The Impact of Data Augmentation on Time Series Classification Models: An In-Depth Study with Biomedical Data

Bikram De, Mykhailo Sakevych, and Vangelis Metsis[0000−0002−7371−8887]

Texas State University, San Marcos, TX 78666, USA

Abstract. Data augmentation is the practice of applying various transformations to existing data to increase their size and diversity without collecting new data. While augmentation strategies are widely recognized and implemented in image-based deep learning (DL) workflows, the degree to which they are effective in the time series domain is unclear. This paper experimentally evaluates the utility of various common time series augmentation techniques, especially those relevant to the medical sector where data limitations are prevalent. We thoroughly examine popular time series augmentation and synthetic data generation methods to evaluate their effectiveness in downstream classification tasks, encompassing both traditional and DL-based approaches. This research aims to offer insights into the applicability and efficacy of data augmentation strategies in improving model generalization and mitigating data scarcity challenges, with a focus on biomedical time-series data.

Keywords: Time Series · Data Augmentation · Classification

1 Introduction

Data augmentation is a technique for enhancing the size and diversity of training datasets in machine learning. It involves creating modified versions of existing data or synthesizing new data samples based on the statistical properties of the original dataset. This method is instrumental across various data types, including images, audio, video, and text. Our focus, however, narrows down to the application of data augmentation techniques on time series data, a domain that presents unique challenges and opportunities, especially within the medical field.

The infusion of augmented data into the training process of time series models offers significant advantages. It aids in the development of robust, flexible models capable of generalizing effectively to new, unseen data. By introducing a variety of scenarios and patterns through augmentation, models can better learn the complex, non-linear relationships and temporal dependencies that characterize time series data. This is particularly crucial in mitigating the risks of overfitting and enhancing the performance of models trained on limited, imbalanced, or

[0] Project source code: https://github.com/imics-lab/time-series-augmentation

noisy datasets. Moreover, by limiting augmentation to the training phase, the integrity and authenticity of the data during inference are preserved, ensuring the models' applicability to real-world scenarios remains uncompromised.

However, these advantages come with their own set of limitations. Despite not directly affecting the inference phase, the process of data augmentation must navigate the complexities of medical data's sensitivity, specificity, and multidimensional nature. The creation of augmented data requires careful consideration to avoid introducing biases or artifacts that could mislead the learning process or obscure critical information. Some augmentation techniques that directly apply to other domains, for example, image flipping or rotating, may be ineffective or even detrimental for time series data because they distort the inherent temporal dependencies. Additionally, ethical and privacy concerns are paramount, as the augmentation process involves manipulating sensitive patient data, necessitating stringent adherence to data protection and privacy regulations. In conclusion, while augmenting time series training sets offers a pathway to developing more capable and generalizable models, it necessitates a careful, ethically mindful approach.

In this paper, we present a survey of existing time series augmentation techniques and their effectiveness on different types of time series data. We categorize augmentation techniques into two parts: (1) traditional methods and (2) generative methods (Fig. 1). Traditional methods involve simple signal transformations such as jittering, scaling, magnitude warping, time warping, and window slicing. Augmented copies of the original training samples are added to the training set. Deep learning-based generative methods introduce new data samples by first modeling the statistical properties of the dataset and then generating new data that obey these statistical properties but are not identical to any of the original samples on which the models were trained. The three most popular categories of generative methods at the time of writing are Generative Adversarial Networks (GANs) [7], Variational Autoencoders (VAEs) [1], and Diffusion models [8].

We assess the impact of these augmentation techniques on four distinct time series datasets – pertaining to human activity recognition, sleep studies, heart disorder recognition, and epileptic seizure detection (see section 3.1 for more details) – and use the original and augmented versions of the data for downstream classification tasks with three popular deep learning time series classification architectures, namely LSTM [5], CNN [3], Transformer [14].

Our experimental findings underscore the nuanced effects of different augmentation strategies on model accuracy, influenced by the specific characteristics of the data and the architecture of the classifiers. These insights highlight the absence of a universally optimal augmentation approach, advocating for a tailored selection of techniques based on the specific requirements of each task. A detailed discussion of our empirical observations and their implications for machine learning practice in time series analysis will be elaborated in subsequent sections of this paper.

