Time Series Data Augmentation for Neural Networks by Time Warping with a Discriminative Teacher

Brian Kenji Iwana
Department of Advanced Information Technology
Kyushu Univeristy, Fukuoka, Japan
Email: brian@human.ait.kyushu-u.ac.jp

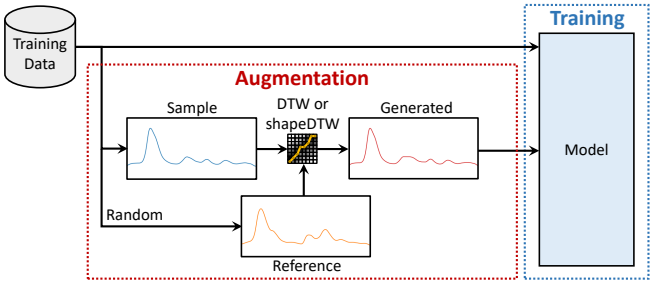
Seiichi Uchida
Department of Advanced Information Technology
Kyushu Univeristy, Fukuoka, Japan
Email: uchida@ait.kyushu-u.ac.jp

***Abstract*—Neural networks have become a powerful tool in pattern recognition and part of their success is due to generalization from using large datasets. However, unlike other domains, time series classification datasets are often small. In order to address this problem, we propose a novel time series data augmentation called guided warping. While many data augmentation methods are based on random transformations, guided warping exploits the element alignment properties of Dynamic Time Warping (DTW) and shapeDTW, a high-level DTW method based on shape descriptors, to deterministically warp sample patterns. In this way, the time series are mixed by warping the features of a sample pattern to match the time steps of a reference pattern. Furthermore, we introduce a discriminative teacher in order to serve as a directed reference for the guided warping. We evaluate the method on all 85 datasets in the 2015 UCR Time Series Archive with a deep convolutional neural network (CNN) and a recurrent neural network (RNN). The code with an easy to use implementation can be found at https://github.com/uchidalab/time_series_augmentation.**

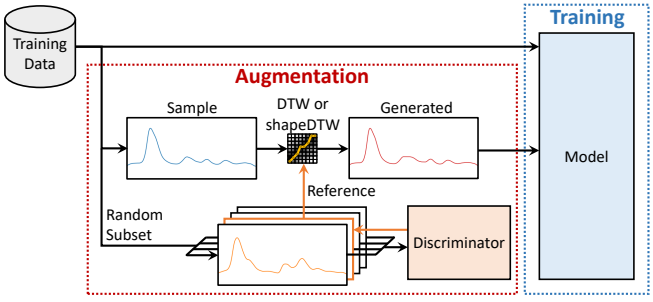
I. Introduction

In recent times, deep neural networks have become commonplace and continue to set the state-of-the-art benchmarks across many domains, such as natural scene object classification [1], machine translation [2], graph classification [3], and more. Part of the recent successes is due to the increase in data availability and the advancement of hardware to support it [4]. In fact, it is well-known that increasing the amount of data helps with generalization and, in turn, the accuracy of many machine learning models [5]–[7].

However, unlike the image domain, time series datasets tend to be tiny in comparison. For example, one of the most used sources of time series classification datasets, the University of California Riverside (UCR) Time Series Archive [8], contains 85 time series datasets but only 10 have more than 1,000 training samples and the largest, ElectricDevices, only has 8,926. By comparison, the popular image datasets, ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9], MNIST [10], and CIFAR [11], have 1.2 million, 60,000, and 50,000 training patterns respectively. Thus, in order to use the full potential of modern machine learning methods, there is a need for time series classification data.



(a) Proposed Random Guided Warping



(b) Proposed Discriminative Guided Warping

Fig. 1. Workflow of the proposed method. For data augmentation, in (a), two patterns are selected and DTW is used to warp a sample by the reference. In (b), the sample is warped by a reference selected by a discriminator using a small subset of samples. The generated patterns are used together with the original training data to train a model.

One solution to tackle this problem is the use of data augmentation. Specifically, data augmentation is a common data-space solution to increase the generalization ability of machine learning models. It does this by increasing the size of the training dataset using synthetic patterns. Data augmentation has shown to be an effective model-independent method of reducing overfitting and expanding the decision boundary modeled by the data [12].

Data augmentation for images is a well-explored field, especially in combination with neural networks. It has almost become a standard practice for image classification. For example, many of the popular Convolutional Neural Networks (CNN) based architectures used a form of data augmentation in their training, such as the original proposals of

AlexNet [11], Very Deep Convolutional Networks (VGG) [13], and Residual Networks (ResNet) [14]. Comparatively, there are fewer established time series data augmentation methods and fewer standard data augmentation practices in time series classification [15]. Most of the methods that exist are just time series adaptations inspired by image recognition. These methods generally rely on simple transformations, such as jittering (adding noise), scaling, rotation, etc.

However, time series have different properties than images and these methods might not be applicable to all time series. In addition, while some time series specific data augmentation methods exist, such as magnitude warping [16] and time warping [16], [17], they are still random transformations that carry assumptions about the patterns in the underlying dataset.

In order to tackle the problem of time series data augmentation, we propose the use of a new pattern mixing based augmentation, called *guided warping*. Guided warping combines the idea of time warping [16] with pattern mixing [18]. Specifically, to augment the training dataset with new samples, we warp the features of a *sample* to the time step relations of a *reference*. To align the features between the two time series, Dynamic Time Warping (DTW) [19] is used. While typically DTW is used as a distance measure, it can be used for its ability to align similar features while maintaining the temporal properties [20], [21]. Furthermore, we demonstrate that the dynamic warping can be further improved using shapeDTW [22] due to a smoother alignment by using high-level shape descriptors instead of element-wise alignment.

