Time Series Data Augmentation for Deep Learning: A Survey

Qingsong Wen¹, Liang Sun¹, Fan Yang², Xiaomin Song¹, Jingkun Gao³*, Xue Wang¹, Huan Xu²
¹DAMO Academy, Alibaba Group, Bellevue, WA, USA
²Alibaba Group, Hangzhou, China
³Twitter, Seattle, WA, USA
{qingsong.wen, liang.sun, fanyang.yf, xiaomin.song, xue.w, huan.xu}@alibaba-inc.com,
jingkung@twitter.com

Abstract

Deep learning performs remarkably well on many time series analysis tasks recently. The superior performance of deep neural networks relies heavily on a large number of training data to avoid overfitting. However, the labeled data of many real-world time series applications may be limited such as classification in medical time series and anomaly detection in AIOps. As an effective way to enhance the size and quality of the training data, data augmentation is crucial to the successful application of deep learning models on time series data. In this paper, we systematically review different data augmentation methods for time series. We propose a taxonomy for the reviewed methods, and then provide a structured review for these methods by highlighting their strengths and limitations. We also empirically compare different data augmentation methods for different tasks including time series classification, anomaly detection, and forecasting. Finally, we discuss and highlight five future directions to provide useful research guidance.

1 Introduction

Deep learning has achieved remarkable success in many fields, including computer vision (CV), natural language processing (NLP), and speech processing, etc. Recently, it is increasingly embraced for solving time series related tasks, including time series classification [Fawaz et al., 2019], time series forecasting [Han et al., 2019], and time series anomaly detection [Gamboa, 2017]. The success of deep learning relies heavily on a large number of training data to avoid overfitting. Unfortunately, many time series tasks do not have enough labeled data. As an effective tool to enhance the size and quality of the training data, data augmentation is crucial to the successful application of deep learning models. The basic idea of data augmentation is to generate synthetic dataset covering unexplored input space while maintaining correct labels. Data augmentation has shown its effectiveness in many applications, such as AlexNet [Krizhevsky et al., 2012] for ImageNet classification.

However, less attention has been paid to find better data augmentation methods specifically for time series data. Here we highlight some challenges arising from data augmentation methods for time series data. Firstly, the intrinsic properties of time series data are not fully utilized in current data augmentation methods. One unique property of time series data is the so-called temporal dependency. Unlike image data, the time series data can be transformed into the frequency and time-frequency domains and effective data augmentation methods can be designed and implemented in the transformed domain. This becomes more complicated when we model multivariate time series where we need to consider the potentially complex dynamics of these variables across time. Thus, simply applying those data augmentation methods from image and speech processing may not result in valid synthetic data. Secondly, the data augmentation methods are also task dependent. For example, the data augmentation methods applicable for time series classification may not be valid for time series anomaly detection. In addition, data augmentation becomes more crucial in many time series classification problems where class imbalance is often observed. In this case, how to effective generate a large number of synthetic data with labels with less samples remains a challenge.

Unlike data augmentation for CV [Shorten and Khoshgoftaar, 2019] or speech [Cui et al., 2015], data augmentation for time series has not yet been comprehensively and systematically reviewed to the best of our knowledge. One work closely related to ours is [Iwana and Uchida, 2020] which presents a survey of existing data augmentation methods for time series classification. However, it does not review the data augmentation methods for other common tasks like time series forecasting [Bandara et al., 2020; Hu et al., 2020; Lee and Kim, 2020] and anomaly detection [Lim et al., 2018; Zhou et al., 2019; Gao et al., 2020]. Furthermore, the potential avenues for future research opportunities of time series data augmentations are also missing.

In this paper, we aim to fill the aforementioned gaps by summarizing existing time series data augmentation methods in common tasks, including time series forecasting, anomaly detection, classification, as well as providing insightful future directions. To this end, we propose a taxonomy of data augmentation methods for time series, as illustrated in Fig. 1. Based on the taxonomy, we review these data augmentation methods systematically. We start the discussion from the sim-
Data Augmentation

<table>
<thead>
<tr>
<th>Time Domain</th>
<th>Frequency Domain</th>
<th>Time-Freq Domain</th>
<th>Decomposition Methods</th>
<th>Statistical Generative Models</th>
<th>Learning Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cropping, Flipping, Jittering, ...</td>
<td>Embedding Space</td>
<td>Deep Generative Models</td>
<td>Automated Data Augmentation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 1: A taxonomy of time series data augmentation techniques.