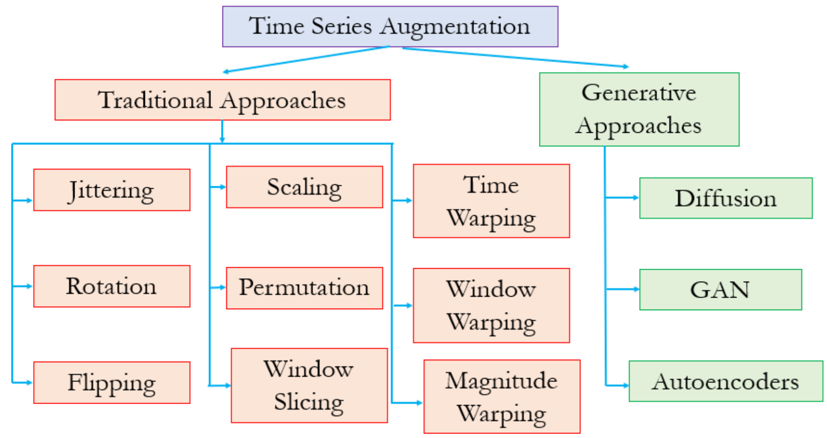

Fig. 1. Overview of various Time series augmentation techniques.

2 Background

The exploration of time series data augmentation techniques has evolved significantly, with various methodologies being developed to enhance the robustness and performance of machine learning models. This body of work encompasses a range of strategies aimed at enriching training datasets, thereby improving model generalization across diverse applications such as classification, forecasting, and anomaly detection. Previous studies, such as those by Wen et al. [12] and Iglesias et al. [2], have provided comprehensive overviews of augmentation methods, discussing their applications, the metrics for evaluation, and the challenges encountered with each technique. Despite these efforts, a gap remains in directly comparing the effects of these augmentation methods across different types of datasets, particularly those related to human activity and medical diagnosis.

In this context, our paper endeavors to bridge this gap by offering a detailed experimental comparison of traditional and deep learning-based generative augmentation techniques. Figure 1 provides an overview of the augmentation techniques compared in this work. We assess their impact on datasets pertinent to human activity recognition and medical diagnosis, employing various model architectures to evaluate the effectiveness of each augmentation method.

2.1 Augmentation Techniques Overview

In this section, we delve into several common augmentation techniques examined in this work and provide a brief formal definition of each.

Rotation: Rotation augmentation involves applying a transformation matrix to the original time series data to generate new samples. This method is mathematically represented as: $X_{\text{rotated}} = R(\theta)X$ where X is the original data, $R(\theta)$ is the rotation matrix defined by the rotation angle θ, and X_{rotated} is the rotated data.

Jittering: Jittering introduces small, random variations to the data, effectively modeled as: $X_{\text{jittered}} = X + \mathcal{N}(0, \sigma^2)$ where X is the original data and $\mathcal{N}(0, \sigma^2)$ represents Gaussian noise with mean 0 and variance σ^2.

Flipping: Flipping reverses the time series data, mathematically described by: $X_{\text{flipped}}[t] = X[N - t]$ where X is the original series, N is the length of the series, and t is the time step.

Scaling: Scaling adjusts the amplitude of the data either by magnifying or by shrinking the data point range of values. $X_{\text{scaled}} = \text{multiplier} \times X_{\text{original}}$ where X is the original data and *multiplier* is a scaling factor which can be either greater or less than 1.

Permutation: Permutation reorders the data points randomly: $X_{\text{permuted}} = X[\pi(i)]$ where X is the original data and π represents a permutation of the indices i.

Window Slicing: Window slicing segments the data into windows, formally: $X_{\text{slice}} = X[t : t + w]$ where X is the original series, t is the starting point, and w is the window size.

Time Warping: Time warping alters the temporal scale: $X_{\text{warped}}(t) = X(\lambda t)$ where X is the original series and λ is the warping factor.

Window Warping: Window warping applies localized transformations: $X_{\text{window warped}} = \text{Transform}(X_{\text{window}})$ where X_{window} is a segment of the original series and Transform denotes the applied warping.

Magnitude Warping: Magnitude warping modifies the amplitude: $X_{\text{magnitude warped}} = X \cdot \lambda$ where X is the original series and λ is the warping factor.