Furthermore, there is a question on how to choose the reference time series. Existing time series pattern mixing based data augmentation methods ten to either select the mixed patterns at random [18], [23], [24] or using a medoid [23]. We show that these are not necessarily the best strategies and propose a novel method of using a discriminator for selection, as shown in Fig 1. The reference pattern selected by the discriminator is referred to as a discriminative teacher and it is determined by finding the sample within a bootstrap set with the maximal distance between the patterns of the same class and patterns of a different class. For this work, we use a simple nearest centroid classifier on a small batch of random samples. Using a discriminative teacher rather than a random teacher, allows the guided warping to directly choose patterns that might aid the classifier.

The contributions are as follows:

- We demonstrate that using a DTW based warping is effective at producing samples for data augmentation.
- We show that the use of shapeDTW in the proposed guided warping can help maintain the original features of the sample time series and generate more suitable samples.
- We propose the use of a discriminative teacher time series instead of a random reference to be used as the basis of the warping.
- We do a thorough evaluation on all 85 datasets of the 2015 UCR Time Series Archive [8] using nine established time series data augmentation methods and the four proposed methods. The data augmentation methods are evaluated using a temporal 1D VGG [13] and a Long Short-Term Memory (LSTM) [25] network.

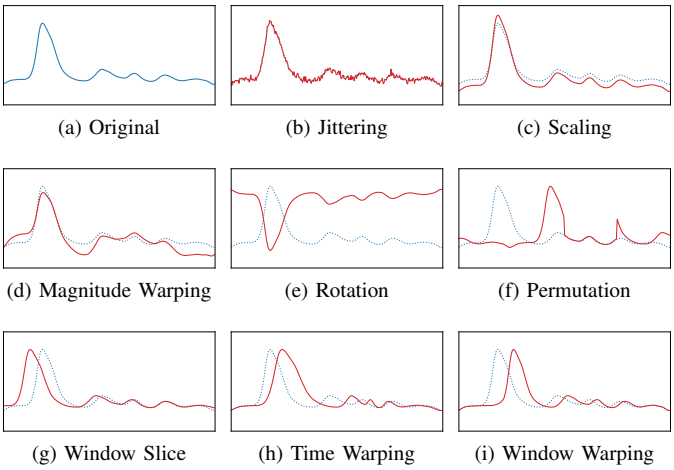


Fig. 2. A sample pattern from the 50words dataset in the UCR Time Series Archive. The blue line is the original time series and the red lines are transformed time series that are used as data augmentation.

II. Related Work

Most instances of time series data augmentation are random transformations. Some variations include *jittering* (noise addition), *rotation* (flipping for univariate; rotation for multivariate), *slicing* (cropping), *permutation* (rearranging slices), *scaling* (pattern-wise magnitude change), *magnitude warping* (smooth element-wise magnitude change), *time warping* (time step deformation), and *frequency warping* (frequency deformation). Examples of these are shown in Fig. 2. These random transformation based methods have been used for a wide variety of time series.

There are many examples of using random transformations for different applications. For example, Um et al. [16] used a variety of augmentations, such as jittering, scaling, rotation, slicing, permutation, magnitude warping, and time warping, to improve wearable sensor data for deep temporal CNNs. Rashid and Louis [17] recently used jittering, scaling, rotation, and time warping with LSTMs for construction equipment activity recognition. Frequency warping, in particular, is often used in acoustic recognition [26], [27]. In addition, there have been improvements on the random transformations. An example of this is *window warping* [28], which is a version of time warping that expands or contracts random windows of the data by fixed amounts.

Another approach is *pattern mixing*. In pattern mixing, instead of adding random transformations, multiple samples of the same class are mixed. In one example of pattern mixing, Takahashi et al. [18] adds together random segments of intra-class sounds at different ratios. However, adding two sequences together might result in out of phase overlapping or in the case of non-periodic time series, malformed patterns. Therefore, Forestier et al. [23] utilized DTW Barycentric Averaging (DBA) [20] to generate patterns. Specifically, DBA

is an iterative method of using DTW to align time series to find an average pattern with the features preserved. Forestier et al. proposed taking small subsets of the data and averaged them using a weighted DBA (wDBA). Other pattern mixing methods include using a randomly weighted DBA [29], averaging patterns with sub-optimal time warping [24], and stochastic feature mapping [27].

Finally, there are also many miscellaneous methods of generating data such as trained generative models [30] and handcrafted mathematical models [31]. In particular, Generative Adversarial Networks (GAN) [32] are a popular method of generating time series data, as they have shown to be useful for data augmentation [30]. However, the problem with these methods is that either they require domain specific properties (e.g. mathematical models) or they require external training (e.g. GANs).

The difference between the proposed method and these methods is that we attempt to address the data augmentation problem with as few assumptions as possible. The problem with many of the random transformations is that not all transformations are applicable to every dataset. For example, something simple as jittering carries the assumption that it is typical for the dataset to have noise. Adding jittering to an ECG dataset seems to fit, however, adding jittering to a dataset with only smooth shape outlines (such as the 50words dataset shown in Fig. 2) does not. In the 50words dataset, the heights of cursive words are mapped to a time series and the patterns created from jittering (Fig. 2 (b)) and rotation (Fig. 2 (e)) become unnatural. Pattern mixing augmentation can overcome these issues, but the previously proposed pattern mixing methods also have faults. For example, DBA is an effective method of averaging time series, but in wDBA averaging similar patterns might not help in increasing the distribution of patterns for better generalization.

III. Guided Warping

In this section, we will define DTW and propose the use of it as a method to time warp time series and how it can be used for data augmentation.