### 2 Basic Data Augmentation Methods

#### 2.1 Time Domain

The transforms in the time domain are the most straightforward data augmentation methods for time series data. Most of them manipulate the original input time series directly, like injecting Gaussian noise or more complicated noise patterns such as spike, step-like trend, and slope-like trend. Besides this straightforward methods, we will also discuss a particular data augmentation method for time series anomaly detection, i.e., label expansion in the time domain.

Window cropping or slicing has been mentioned in [Le Guennec et al., 2016]. Introduced in [Cui et al., 2016], window cropping is similar to cropping in CV area. It is a subsample method to randomly extract continuous slices from the original time series. The length of the slice is a tunable parameter. For classification problem, the labels of sliced samples are the same as the original time series. During test time, each slice from a test time series is classified using majority voting. For anomaly detection problem, the anomaly label will be sliced along with value series.

Window warping is a unique augmentation method for time series. Similar to dynamic time warping (DTW), this method selects a random time range, then compresses (down sample) or extends (up sample) it, while keeps other time range unchanged. Window warping would change the total length of the original time series, so it should be conducted along with window cropping for deep learning models. This method contains the normal down sampling which takes down sample through the whole length of the original time series.

Flipping is another method that generates the new sequence \( x_1', \ldots, x_N' \) by flipping the sign of original time series \( x_1, \ldots, x_N \), where \( x_i = -x_i \). The labels are still the same, for both anomaly detection and classification, assuming that we have symmetry between up and down directions.

Another interesting perturbation and also ensemble based method is introduced in [Fawaz et al., 2018]. This method generates new time series with DTW and then ensembles them by a weighted version of the Barycentric Averaging (DBA) algorithm. It shows improvement of classification in some of the UCR datasets.

Noise injection is a method by injecting small amount of noise/outlier into time series without changing the corresponding labels. This includes injecting Gaussian noise, spike, step-like trend, and slope-like trend, etc. For spike, we can randomly pick index and direction, randomly assign magnitude but bounded by multiples of standard deviation of the original time series. For step-like trend, it is the cumulative summation of the spikes from left index to right index. The slope-like trend is adding a linear trend into the original time series. These schemes are mostly mentioned in [Wen and Keyes, 2019].

In time series anomaly detection, the anomalies generally last long enough during a continuous span so that the start and end points are sometimes “blurry”. As a result, a data point close to a labeled anomaly in terms of both time distance and value distance is very likely to be an anomaly. In this case, the label expansion method is proposed to change those data points and their labels as anomalies (by assign it an anomaly score or switch its label), which brings performance improvement for time series anomaly detection as shown in [Gao et al., 2020].

#### 2.2 Frequency Domain

While most of the existing data augmentation methods focus on time domain, only a few studies investigate data augmentation from frequency domain perspective for time series.

A recent work in [Gao et al., 2020] proposes to utilize perturbations in both amplitude spectrum and phase spectrum in frequency domain for data augmentation in time series anomaly detection by convolutional neural network. Specifically, for the input time series \( x_1, \ldots, x_N \), its frequency spectrum \( F(\omega_k) \) through Fourier transform is calculated as:

\[
F(\omega_k) = \frac{1}{N} \sum_{t=0}^{N-1} x_t e^{-j\omega_k t} = A(\omega_k) \exp[j\theta(\omega_k)]
\]

where \( \omega_k = \frac{2\pi k}{N} \) is the angular frequency, \( A(\omega_k) \) is the amplitude spectrum, and \( \theta(\omega_k) \) is the phase spectrum. For perturbations in amplitude spectrum \( A(\omega_k) \), the amplitude values of randomly selected segments are replaced with Gaussian noise by considering the original mean and variance in the amplitude spectrum. While for perturbations in phase spectrum \( \theta(\omega_k) \), the phase values of randomly selected segments are added by an extra zero-mean Gaussian noise in the phase spectrum. The amplitude and phase perturbations (APP) based data augmentation combined with aforementioned time-domain augmentation methods bring significant improvement.
time series anomaly detection improvements as shown in the experiments of [Gao et al., 2020].