Fourier Transform: Fourier Transform augmentation modifies the frequency components: $\mathcal{F}(X_{\text{augmented}}) = \mathcal{F}(X) + \Delta\mathcal{F}$ where $\mathcal{F}(X)$ is the Fourier transform of the original data, and $\Delta\mathcal{F}$ represents the modifications in the frequency domain.

Generative Adversarial Networks (GANs): GANs generate synthetic data by training a generator G to produce data that a discriminator D cannot distinguish from real data, represented as: $G(z) \approx X$ where z is random noise input and X is the real time series data.

Variational Autoencoders (VAEs): VAEs generate synthetic data by encoding input data X into a latent space z and then decoding it, shown as: $X_{\text{synthetic}} = \text{Decoder}(\text{Encoder}(X))$

Diffusion Models: Diffusion models represent a class of generative models that gradually transform data from a simple distribution (e.g., Gaussian noise) into complex data distributions by learning to reverse a diffusion process, $X_{t-1} = f(X_t, \theta)$, where X_t represents the data at step t, X_{t-1} is the data at the previous step, and f is a learned function parameterized by θ. In the context of time series data augmentation, diffusion models can be employed to generate synthetic time series data that captures the intricate temporal dynamics and distributions of the original dataset.

3 Methodology

Our methodology encompasses a rigorous approach to evaluating the efficacy of various data augmentation techniques applied to time series datasets, ensuring the integrity of our experimental setup and the reliability of our results. To address the challenges inherent in time series data analysis, particularly the risk of data leakage, we meticulously partition the datasets based on subjects. This strategy guarantees that each subject is exclusively included in either the training or testing set, thereby preserving the independence of our test data and ensuring it remains unseen during training.

In our analysis, we explore both traditional and deep learning-based augmentation methods. Traditional techniques such as jittering, scaling, magnitude warping, time warping, window warping, and window slicing are systematically evaluated. Each technique is parameterized to quantify the extent of augmentation, with experiments conducted across a spectrum of parameters using 10-fold cross-validation. This process allows us to identify the parameter setting that maximizes mean accuracy for each augmentation method. Using the selected parameter, we augment the training data doubling the size of each class. Thus the entire training data is doubled and the class ratios remain unaltered. Using the best augmentation hyperparameter, we again shuffle the data 10 times and calculate the accuracy for each shuffle. Using the 10 accuracies, we calculate the technique's average accuracy and 95% confidence intervals.

For the task of classification, we leverage three distinct classifier models: LSTM, CNN, and Transformer-based models, utilizing the TSAI library [11] for state-of-the-art implementations. Specifically, we employ the LSTM-FCN architecture [5] for the LSTM model, the Inception Time model [3] for the CNN, and the TST architecture [14] for the Transformer model, as implemented in the TSAI library [11].

Data preprocessing forms the initial phase of our methodology, where data from four distinct datasets are prepared for analysis. This involves loading data from various channels, processing it through data loaders, and splitting it into training, testing, and validation sets based on subjects. This subject-based splitting is critical for avoiding data leakage in time series analysis.

Our augmentation pipeline is depicted in Figure 2 for traditional methods. In contrast, the pipeline for deep learning-based methods differs in the final step by eliminating the need for multiple shuffling iterations, instead requiring only a single iteration of data generation. We apply six traditional augmentation methods, sourcing implementations [4]. For deep learning-based augmentation, we investigate three techniques: a Transformer-based GAN [7], a Variational Autoencoder [1], and a Diffusion model [8], each implemented from recent literature.