A. Dynamic Time Warping

DTW [19] is a classic, yet, effective method of determining an optimized distance measure for time series. Consider two time series $\mathbf{r} = r_1, \ldots, r_i, \ldots, r_I$ and $\mathbf{s} = s_1, \ldots, s_j, \ldots, s_J$ with sequence lengths I and J, respectively. Elements r_i and s_j at sequence indices i and j can be univariate or multivariate with dimensions $r_i = (\alpha_1, \ldots, \alpha_u, \ldots, \alpha_U)^\top$ and $s_j = (\beta_1, \ldots, \beta_v, \ldots, \beta_V)^\top$, respectively. Given the two sequences, $\mathbf{p}$ and $\mathbf{s}$, DTW can be used to determine the global distance between them. This global distance is robust to issues such as temporal distortions and has had many successes as a distance measure [33].

The key feature of DTW is that it non-linearly matches time series elements in the time dimension in order to match features and remove time distortions. It does this by warping the sequences so that there is an optimized alignment between elements that minimizes the global cost under constraints. Specifically, DTW finds the minimal path on an element-wise cost matrix C using dynamic programming. This minimal path is referred to as the *warping path* and the warping path becomes a mapping for the time steps of one series to the time steps of another. To solve for the minimal path, a minimal cumulative sum matrix is calculated using the recurrent function:

$$D(i,j) = C(r_i, s_j) + \min_{(i',j') \in \{(i,j-1),(i-1,j),(i-1,j-1)\}} D(i',j'), \quad (1)$$

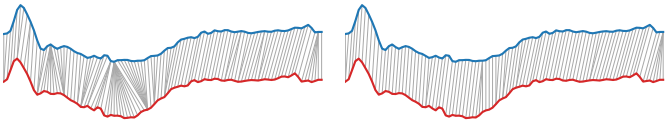
where $D(i,j)$ is the cumulative sum of the i-th and j-th elements and $C(r_i, s_j)$ is the local distance between r_u and s_t. In this paper, we use the Euclidean distance, or $C(r_i, s_j) = ||r_i - s_j||$, as the cost function. In the typical use of DTW, the global distance is defined as the value at $D(I,J)$. It should be noted that the slope constraint defined in Eq. (1) is a symmetric slope constraint [19] and other slope constraints, such as asymmetric and weighted constraints, have been proposed [34]. However, the proposed method can work with any variation. The symmetric slope constraint was selected due to being the most commonly used, but the proposed method can work with any variation of DTW.

B. Guided Warping for Data Augmentation

The purpose of data augmentation for time series is to generate patterns that extend the data in order to improve generalization. Namely, given training set $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n, \ldots, \mathbf{s}_N\}$ with individual time series $\mathbf{s}_n$, our goal is to create augmented set $\mathbf{S}'$ such that the accuracy of a model trained on $\mathbf{S} \cup \mathbf{S}'$ is greater than $\mathbf{S}$ alone. However, the patterns of $\mathbf{S}'$ need to be similar to the original feature distribution of $\mathbf{S}$ as to not introduce too much noise or create illogical patterns.

To generate time series, we propose the use of guided warping, or using guidance to instruct warping in the time domain based on other reference patterns. Typically, time warping for data augmentation is done using random warping [16] or random windows [28]. However, instead of randomly time warping, we propose creating augmentation set $\mathbf{S}'$ using patterns from training set $\mathbf{S}$ that are time warped using the guidance of other intra-class patterns from $\mathbf{S}$. Doing so is a form of pattern mixing where we preserve the features of student time series $\mathbf{s}$ and set it to the pace of teacher time series $\mathbf{r}$. The advantage of warping using a reference over randomly warping is that both the local features and the time steps they occur at exist in the original dataset. Randomly warping only hopes that the generated patterns are realistic.

To align the elements of $\mathbf{s}$ and $\mathbf{r}$ so guided warping can take place, we use DTW. As explained previously, traditionally, DTW is used as a distance measure for finding the global distance between the two. However, similar to other works [20], [21], [35], we can exploit the warping path to align the elements of the sequences. By aligning the elements in this way, sections of $\mathbf{s}$ are warped in the time dimension to fit $\mathbf{r}$.



(a) DTW (b) shapeDTW

Fig. 3. Comparison between alignment with (a) DTW and (b) shapeDTW. The red and blue lines are time series and the gray connections are the element alignments.

Specifically, a minimal cumulative sum matrix for DTW is calculated using Eq. (1) and the minimum warping path is found by tracing $D(I, J)$ back to the origin $D(0, 0)$ through matched elements (i', j'). Next, time series $\mathbf{s}'$ is created by warping $\mathbf{s}$ by the time steps $1, \ldots, j', \ldots, J'$. The result is a sequence $\mathbf{s}'$ that has the feature values of $\mathbf{s}$ but the time steps of $\mathbf{r}$ under the warping path constraints provided by DTW. Finally, the process is repeated by selecting any two random patterns in $\mathbf{S}$ that are the same class. Using this method, it is possible to synthesize $\sum_y N_y^2$ number of time series where N_y is the number of patterns in each class y.

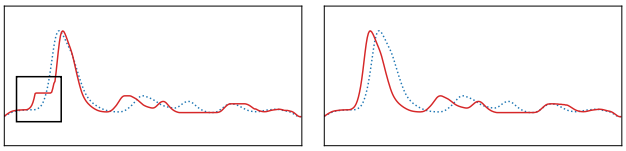
IV. Improved Time Series Generation with Shape Descriptors

DTW finds the optimal alignment between individual elements by minimizing the global cost between elements on the warping path. While this is effective for distance measures, this might not be optimal for pattern generation. Using a strict minimal optimization can produce jagged and abrupt changes in the warping path, as shown in Fig. 3. To overcome this problem, we propose using a method of alignment based on high-level features, namely shapeDTW [22].