Another recent work in [Lee et al., 2019] proposes to utilize the surrogate data to improve the classification performance of rehabilitative time series in deep neural networks. Two conventional types of surrogate time series are adopted in the work: the amplitude adjusted Fourier transform (AAFT) and the iterated AAFT (IAAFT) [Schreiber and Schmitz, 2000]. The main idea is to perform random phase shuffle in phase spectrum after Fourier transform and then perform rank-ordering of time series after inverse Fourier transform. The generated time series from AAFT and IAAFT can approximately preserve the temporal correlation, power spectra, and the amplitude distribution of the original time series. In the experiments of [Lee et al., 2019], the authors conducted two types of data augmentation by extending the data by 10 then 100 times through AAFT and IAAFT methods, and demonstrated promising classification accuracy improvements compared to the original time series without data augmentation.

2.3 Time-Frequency Domain

Time-frequency analysis is a widely applied technique for time series analysis, which can be utilized as an appropriate input features in deep neural networks. However, similar to data augmentation in frequency domain, only a few studies considered data augmentation from time-frequency domain for time series.

The authors in [Steven Eyobu and Han, 2018] adopt short Fourier transform (STFT) to generate time-frequency features for sensor time series, and conduct data augmentation on the time-frequency features for human activity classification by a deep LSTM neural network. Specifically, two augmentation techniques are proposed. One is the local averaging based on a defined criteria with the generated features appended at the tail end of the feature set. Another is the shuffling of feature vectors to create variation in the data. Similarly, in speech time series, recently SpecAugment [Park et al., 2019] is proposed to make data augmentation in Mel-Frequency (a time-frequency representation based on STFT for speech time series), where the augmentation scheme consists of warping the features, masking blocks of frequency channels, and masking blocks of time steps. They demonstrate that SpecAugment can greatly improve the performance of speech recognition neural networks and obtain state-of-the-art results.

For illustration, we summarize several typical time series data augmentation methods in time, frequency, and time-frequency domains in Fig. 2.

3  Advanced Data Augmentation Methods

3.1 Decomposition-based Methods

Decomposition-based time series augmentation has also been adopted and shown success in many time series related tasks, such as forecasting and anomaly detection. Common decomposition method like STL [Cleveland et al., 1990] or Robust-STL [Wen et al., 2019b] decomposes time series \( x_t \) as

\[
x_t = \tau_t + s_t + r_t, \quad t = 1, 2, \ldots N
\]

where \( \tau_t \) is the trend signal, \( s_t \) is the seasonal/periodic signal, and the \( r_t \) denotes the remainder signal.

In [Kegel et al., 2018], authors discussed the decomposition method to generate new time series. After STL, it recombines new time series with a deterministic component and a stochastic component. The deterministic part is reconstructed by adjusting weights for base, trend, and seasonality. The stochastic part is generated by building a composite statistical model based on residual, such as an auto-regressive model. The summed generated time series is validated by examining whether a feature-based distance to its original signal is within certain range. Meanwhile, authors in [Bergmeir et al., 2016] proposed to apply bootstrapping on the STL decomposed residuals to generate augmented signals, which are then added back with trend and seasonality to assemble a new time series. An ensemble of the forecasting models on the augmented time series has outperformed the original forecasting model consistently, demonstrating the effectiveness of decomposition-based time series augmentation approaches.

Recently, in [Gao et al., 2020], authors showed that applying time-domain and frequency-domain augmentation on the decomposed residual that is generated using robust decomposition [Wen et al., 2020; Wen et al., 2019a] can help increase the performance of anomaly detection significantly, compared with the same method without augmentation.