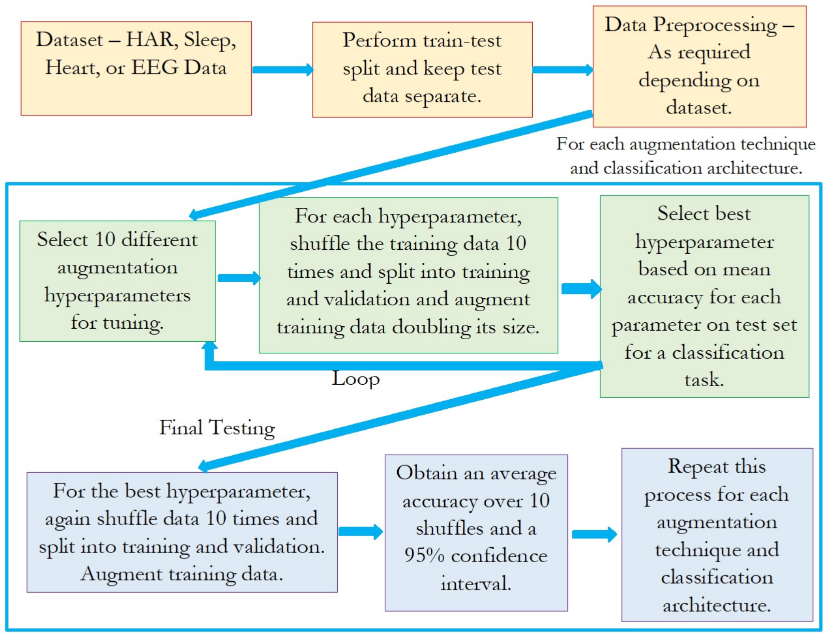

Fig. 2. Overview of augmentation and testing pipeline for traditional time series augmentation techniques.

Through this comprehensive methodology, we aim to provide a detailed comparison of the impact of various augmentation techniques on time series datasets, focusing on human activity recognition and medical diagnosis. Our approach ensures a robust evaluation framework, leveraging advanced classification models to assess the effectiveness of each augmentation technique in enhancing dataset quality and model performance.

3.1 Dataset Description

The four datasets described below were selected as representative of typical biomedical applications using machine learning models.

Human Activity Recognition: The UniMiB SHAR [9], is a dataset of acceleration samples acquired with an Android smartphone designed for human activity recognition and fall detection. The dataset includes 11,771 samples of both human activities and falls performed by 30 subjects of ages ranging from 18 to 60 years. The dataset contains 9 types of daily living activities. The 9 types of daily living activities include: Standing Up From Sitting, Standing Up From Laying, Walking, Running, Going Upstairs, Jumping, Going Downstairs, Lying Down From Sitting, Sitting Down.

Sleep Event Detection: The Polysomnography (PSG) dataset [6] used in this work contains data recorded on 212 individuals in a hospital setting for sleep apnea syndrome (SAS) diagnosis. Five categories of abnormal events were annotated by a medical team ("respiratory", "neurological", "limb activity related", "nasal", and "cardiac"). In this study, we detect only the respiratory events, thus forming a binary classification task. We use only the 12 signal channels that are most relevant to the respiratory events. Nine of the channels (3 x EEG, 2 x EMG, 2 x EOG, 2 leg sensors, and ECG) were downsampled from 200 Hz to 100 Hz to match the remaining three sensors used (flow thermistor plus thoracic and abdominal respiratory belts).

Heart Disorder Detection: The MIT-BIH disorder dataset [10] contains 48 snippets of ambulatory ECG recordings spanning half an hour each from 47 subjects across five heart conditions. The samples, originally recorded at 125Hz, have been adjusted to 187 in length for U-Net compatibility. The training set has 87554 samples, with the majority class having 72471 samples and the smallest class having 641. The test set includes 21892 samples, ranging from 162 to 18118 samples per class. The majority class of both the training and testing set was reduced to 10% to prevent class imbalance.

Epileptic Seizure Recognition: This dataset [13] consists of 5 different folders, each with 100 files, with each file representing a single subject/person. Each file is a recording of brain activity for 23.6 seconds. The corresponding time-series is sampled into 4097 data points. Each data point is the value of the EEG recording at a different point in time. So we have a total of 500 individuals, with each having 4097 data points for 23.5 seconds. The five different folders represent five different situations in which the EEG signal is recorded from the brain. The folders include eyes open, and eyes closed, recordings from healthy brain areas with a tumor in the brain, recordings from the part of the brain with the tumor, and the last folder, recordings of seizure activity. A binary classification is performed with this data for recording of seizures against others.

4 Results

In Figure 3, we show the downstream classification of each combination of augmentation technique and classification architecture on the four datasets used in this study. Each of the four plots shown in the figure corresponds to a different dataset, as indicated by the label on the top of the plot. Each plot is a bar chart, with the bars separated into three groups corresponding to the three deep-learning classification architectures. Each bar within a group corresponds to a different augmentation technique, showing the mean accuracy accomplished when applying that augmentation technique to data. Along with the accuracy, a 95% confidence interval range is shown at the top of each bar. The order is maintained across groups and plots for comparison consistency.