Instead of using element-wise matching as in DTW, shapeDTW dynamically matches shape descriptors within the sequence. Given time series $\mathbf{s}$, a shape descriptor is a multivariate vector $d_{s_j} = (s_{j-\lceil \frac{1}{2}W \rceil}, \ldots, s_j, \ldots, s_{j+\lfloor \frac{1}{2}W \rfloor})^\top$ created from a subsequence of $\mathbf{s}$ with a length of W, centered on element j. Using a stride of 1, a new sequence of shape descriptors $\mathbf{d_s} = d_{s_1}, \ldots, d_{s_j}, \ldots, d_{s_J}$ is created of equal length to the original $\mathbf{s}$. Also, padding is added for $j - \lceil \frac{1}{2}W \rceil < 1$ and $j + \lfloor \frac{1}{2}W \rfloor > J$ with duplicates of s_1 and s_J, respectively. These shape descriptors represent higher-level features (i.e. segments of a time series) than the individual elements themselves.

ShapeDTW proceeds with a standard DTW calculation, however, with shape descriptor sequences $\mathbf{d_s}$ and $\mathbf{d_p}$ instead of the raw elements. In other words, time series $\mathbf{s}$ and $\mathbf{p}$ are temporally aligned using their shape descriptors. Essentially, instead of the element-wise cost function $C(p_i, s_j) = ||p_i - s_j||$ in Eq. (1) of DTW, shapeDTW uses a cost function $C(p_i, s_j) = ||d_{p_i} - d_{s_j}||$ between the shape descriptors. The result is a non-linear alignment much like DTW, but using the similarity between neighboring points to the elements.

Similar to Section III-B, we can exploit the alignment byproduct of shapeDTW to generate new time series. Using shapeDTW, a new pattern is created by warping the features of $\mathbf{s}$ to the time steps of $\mathbf{p}$. Compared to guided warping with standard DTW, we are able to preserve the features of the original pattern and create more natural time series. For example, in Fig. 4, the boxed region contains an abnormal feature that caused by overzealous minimization of the warping path by DTW.



(a) Warping with DTW (b) Warping with shapeDTW

Fig. 4. The difference between (a) warping using DTW and (b) warping using shapeDTW. The boxed region in (a) highlights an unnatural feature created from the dynamic alignment.

V. Discriminative Teacher Selection

While the reference prototype $\mathbf{r}$ can be chosen at random (within the same class), we posit that using a directed prototype is better than a random one. In this case, we propose to select the *most discriminative* prototype from a bootstrap set of samples to be the reference, for DTW. The bootstrap set is used to represent the distribution of the training samples with $B \ll N$ patterns selected at random with replacement, where B is the number of bootstrap samples and N is the total training set size.

A. Formulation

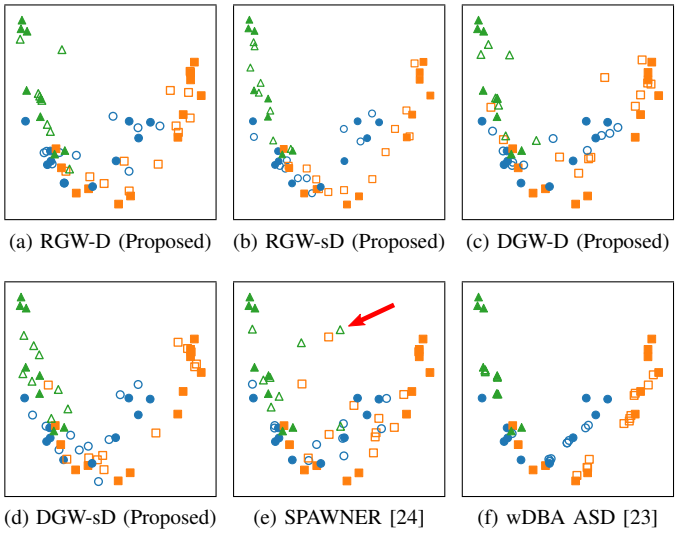
To determine the most discriminative teacher from the bootstrap set, we use a nearest centroid classifier based on DTW (or shapeDTW) distance [36]. Specifically, for each time series $\mathbf{b}_m$ in $\mathbf{B} \subset \mathbf{S}$, with subset size M:

$$h(\mathbf{b}_m) = \frac{1}{\sum_{m'} [l_{m'} \neq l_m]} \sum_{m'} \mathcal{D}(\mathbf{b}_{m'}, \mathbf{b}_m) | [l_{m'} \neq l_m] - \frac{1}{\sum_{m'} [l_{m'} = l_m]} \sum_{m'} \mathcal{D}(\mathbf{b}_{m'}, \mathbf{b}_m) | [l_{m'} = l_m], \quad (2)$$

where $l_{m'}$ is the label for each $\mathbf{b}_m$ and the hypothesis is $\text{sign}(h(\mathbf{b}_m))$. Due to the high possibility of ties, instead of using the prediction $\text{sign}(h(\mathbf{b}_m))$, we use the reference with the maximal distance between the positive and negative centroids, in other words:

$$\mathbf{b}_{\text{disc}} = \underset{\{m=1,\ldots,M\}}{\text{argmax}} \; h(\mathbf{b}_m), \quad (3)$$

where $\mathbf{b}_{\text{disc}}$ is the reference that we define as the most discriminative in $\mathbf{B}$. Also, it should be noted that to ensure successful selection, we sample from $\mathbf{S}$ as close to evenly for $\mathbf{B}$ from same class patterns $l_{m'} = l_m$ and different class patterns $l_{m'} \neq l_m$ as possible. Using the selected discriminative teacher $\mathbf{b}_{\text{disc}}$, $\mathbf{s}$ is warped using DTW (or shapeDTW) as described previously. We distinguish the two proposed methods by referring to the random selection as Random Guided Warping (RGW) and the discriminative teacher selection as Discriminative Guided Warping (DGW) with the DTW variants (-D) referred to as RGW-D and DGW-D and



(a) RGW-D (Proposed) (b) RGW-sD (Proposed) (c) DGW-D (Proposed)

(d) DGW-sD (Proposed) (e) SPAWNER [24] (f) wDBA ASD [23]

Fig. 5. Visualization of the CBF dataset using PCA using the proposed augmentation and the other pattern mixing methods. The solid shapes are the original time series and the hollow shapes are the generated time series. The arrow points to examples generated patterns that do not fit the expected distribution of features in the dataset.