3.2 Statistical Generative Models

Time series augmentation approaches based on statistical generative models typically involve modelling the dynamics of the time series with statistical models. In [Cao et al., 2014], authors proposed a parsimonious statistical model, known as mixture of Gaussian trees, for modeling multi-modal minority class time series data to solve the problem of imbalanced classification, which shows advantages compared with existing oversampling approaches that do not exploit time series correlations between neighboring points. Authors in [Smyl and Kuber, 2016] use samples of parameters and forecast paths calculated by a statistical algorithm called LGT (Local and Global Trend). More recently, in [Kang et al., 2020] researchers use mixture autoregressive (MAR) models to simulate sets of time series and investigate the diversity and cover-
age of the generated time series in a time series feature space.

Essentially, these models describe the conditional distribution of time series by assuming the value at time $t$ depends on previous points. Once the initial value is perturbed, a new time series sequence could be generated following the conditional distribution.

### 3.3 Learning-based Methods

Time series data augmentation methods should be capable of not only generating diverse samples, but also mimicking the characteristics of real data. In this section, we summarize some recent learning based schemes that have such potentials.

**Embedding Space**

In [DeVries and Taylor, 2017], the data augmentation is proposed to perform in the learned embedding space (aka., latent space). It assumes that simple transforms applied to encoded inputs rather than the raw inputs would produce more plausible synthetic data due to the manifold unfolding in feature space. Note that the selection of the representation model in this framework is open and depends on the specific task and data type. When the time series data is addressed, a sequence autoencoder is selected in [DeVries and Taylor, 2017]. Specifically, the interpolation and extrapolation are applied to generate new samples. The first $k$ nearest labels in the transformed space with the same label are identified. Then for each pair of neighboring samples, a new sample is generated which is the linear combination of them. The difference of interpolation and extrapolation lies in the weight selection in sample generation. This technique is particular useful for time series classification as demonstrated in [DeVries and Taylor, 2017]. Recently, another data augmentation method in the embedding space named MODALS (Modality-agnostic Automated Data Augmentation in the Latent Space) is proposed in [Cheung and Yeung, 2021]. Instead of training an autoencoder to learn the latent space and generate additional synthetic data for training, the MODALS method train a classification model jointly with different compositions of latent space augmentations, which demonstrates superior performance for time series classification problems.

**Deep Generative Models**

Deep generative models (DGMs) have recently been shown to be able to generate near-realistic high-dimensional data objects such as images and sequences. DGMs developed for sequential data, such as audio and text, often can be extended to model time series data. Among DGMs, generative adversarial networks (GANs) are popular methods to generate synthetic samples and increase the training set effectively. Although the GAN frameworks have received significant attention in many fields, how to generate effective time series data still remains a challenging problem. In this subsection, we briefly review several recent works on GANs for time series data augmentation.

In [Esteban et al., 2017], a Recurrent GAN (RGAN) and Recurrent Conditional GAN (RCGAN) are proposed to produce realistic real-valued multi-dimensional time series data. The RGAN adopts RNN in the generator and discriminator, while the RCGAN adopts both RNNs conditioned on auxiliary information. Besides desirable performance of RGAN and RCGAN for time series data augmentation, differential privacy can be used in training the RCGAN for stricter privacy guarantees like medicine or other sensitive domains. Recently, [Yoon et al., 2019] proposed TimeGAN, a natural framework for generating realistic time series data in various domains. TimeGAN is a generative time series model, trained adversarially and jointly via a learned embedding space with both supervised and unsupervised losses. Specifically, a stepwise supervised loss is introduced to learn the stepwise conditional distributions in data. It also introduces an embedding network to provide a reversible mapping between features and latent representations to reduce the high-dimensionality of the adversarial learning space. Note that the supervised loss is minimized by jointly training both the embedding and generator networks.

**Automated Data Augmentation**

The idea of automated data augmentation is to automatically search for optimal data augmentation policies through reinforcement learning, meta learning, or evolutionary search [Ratner et al., 2017; Cubuk et al., 2019; Zhang et al., 2020; Cheung and Yeung, 2021]. The TANDA (Transformation Adversarial Networks for Data Augmentations) scheme in [Ratner et al., 2017] is designed to train a generative sequence model over specified transformation functions using reinforcement learning in a GAN-like framework to generate realistic transformed data points, which yields strong gains over common heuristic data augmentation methods for a range of applications including image recognition and natural language understanding tasks. [Cubuk et al., 2019] proposes a procedure called AutoAugment to automatically search for improved data augmentation policies in a reinforcement learning framework. It adopts a controller RNN network to predicts an augmentation policy from the search space and another network is trained to achieve convergence accuracy. Then, the accuracy is used as reward to update the RNN controller for better policies in the next iteration. The experimental results show that AutoAugment improves the accuracy of modern image classifiers significantly in a wide range of datasets.