Due to space limitations, only the plots are shown here. For the detailed numeric results in tabular format and specific parameter values used by each augmentation technique, the reader should refer to the Appendix of this paper.

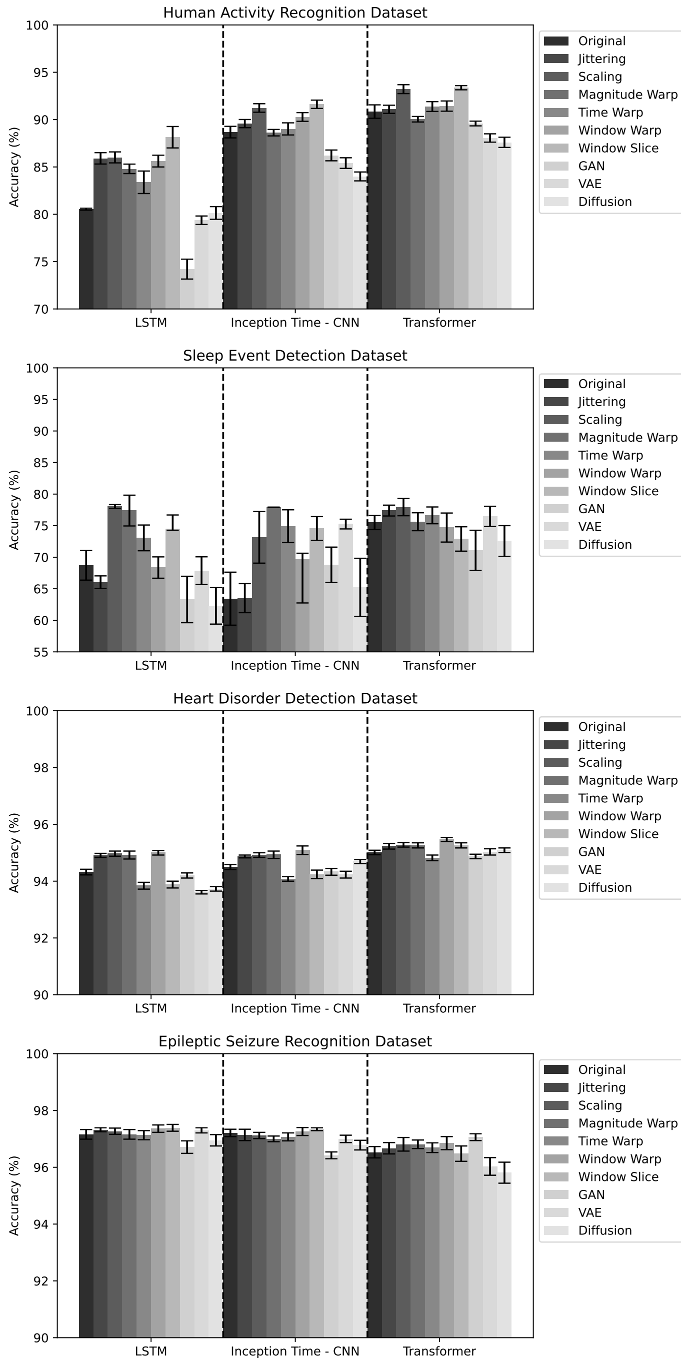

Fig. 3. Classification accuracy results for the combination of four datasets, three classifier architectures, and ten augmentation techniques explored in this study.

5 Discussion

Our investigation into data augmentation's impact on time series classification accuracy has elucidated several crucial insights. Notably, while augmentation typically boosts classification accuracy, the effectiveness of specific techniques varies depending on the dataset and classification architecture used.

Traditional augmentation methods, such as jittering and scaling, generally enhance model performance across various datasets by introducing necessary variability without significantly distorting the time series' inherent dynamics. However, the mixed results observed with window warping and slicing highlight the context-sensitive nature of augmentation effectiveness, indicating that a tailored approach, possibly involving a combination of techniques, might yield the best results.

The effectiveness of generative deep learning-based augmentation methods also varies. It appears that the addition of synthetic examples with class ratio distribution equal to the original dataset does not significantly boost the overall accuracy. However, the boost in performance may be more pronounced when the class distribution is imbalanced and synthetic examples are introduced to the minority class(es) to mitigate the class imbalance.