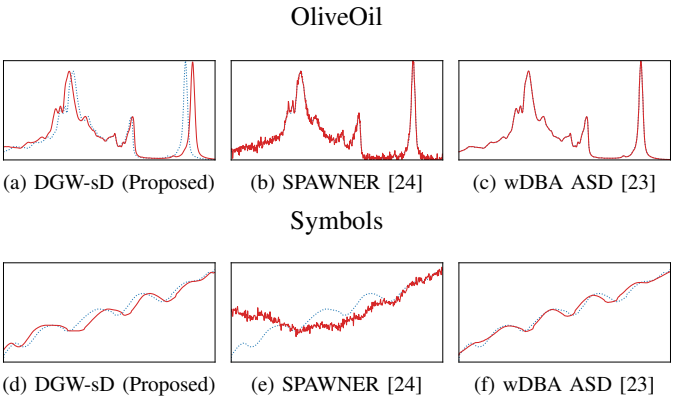
the shapeDTW variants (-sD) as RGW-sD and DGW-sD, respectively.

B. Visualization

To demonstrate the effects of the proposed selection process, the data can be visualized using Principal Component Analyis (PCA) [37]. For Fig. 5, CBF is selected as an example time series dataset due to having a very small amount of samples (only 30). In the figure, each color represents one class with the solid shapes being original samples and the hollow shapes being generated examples.

The figure highlights the differences between the proposed methods and the two other DTW based pattern mixing data augmentation methods. Since RGW (Fig. 5 (a)) and SPAWNER [24] (Fig. 5 (e)) use randomly selected references, it is possible to generate patterns that seem to not fit with the data, such as the patterns in the center of the "U" shape. Using a discriminative teacher (Fig. 5 (c)) can help direct the warping toward useful patterns. Furthermore, incorporating shapeDTW helps both methods in maintaining more consistency in the data distribution. As for wDBA ASD [23] (Fig. 5 (f)), the generated patterns are too similar to existing patterns to be useful for data augmentation. Using shapeDTW with DGW, in Fig. 5 (d), provides the best balance of maintaining the data distribution while producing new patterns.

Specific examples of these observations can be seen in Fig. 6. In Fig. 6 (c) and (f), The time series generated by wDBA are too similar to the original time series. Whereas, SPAWNER created patterns that are not typical to the dataset. Especially in the case of Fig. 6 (e), where two patterns of the same class that were drastically different from each other were selected. DGW-sD was able to generate time series that fit the distribution of the dataset but are different enough to be useful for augmentation.



(a) DGW-sD (Proposed) (b) SPAWNER [24] (c) wDBA ASD [23]

(d) DGW-sD (Proposed) (e) SPAWNER [24] (f) wDBA ASD [23]

Fig. 6. Examples of generated patterns. The blue dotted line is the original time series and the red solid lines are the generated time series.

VI. Experimental Results

A. Datasets

We want to assess the effects that the proposed method and the other data augmentation techniques have on a wide variety of datasets. Therefore, we used all 85 datasets of the 2015 UCR Time Series Archive [8]. The datasets are a collection of 6 electric device, 6 ECG, 29 image outline, 14 motion capture, 18 sensor reading, 5 simulated, and 7 spectrograph time series. They have fixed training and test sets with between 16 and 8,926 training samples and between 20 and 8,236 test samples. Furthermore, the time series lengths range between 24 and 2,709 time steps. For pre-processing, each dataset is normalized so that the largest and smallest values in the training dataset is 1 and -1, respectively.

B. Augmentation Methods

To evaluate the proposed method, we used nine general time series data augmentation techniques found from literature. The parameters of each of the comparison augmentation methods were set to be the same as used by the respective works. The following comparison methods were used for evaluations:

- **None**: Uses no augmentation for a baseline.
- **Jittering (Jit)**: Random noise from a Gaussian distribution with a mean $\mu = 0$ and a standard deviation $\sigma = 0.03$, as suggested in [16], is added to the original time series.
- **Rotation (Rot)**: For rotation, since the patterns in the UCR Time Series Archive are univariate, patterns are randomly flipped.
- **Scaling (Scal)**: In scaling, the magnitude of all elements in the time series is increased or decreased by a scalar. As in [16], the scalar is determined by a Gaussian distribution with $\mu = 1$ and $\sigma = 0.1$.
- **Magnitude Warping (MagW)**: The magnitude of each time series is multiplied by a curve created by cubic spline with four knots at random magnitudes with $\mu = 1$ and $\sigma = 0.2$ [16].
- **Time Warping (TimW)**: Time warping based on a random smooth warping curve generated by cubic spline

