analysis

To study the effectiveness of each component in our model, we perform ablation experiments on benchmark datasets with 30% symmetric noise as shown in Table 5. The model without the selection part gets the worst result as Time-Warping is implemented on all data.

As Table 6 shown, reconstructing the augmented samples is beneficial for some augmentation methods. But it can be found from the p-value (paired t-test) that there is no significant difference whether there is reconstruction or not. Therefore, we recommend not reconstructing the augmented samples, which reduces the computational cost.

4.6 Loss Analysis

Figure 6 shows the difference of training loss between noisy and clean samples in different models. Vanilla (Figure 6a)



(a) Vanilla

(b) CTW

Figure 6: Loss analysis on MelbournePedestrian with 40% symmetric noise. The red line shows the average loss of noisy samples. The blue line shows the average loss of clean ones.

learns clean patterns first (there is a large gap between two kinds of losses.), then fits noisy patterns gradually (the gap shrinks.). However, CTW (Figure 6b) can expand the gap of losses in training so that the small-loss criterion remains valid.

5 Conclusion

In this paper, we propose a new method focusing on time series label-noise learning called CTW, which can learn more about the distribution of confident instances and improve the generalization performance of the model. Instead of simply using data augmentation methods on all data, CTW applies Time-Warping to the confident set to help the model learn the pattern of clean instances. Additionally, we eliminate selection preference for instances of different classes, which further improves the reliability of the confident set. In order to get a more general conclusion, we replace Time-Warping in our framework with different augmentation methods. Extensive experiments show that our method can effectively promote the learning of clean instances instead of simply enlarging the scale of the training set. Finally, we demonstrate the state-of-the-art performance of CTW with extensive experiments on multiple noisy datasets.

Acknowledgments

We thank the anonymous reviewers for their helpful feedback. The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant Nos. 62272173, 61872148), the Natural Science Foundation of Guangdong Province (Grant Nos. 2022A1515010179, 2019A1515010768).

References

[Arazo *et al.*, 2019] Eric Arazo, Diego Ortego, Paul Albert, Noel O'Connor, and Kevin McGuinness. Unsupervised label noise modeling and loss correction. In *International conference on machine learning*, pages 312–321. PMLR, 2019.

[Arpit *et al.*, 2017] Devansh Arpit, Stanisław Jastrzebski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, et al. A closer look at memorization in deep networks. In *International conference on machine learning*, pages 233–242. PMLR, 2017.

[Atkinson and Metsis, 2020] Gentry Atkinson and Vangelis Metsis. Identifying label noise in time-series datasets. In *Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pages 238–243, 2020.

[Bagnall *et al.*, 2018] Anthony Bagnall, Hoang Anh Dau, Jason Lines, Michael Flynn, James Large, Aaron Bostrom, Paul Southam, and Eamonn Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[Berthelot *et al.*, 2019] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019.

[Castellani *et al.*, 2021] Andrea Castellani, Sebastian Schmitt, and Barbara Hammer. Estimating the electrical power output of industrial devices with end-to-end time-series classification in the presence of label noise. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 469–484. Springer, 2021.

[Chen *et al.*, 2019] Pengfei Chen, Ben Ben Liao, Guangyong Chen, and Shengyu Zhang. Understanding and utilizing deep neural networks trained with noisy labels. In *International Conference on Machine Learning*, pages 1062–1070. PMLR, 2019.

[Cheng *et al.*, 2020] Hao Cheng, Zhaowei Zhu, Xingyu Li, Yifei Gong, Xing Sun, and Yang Liu. Learning with instance-dependent label noise: A sample sieve approach. *arXiv preprint arXiv:2010.02347*, 2020.

[Dau *et al.*, 2019] Hoang Anh Dau, Anthony Bagnall, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, and Eamonn Keogh. The ucr time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[Demšar, 2006] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, 7:1–30, 2006.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Gui *et al.*, 2021] Xian-Jin Gui, Wei Wang, and Zhang-Hao Tian. Towards understanding deep learning from noisy labels with small-loss criterion. *arXiv preprint arXiv:2106.09291*, 2021.