For time series data augmentation, the MODALS [Cheung and Yeung, 2021] is designed to find the optimal composition of latent space transformations for data augmentation using evolution search strategy based on population based augmentation (PBA) [Ho et al., 2019], which demonstrates superior performance on classification problems in continuous and discrete time series data. Another recent work on automated data augmentation is proposed in [Fons et al., 2021], where two sample-adaptive automatic weighting schemes are designed specifically for time series data: one learns to weight the contribution of the augmented samples to the loss, and the other selects a subset of transformations based on the ranking of the predicted training loss. Both adaptive policies demonstrate improvement on classification problems in multiple time series datasets.

### 4 Preliminary Evaluation

In this section, we demonstrate preliminary evaluations in three common time series tasks to show the effectiveness of data augmentation for performance improvement.
4.1 Time Series Classification
In this experiment, we compare the classification performance with and without data augmentation. Specifically, we collect 5000 time series of one-week long and 5-min interval samples with binary class labels (seasonal or non-seasonal) from Alibaba Cloud monitoring system. The data is randomly split into training and test sets where training contains 80% of total samples. We train a fully convolutional network [Wang et al., 2017] to classify each time series in the training set. In our experiment, we inject different types of outliers, including spike, step, and slope, into the test set to evaluate the robustness of the trained classifier. The data augmentations methods applied include cropping, warping, and flipping. Table 1 summarizes the accuracies with and without data augmentation when different types of outliers are injected into the test set. It can be observed that data augmentation leads to 0.1% ~ 1.9% accuracy improvement.

<table>
<thead>
<tr>
<th>Outlier injection</th>
<th>w/o aug</th>
<th>w/ aug</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>spike</td>
<td>96.26%</td>
<td>96.37%</td>
<td>0.11%</td>
</tr>
<tr>
<td>step</td>
<td>93.70%</td>
<td>95.62%</td>
<td>1.92%</td>
</tr>
<tr>
<td>slope</td>
<td>95.84%</td>
<td>96.16%</td>
<td>0.32%</td>
</tr>
</tbody>
</table>

Table 1: Accuracy improvement from data augmentation under outlier injection in time series classification.

4.2 Time Series Anomaly Detection
Given the challenges of both data scarcity and data imbalance in time series anomaly detection, it is beneficial by adopting data augmentation to generate more labeled data. We briefly summarize the results in [Gao et al., 2020], where a U-Net based network is designed and evaluated on public Yahoo! dataset [Laptev et al., 2015] for time series anomaly detection. The performance comparison under different settings are summarized in Table 2, including applying the model on the raw data (U-Net-Raw), on the decomposed residuals (U-Net-DeW), and on the residuals with data augmentation (U-Net-DeWA). The applied data augmentation methods include flipping, cropping, label expansion, and APP based augmentation in frequency domain. It can be observed that the decomposition helps the increase of the F1 score and the data augmentation further boosts the performance.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>U-Net-Raw</td>
<td>0.473</td>
<td>0.351</td>
<td>0.403</td>
</tr>
<tr>
<td>U-Net-DeW</td>
<td>0.793</td>
<td>0.569</td>
<td>0.662</td>
</tr>
<tr>
<td>U-Net-DeWA (w/ aug)</td>
<td>0.859</td>
<td>0.581</td>
<td>0.693</td>
</tr>
</tbody>
</table>

Table 2: Time series anomaly detection improvement from data augmentation based on precision, recall, and F1 score.