TABLE I
TEST ACCURACY (%) GROUPED BY TIME SERIES TYPE

Type	Comparative Augmentation with VGG										Proposed Augmentation with VGG			
(count)	None	Jit	Rot	Scal	MagW	TimW	Slic	WinW	SPAWNER	wDBA	RGW-D	RGW-sD	DGW-D	DGW-sD
Device (6)	54.9	54.9	56.4	57.5	56.4	58.7	**62.1**	57.6	58.5	56.5	60.4	59.8	60.0	59.5
ECG (6)	93.7	93.7	93.6	93.8	**93.9**	90.2	93.1	93.0	92.9	88.4	92.8	91.9	92.2	92.2
Image (29)	75.4	78.3	78.1	74.3	78.6	79.8	80.2	**81.5**	78.8	78.2	80.1	80.3	81.0	81.1
Motion (14)	70.7	69.9	66.7	68.7	70.3	73.1	70.1	72.7	71.3	71.8	72.3	73.2	**73.9**	73.7
Sensor (18)	80.3	80.4	79.7	79.3	80.4	79.5	79.9	81.2	80.2	79.7	80.3	80.8	**81.7**	80.8
Sim. (5)	89.1	89.1	73.7	86.8	88.3	91.0	94.8	96.6	96.5	87.9	89.4	89.5	91.9	**98.6**
Spectro (7)	76.9	77.3	81.5	79.0	**86.0**	74.4	82.5	76.5	83.5	80.7	85.9	81.4	83.5	81.7
Total (85)	76.44	77.32	74.84	77.06	78.30	78.10	79.15	79.58	78.84	77.42	79.39	79.27	80.12	**80.17**
Type	Comparative Augmentation with LSTM										Proposed Augmentation with LSTM			
(count)	None	Jit	Rot	Scal	MagW	TimW	Slic	WinW	SPAWNER	wDBA	RGW-D	RGW-sD	DGW-D	DGW-sD
Device (6)	40.8	43.0	41.1	41.7	43.4	43.5	**44.6**	42.5	42.5	42.5	44.2	44.3	**44.6**	42.3
ECG (6)	57.6	70.3	61.1	**78.7**	63.4	49.2	61.0	55.1	65.0	59.5	61.1	55.6	59.6	59.1
Image (29)	62.2	61.5	59.4	57.8	63.5	54.7	55.0	62.4	**63.6**	58.8	63.2	62.0	57.5	60.1
Motion (14)	45.0	45.5	37.7	42.6	43.8	41.4	40.1	**48.4**	43.1	44.6	38.4	43.5	43.7	43.9
Sensor (18)	59.2	62.2	58.1	61.5	61.4	57.6	61.5	60.3	62.0	61.4	**63.2**	61.4	62.9	63.1
Sim. (5)	72.6	70.5	68.6	**76.6**	72.8	72.7	63.7	68.8	76.1	74.2	72.3	62.2	63.6	73.8
Spectro (7)	58.9	55.4	57.9	**61.6**	54.1	52.5	55.6	52.8	60.5	49.0	54.1	53.4	58.3	53.3
Total (85)	57.24	58.35	54.78	57.98	58.04	52.80	54.08	57.49	**58.98**	56.01	57.42	56.43	56.01	56.99

with four knots at random magnitudes ($\mu = 1$, $\sigma = 0.2$). We followed the same procedure as Um et al. [16]

- **Slicing (Slic)**: For this augmentation method, we use window slicing [28]. In window slicing, a window of 90% of the original time series is chosen at random. In our implementation, we interpolate this back to the original size to fit with the classifier.
- **Window Warping (WinW)**: Window warping [28] selects a random window of 10% of the original data and either speeds it up by 2 or slows it down by 0.5.
- **Suboptimal Warped Time Series Generator (SPAWNER)**: SPAWNER [24] is a pattern mixing method that creates a time series from the average of two random suboptimally aligned intra-class patterns. Furthermore, as recommended, noise is added to the average with a $\sigma = 0.5$ in order to avoid instances where there is very little change.
- **wDBA**: We use the Average Selected with Distance (ASD) version due to it having the best results in [23]. In this data augmentation method, 6 patterns weighted by their DTW distance to the medoid are averaged using DBA.

As for the proposed method, we used the following evaluations:

- **Random Guided Warping with DTW (RGW-D)**: RGW-D is the proposed method described in Section III-B which selects two intra-class patterns and warps the features of one pattern by the time steps of the second.
- **Random Guided Warping with shapeDTW (RGW-sD)**: This evaluation is used to show the effects of using shapeDTW to warp the high-level features. It follows the same procedure as RGW-D but with shapeDTW. The length of the shape descriptor was set to $W = \frac{1}{20} \times J$ with a minimum of $W = 5$ and a maximum of $W = 100$.
- **Discriminative Guided Warping with DTW (DGW-D)**: This is the proposed augmentation process of using a discriminative teacher as the reference for guided warping, as described in Section V. We use $M = 6$ for the bootstrap batch size.
- **Discriminative Guided Warping with shapeDTW (DGW-sD)**: DGW-sD is DGW-D but with shapeDTW as the distance measure.

For each data augmentation technique, each training set is augmented with $4\times$ the original set size. Also, for the methods that incorporate DTW into their algorithms, we used the symmetric slope constraint defined by Eq. (1) with a warping window of 10% of the original time series length.

C. Evaluation Settings

The proposed method is evaluated on two established temporal neural network models, a VGG [13] and an LSTM [25]. A separate network is trained and tested using each dataset with each augmentation technique and one extra with no augmentation. They are all trained with a fixed 10,000 iterations in order to have a fair comparison between the difference between no augmentation and having augmentation.