[Han *et al.*, 2018] Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. *Advances in neural information processing systems*, 31, 2018.

[Han *et al.*, 2020] Bo Han, Gang Niu, Xingrui Yu, Quanming Yao, Miao Xu, Ivor Tsang, and Masashi Sugiyama. Sigua: Forgetting may make learning with noisy labels more robust. In *International Conference on Machine Learning*, pages 4006–4016. PMLR, 2020.

[Hu *et al.*, 2021] Zijian Hu, Zhengyu Yang, Xuefeng Hu, and Ram Nevatia. Simple: similar pseudo label exploitation for semi-supervised classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15099–15108, 2021.

[Huang *et al.*, 2022] Yingsong Huang, Bing Bai, Shengwei Zhao, Kun Bai, and Fei Wang. Uncertainty-aware learning against label noise on imbalanced datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 6960–6969, 2022.

[Ioffe and Szegedy, 2015] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[Jiang *et al.*, 2018] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International conference on machine learning*, pages 2304–2313. PMLR, 2018.

[Karim *et al.*, 2022] Nazmul Karim, Mamshad Nayeem Rizve, Nazanin Rahnavard, Ajmal Mian, and Mubarak Shah. Unicon: Combating label noise through uniform selection and contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9676–9686, 2022.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Le Guennec *et al.*, 2016] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. Data augmentation for

time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data*, 2016.

[Li *et al.*, 2020] Junnan Li, Richard Socher, and Steven CH Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. *arXiv preprint arXiv:2002.07394*, 2020.

[Li *et al.*, 2022] Shikun Li, Xiaobo Xia, Shiming Ge, and Tongliang Liu. Selective-supervised contrastive learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 316–325, 2022.

[Nishi *et al.*, 2021] Kento Nishi, Yi Ding, Alex Rich, and Tobias Hollerer. Augmentation strategies for learning with noisy labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8022–8031, 2021.

[Reed *et al.*, 2014] Scott Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. Training deep neural networks on noisy labels with bootstrapping. *arXiv preprint arXiv:1412.6596*, 2014.

[Song *et al.*, 2019] Hwanjun Song, Minseok Kim, Dongmin Park, and Jae-Gil Lee. How does early stopping help generalization against label noise? *arXiv preprint arXiv:1911.08059*, 2019.

[Song *et al.*, 2022] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Um *et al.*, 2017] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*, pages 216–220, 2017.

[Verma *et al.*, 2019] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

[Wei *et al.*, 2020] Hongxin Wei, Lei Feng, Xiangyu Chen, and Bo An. Combating noisy labels by agreement: A joint training method with co-regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13726–13735, 2020.

[Xia *et al.*, 2019] Xiaobo Xia, Tongliang Liu, Nannan Wang, Bo Han, Chen Gong, Gang Niu, and Masashi Sugiyama. Are anchor points really indispensable in label-noise learning? *Advances in Neural Information Processing Systems*, 32, 2019.

[Xia *et al.*, 2020] Xiaobo Xia, Tongliang Liu, Bo Han, Nannan Wang, Mingming Gong, Haifeng Liu, Gang Niu, Dacheng Tao, and Masashi Sugiyama. Part-dependent label noise: Towards instance-dependent label noise. *Advances in Neural Information Processing Systems*, 33:7597–7610, 2020.

[Yao *et al.*, 2020] Yu Yao, Tongliang Liu, Bo Han, Mingming Gong, Jiankang Deng, Gang Niu, and Masashi Sugiyama. Dual t: Reducing estimation error for transition matrix in label-noise learning. *Advances in neural information processing systems*, 33:7260–7271, 2020.

[Yu *et al.*, 2019] Xingrui Yu, Bo Han, Jiangchao Yao, Gang Niu, Ivor Tsang, and Masashi Sugiyama. How does disagreement help generalization against label corruption? In *International Conference on Machine Learning*, pages 7164–7173. PMLR, 2019.