4.3 Time Series Forecasting
In this subsection we demonstrate the practical effectiveness of data augmentation in two popular deep models DeepAR [Salinas et al., 2019] and Transformer [Vaswani et al., 2017]. In Table 3, we report the performance improvement on mean absolute scaled error (MASE) on several public datasets: electricity and traffic from UCI Learning Repository1 and 3 datasets from the M4 competition. We consider the basic augmentation methods including cropping, warping, flipping, and APP based augmentation in frequency domain. In Table 3, we summarize average MASE without augmentation, with augmentation and average relative improvement (ARI) which is computed as the mean of (MASEw/o aug − MASEw/ aug)/MASEw/o aug. We observe that the data augmentation methods bring promising results for all models in average sense. However, the negative results can still be observed for specific data/model pairs. As a future work, it motivates us to search for advanced automated data augmentation policies that stabilize the influence of data augmentation specifically for time series forecasting.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DeepAR</th>
<th>Transformer</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>w/o aug</td>
<td>w/ aug</td>
</tr>
<tr>
<td>electricity</td>
<td>0.87</td>
<td>0.97</td>
</tr>
<tr>
<td>traffic</td>
<td>0.66</td>
<td>0.80</td>
</tr>
<tr>
<td>m4-hourly</td>
<td>6.33</td>
<td>5.35</td>
</tr>
<tr>
<td>m4-daily</td>
<td>4.88</td>
<td>4.48</td>
</tr>
<tr>
<td>m4-weekly</td>
<td>12.00</td>
<td>9.34</td>
</tr>
</tbody>
</table>

Table 3: Time series forecasting improvement from data augmentation based on MASE.

5 Discussion for Future Opportunities
5.1 Augmentation in Time-Frequency Domain
As discussed in Section 2.3, so far there are only limited studies of time series data augmentation methods based on STFT in the time-frequency domain. Besides STFT, wavelet transform and its variants including continuous wavelet transform (CWT) and discrete wavelet transform (DWT), are another family of adaptive time–frequency domain analysis methods to characterize time-varying properties of time series. Compared to STFT, they can handle non-stationary time series and non-Gaussian noises more effectively and robustly. Among many wavelet transform variants, maximum over-sampling (MOSWT) is especially attractive for time series analysis [Percival and Walden, 2000; Wen et al., 2021] due to the following advantages: 1) more computationally efficiency compared to CWT; 2) ability to handle any time series length; 3) increased resolution at coarser scales compared with DWT. MOSWT based surrogate time series have been proposed in [Keylock, 2006], where wavelet iterative amplitude adjusted Fourier transform (WIAAFT) is designed by combining the iterative amplitude adjusted Fourier transform (IAAFT) scheme to each level of MODWT coefficients. In contrast to IAAFT, WIAAFT does not assume stationarity and can roughly maintain the shape of the original data in terms of the temporal evolution. Besides WIAAFT, we can also consider the perturbation of both amplitude spectrum and phase spectrum as [Gao et al., 2020] at each level of MODWT coefficients as a data augmentation scheme.

It would be an interesting future direction to investigate how to exploit different wavelet transforms (CWT, DWT, MODWT, etc.) for an effective time–frequency domain based time series data augmentation in deep neural networks.

2https://github.com/Mcompetitions/M4-methods/tree/master/ Dataset
5.2 Augmentation for Imbalanced Class

In time series classification, class imbalance occurs very frequently. One classical approach addressing imbalanced classification problem is to oversample the minority class as the synthetic minority oversampling technique (SMOTE) [Fernández et al., 2018] to artificially mitigate the imbalance. However, this oversampling strategy may change the distribution of raw data and cause overfitting. Another approach is to design cost-sensitive model by using adjust loss function [Geng and Luo, 2018]. Furthermore, [Gao et al., 2020] designed label-based weight and value-based weight in the loss function in convolution neural networks, which considers weight adjustment for class labels and the neighborhood of each sample. Thus, both class imbalance and temporal dependency are explicitly considered.

Performing data augmentation and weighting for imbalanced class together would be an interesting and effective direction. A recent study investigates this topic in the area of CV and NLP [Hu et al., 2019], which significantly improves text and image classification in low data regime and imbalanced class problems. In future, it is interesting to design deep network by jointly considering data augmentation and weighting for imbalanced class in time series data.