The VGG used in our evaluation is modified for time series by using 1D convolutions and max pooling instead of the traditional 2D convolutions and pooling for images. It consists of multiple blocks of convolutional layers followed by max pooling layers, two fully-connected layers of 4,096 nodes with a dropout probability of 0.5, and an output layer. Each layer has a Rectified Linear Unit (ReLU) activation function except the output which uses softmax. The first two

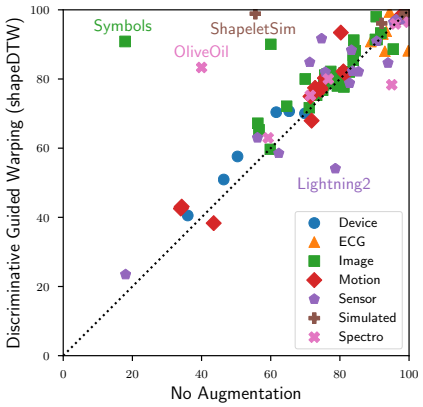


Fig. 7. A comparison of no augmentation and the proposed method with a VGG. Each point is a dataset and the shapes indicate dataset type.

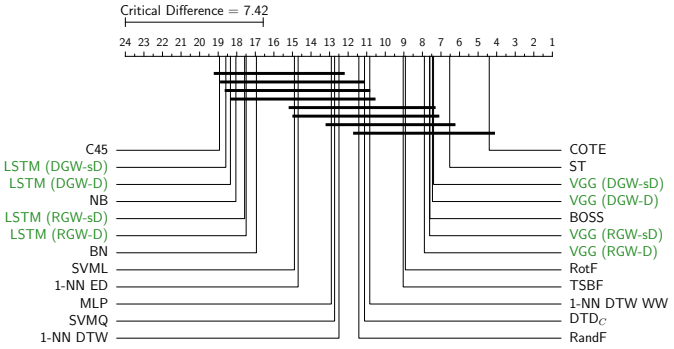


Fig. 8. Critical difference diagram for the proposed methods (green) and 16 benchmarks.

blocks of convolutional layers are made of two consecutive convolutional layers with 64 and 128 nodes, respectively, and the subsequent blocks have three consecutive convolutional layers of 256 nodes for the third and 512 nodes, thereafter. Due to the differences in the sequence lengths of the datasets, we use a varying number of blocks in order to prevent excessive pooling. The number of blocks G used is:

$$G = \text{round}(\log_2(J)) - 3, \tag{4}$$

where J is the number of time steps. This scheme keeps the output of the final max pooling to between 5 and 12 time steps [38]. For training, we use Stochastic Gradient Decent (SGD) with an initial learning rate of 0.01, momentum of 0.9, and weight decay of 5×10^{-4}. The learning rate is reduced by 0.1 upon training accuracy plateau. In addition, the VGG is trained with mini-batches of 256. These settings are used to match [13].

For the LSTM, we use some of the recommendations from Reimers et al. [39]. Specifically, we use a two-layer LSTM with 100 cells each. As following [39], we use an Nadam [40] optimizer with an initial learning rate of 0.001 and mini-batch size 32. We also use a learning rate reduction of 0.1 upon training accuracy plateau for the LSTM.

D. Data Augmentation Comparison Results

The results comparing the data augmentation techniques are shown in Table I. For VGG, there are significant improvements using the proposed data augmentation for most of the datasets. Overall, there was an average of 3-4% increase in accuracy for all of the guided warping methods. Of the proposed methods DGW-sD had the highest overall. In addition, using the discriminative prototype as a reference versus a random reference improved the accuracy for both DTW and shapeDTW. The highest gains for DGW-sD were from Simulated, Image outline, and Device time series with increases of accuracy of 9.5%, 6.2%, and 5.1%, respectively. However, there were dataset categories that did not improve. ECG had a slight degradation in accuracy and Sensor had insignificant improvements. Although, it is notable that ECG and Sensor datasets were not significantly affected by any of the data augmentation methods with VGG.

On the other hand, from Table I, there is no clear trend or advantage in using the proposed method with an LSTM. In general, the time domain augmentations (TimW, Slic, WinW, and the proposed methods) tended not to improve the trained LSTM models. This is due to LSTMs, and RNNs in general, being designed to explicitly combat time distortions. Thus, it is more recommended to use the proposed method with CNN based architectures.

The individual differences between the accuracies of each dataset for no augmentation and DGW-sD for VGG is shown in Fig. 7. The figure shows which kind of datasets improved with DGW-sD data augmentation. As stated previously, image outlines, device, and simulated datasets improved the most. Notably, Symbols, OliveOil, and ShapeletSim had the largest increase in accuracy over no augmentation. For some of these datasets, we can expect increases in accuracy. Symbols is made of online hand-drawn symbols so maintaining the shape structure is important. In addition, ShapeletSim is created from shapelets embedded within time series, thus, using guided warping could help increase the generalization by moving the shapelets around in the time dimension.

E. Model Accuracy Comparison

In Fig. 8, we compare the models with the proposed data augmentation to some of the state-of-the-art methods found in literature. In the past, there have been many proposed methods of classifying the 2015 UCR Time Series Archive. In order to collect comparisons, we use the best classifier from each category of the great time series classification bake off [41] plus all of their baseline methods. We also include the classic 1-Nearest Neighbor (1-NN) comparisons from [8]. Of the reported comparison methods, only Collection of Transformation Ensembles (COTE) [42] and Shapelet Transform (ST) [43] had better results than the proposed augmentation with VGG. However, our purpose is not to specifically get the best results overall, but instead to propose a state-of-the-art data augmentation method that can be used with any classifier.

VII. Conclusion

In this work, we proposed a new data augmentation method for time series based on time warping using DTW. We use the time step relationships from the warping path generated by DTW to warp one time series to a second reference time series. In doing so, the generated pattern has the local features of the first time series and the time step properties of the second time series.