5.3 Augmentation Selection and Combination

Given different data augmentation methods summarized in Fig. 1, one key strategy is how to select and combine various augmentation methods together. The experiments in [Um et al., 2017] show that the combination of three basic time-domain methods (permutation, rotation, and time warping) is better than that of a single method and achieves the best performance in time series classification. Also, the results in [Rashid and Louis, 2019] demonstrate substantial performance improvement for a time series classification task when using a deep neural network by combining four data augmentation methods (i.e., jittering, scaling, rotation and time-warping). However, considering various data augmentation methods, directly combining different augmentations may result in a huge amount of data, and may not be efficient and effective for performance improvement. Recently, RandAugment [Cubuk et al., 2020] is proposed as a practical way for augmentation combination in image classification and object detection. For each random generated dataset, RandAugment is based on only two interpretable hyperparameters $N$ (number of augmentation methods to combine) and $M$ (magnitude for all augmentation methods), where each augmentation is randomly selected from $K=14$ available augmentation methods. Furthermore, this randomly combined augmentation with simple grid search can be used in the reinforcement learning based data augmentation as [Cubuk et al., 2019] for efficient space searching.

An interesting future direction is how to design effective augmentation selection and/or combination strategies suitable for time series data in deep learning. Customized reinforcement learning and meta learning optimized for time series could be potential approaches. Furthermore, algorithm efficiency is another important consideration in practice.

5.4 Augmentation with Gaussian Processes

Gaussian Processes (GPs) [Rasmussen and Williams, 2005] are well-known Bayesian non-parametric models suitable for time series analysis [Roberts et al., 2013]. From the function-space view, GPs induce a distribution over functions, i.e., a stochastic process. Time series can be viewed as functions with time as input and observation as output, and thus can be modeled with GPs. A GP $f(t) \sim \mathcal{GP}(m(t), k(t, t'))$ is characterized by a mean function $m(t)$ and a covariance kernel function $k(t, t')$. The choice of the kernel allows to place assumptions on some general properties of the modeled functions, such as smoothness, scale, periodicity and noise level. Kernels can be composed through addition and multiplication, resulting in compositional function properties, such as pseudo-periodicity, additive decomposability, and change point. GPs are often applied to interpolation and extrapolation tasks, which correspond to imputation and forecasting in time series analysis. Furthermore, deep Gaussian processes (DGP) [Damianou and Lawrence, 2013; Salimbeni and Deisenroth, 2017], which are richer models with hierarchical composition of GPs and often exceed standard (single-layer) GPs significantly in many cases, have not been well studied for time series. We believe GPs and DGP are future directions as they allow to sample time series with those properties mentioned above through the design of kernels, and to generate new data instances from existing ones by exploiting their interpolation/extrapolation abilities.

5.5 Augmentation with Deep Generative Models

Current DGMs adopted for time series data augmentation are mainly GANs. However, other DGMs also have great potentials for time series modeling. For example, deep autoregressive networks (DARNs) exhibit a natural fit for time series because they generate data in a sequential manner, obeying the causal direction of physical time series data generating process. DARNs like Wavenet [Oord et al., 2016] and Transformer [Vaswani et al., 2017] have demonstrated promising performance in time series forecasting tasks [Alexandrov et al., 2020]. Another example is normalizing flows (NFs) [Kobyzev et al., 2020], which recently have shown success in modeling time series stochastic processes with excellent inter/extrapolation performance given observed data [Deng et al., 2020]. Most recently, variational autoencoders (VAEs) based data augmentation [Fu et al., 2020] are investigated for human activity recognition.

In summary, besides the common GAN architectures, how to leverage other deep generative models like DARNs, NFs, and VAEs, which are less investigated for time series data augmentation, remain exciting future opportunities.

6 Conclusion

As deep learning models are becoming more popular on time series data, the limited labeled data calls for effective data augmentation methods. In this paper, we give a comprehensive survey on time series data augmentation methods in various tasks. We organize the reviewed methods in a taxonomy consisting of basic and advanced approaches, summarize representative methods in each category, compare them empirically in typical tasks, and highlight future research directions.
References