Furthermore, we demonstrate that the proposed method can be extended in two ways. First, we apply shapeDTW in order to preserve the relationships between neighboring elements in the warping. Second, the results are further improved by using the most discriminative sample in a batch as a teacher. By using specific discriminative samples as references, we are able to direct the augmentation to target useful teachers.

In the future, we plan on applying the proposed data augmentation method to new applications and new models. In addition, we will further explore the properties and characteristics of guided warping. An implementation of the proposed method can be found at https://github.com/uchidalab/time_series_augmentation.

Acknowledgment

This research was partially supported by MEXT-Japan (Grant No. J17H06100).

References

[1] H. Touvron, A. Vedaldi, M. Douze, and H. Jegou, "Fixing the train-test resolution discrepancy," in *NeurIPS*, 2019, pp. 8252–8262.

[2] S. Edunov, M. Ott, M. Auli, and D. Grangier, "Understanding back-translation at scale," in *CEMNLP*, 2018, pp. 489–500.

[3] S. Verma and Z.-L. Zhang, "Learning universal graph neural network embeddings with aid of transfer learning," *arXiv preprint arXiv:1909.10086*, 2019.

[4] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.

[5] D. Brain and G. Webb, "On the effect of data set size on bias and variance in classification learning," in *AKAW*, 1999, pp. 117–128.

[6] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *AMACL*, 2001.

[7] A. Torralba, R. Fergus, and W. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, 2008.

[8] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," 2015, www.cs.ucr.edu/~eamonn/time_series_data/.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[12] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, 2019.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016.

[15] Q. Wen, L. Sun, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," *arXiv preprint arXiv:2002.12478*, 2020.

[16] T. T. Um, F. M. J. Pfister, D. Pichler, S. Endo, M. Lang, S. Hirche, U. Fietzek, and D. Kulić, "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks," in *ACM ICMI*, 2017, pp. 216–220.

[17] K. M. Rashid and J. Louis, "Time-warping: A time series data augmentation of IMU data for construction equipment activity identification," in *ISARC*, 2019.

[18] N. Takahashi, M. Gygli, B. Pfister, and L. V. Gool, "Deep convolutional neural networks and data augmentation for acoustic event recognition," in *Interspeech*, 2016.

[19] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, and Sig. Process.*, vol. 26, no. 1, pp. 43–49, 1978.

[20] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recogn.*, vol. 44, no. 3, pp. 678–693, 2011.

[21] B. K. Iwana and S. Uchida, "Dynamic weight alignment for temporal convolutional neural networks," in *IEEE ICASSP*, 2019, pp. 3827–3831.

[22] J. Zhao and L. Itti, "shapeDTW: Shape dynamic time warping," *Pattern Recogn.*, vol. 74, pp. 171–184, 2018.

[23] G. Forestier, F. Petitjean, H. A. Dau, G. I. Webb, and E. Keogh, "Generating synthetic time series to augment sparse datasets," in *IEEE ICDM*, 2017.

[24] K. Kamycki, T. Kapuscinski, and M. Oszust, "Data augmentation with suboptimal warping for time-series classification," *Sensors*, vol. 20, no. 1, p. 98, 2019.

[25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] N. Jaitly and G. E. Hinton, "Vocal tract length perturbation (vtlp) improves speech recognition," in *ICML WDLASLP*, 2013.

[27] X. Cui, V. Goel, and B. Kingsbury, "Data augmentation for deep neural network acoustic modeling," in *IEEE ICASSP*, 2014.

[28] A. Le Guennec, S. Malinowski, and R. Tavenard, "Data augmentation for time series classification using convolutional neural networks," in *IWAATD*, 2016.

[29] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Data augmentation using synthetic data for time series classification with deep residual networks," in *IWAATD*, 2018.

[30] K. Nikolaidis, S. Kristiansen, V. Goebel, T. Plagemann, K. Liestøl, and M. Kankanhalli, "Augmenting physiological time series data: A case study for sleep apnea detection," in *ECML/PKDD*, 2019.

[31] F. Wendling, J. J. Bellanger, F. Bartolomei, and P. Chauvel, "Relevance of nonlinear lumped-parameter models in the analysis of depth-EEG epileptic signals," *Bio. Cybernetics*, vol. 83, no. 4, pp. 367–378, 2000.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014, pp. 2672–2680.

[33] C. A. Ratanamahatana and E. Keogh, "Everything you know about dynamic time warping is wrong," in *WMTSD*, 2004, pp. 53–63.

[34] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoustics, Speech, and Sig. Process.*, vol. 23, no. 1, pp. 67–72, 1975.

[35] X. Wu, A. Kimura, S. Uchida, and K. Kashino, "Prewarping siamese network: Learning local representations for online signature verification," in *IEEE ICASSP*, 2019, pp. 2467–2471.

[36] B. K. Iwana, V. Frinken, K. Riesen, and S. Uchida, "Efficient temporal pattern recognition by means of dissimilarity space embedding with discriminative prototypes," *Pattern Recogn.*, vol. 64, pp. 268–276, 2017.

[37] K. Pearson, "On lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philo. Mag. and J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901.

[38] B. K. Iwana and S. Uchida, "Time series classification using local distance-based features in multi-modal fusion networks," *Pattern Recogn.*, vol. 97, p. 107024, 2020.

[39] N. Reimers and I. Gurevych, "Optimal hyperparameters for deep lstm-networks for sequence labeling tasks," *arXiv preprint arXiv:1707.06799*, 2017.

[40] T. Dozat, "Incorporating nesterov momentum into adam," 2016.

[41] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowl. Discovery*, vol. 31, no. 3, pp. 606–660, 2016.

[42] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: The collective of transformation-based ensembles," in *IEEE ICDE*, 2016.
[43] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data Mining and Knowl. Discovery*, vol. 28, no. 4, pp. 851–881, 2013.