Furthermore, our findings reveal that multi-channel datasets tend to benefit more from augmentation than single-channel datasets, likely due to the richer information content that provides more scope for effective augmentation without loss of signal integrity. Conversely, the application of augmentation techniques, especially in datasets with low signal-to-noise ratios like EEG data, requires careful consideration to avoid degrading the classification accuracy.

Interestingly, the impact of augmentation appears to be relatively consistent across different classification architectures, indicating that the benefits of data augmentation transcend architectural differences and largely depend on the quality and diversity of the training data.

In summary, data augmentation emerges as a valuable tool for improving time series classification models, with its effectiveness highly contingent on the dataset characteristics, augmentation technique, and classification architecture. A judicious, context-aware application of augmentation techniques is essential to optimize model performance, highlighting the need for ongoing research to refine these strategies for diverse applications.

6 Conclusion and Future Work

Our comprehensive exploration of data augmentation strategies for time series classification in the biomedical domain has illuminated their varied impacts on model performance. We have shown that the effectiveness of augmentation techniques is highly context-specific, with no one-size-fits-all solution. This underscores the necessity for a tailored approach, informed by the dataset's characteristics and the model's requirements.

Moving forward, the development of more sophisticated, adaptive augmentation methods that can autonomously determine the most effective strategies for a given dataset and task is an exciting area for further exploration. While this study has focused on biomedical time series data, the insights gained are broadly applicable across various domains, pointing towards the broader goal of improving model robustness and generalization through strategic data augmentation.

References

1. Barak, S., Mirafzali, E., Joshaghani, M.: Improving deep learning forecast using variational autoencoders. Available at SSRN 4009937 (2022)
2. Iglesias, G., Talavera, E., González-Prieto, Á., Mozo, A., Gómez-Canaval, S.: Data augmentation techniques in time series domain: a survey and taxonomy. Neural Computing and Applications **35**(14), 10123–10145 (2023)
3. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data Mining and Knowledge Discovery **34**(6), 1936–1962 (2020)
4. Iwana, B.K., Uchida, S.: An empirical survey of data augmentation for time series classification with neural networks. PLOS ONE **16**(7), e0254841 (jul 2021)
5. Karim, F., Majumdar, S., Darabi, H., Chen, S.: Lstm fully convolutional networks for time series classification. IEEE access **6**, 1662–1669 (2017)
6. Korompili, G., Amfilochiou, A., Kokkalas, L., Mitilineos, S.A., Tatlas, N.A., Kouvaras, M., Kastanakis, E., Maniou, C., Potirakis, S.M.: Psg-audio, a scored polysomnography dataset with simultaneous audio recordings for sleep apnea studies. Scientific data **8**(1), 197 (2021)
7. Li, X., Metsis, V., Wang, H., Ngu, A.H.H.: Tts-gan: A transformer-based time-series generative adversarial network. In: International Conference on Artificial Intelligence in Medicine. pp. 133–143. Springer (2022)
8. Li, X., Sakevych, M., Atkinson, G., Metsis, V.: Biodiffusion: A versatile diffusion model for biomedical signal synthesis. arXiv preprint arXiv:2401.10282 (2024)
9. Micucci, D., Mobilio, M., Napoletano, P.: Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. Applied Sciences **7**(10), 1101 (2017)
10. Moody, G.B., Mark, R.G.: The impact of the mit-bih arrhythmia database. IEEE Engineering in Medicine and Biology Magazine **20**(3), 45–50 (2001)
11. Oguiza, I.: tsai - a state-of-the-art deep learning library for time series and sequential data. Github (2023), https://github.com/timeseriesAI/tsai
12. Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., Xu, H.: Time series data augmentation for deep learning: A survey. arXiv preprint arXiv:2002.12478 (2020)
13. Wu, Q., Fokoue, E.: Epileptic Seizure Recognition. UCI Machine Learning Repository (2017), DOI: https://doi.org/10.24432/C5G308
14. Zerveas, G., Jayaraman, S., Patel, D., Bhamidipaty, A., Eickhoff, C.: A transformer-based framework for multivariate time series representation learning. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. pp. 2114–2124 (2021)

Appendix

The tables below show the detailed classification accuracy results for each dataset, augmentation technique, and classification algorithm architecture. The LSTM, CNN, and Transformer table headers correspond to the LSTM-FCN [5], the Inception-Time CNN [3], and the TST [14] architectures respectively, as implemented in the TSAI library [11]. The "Par." table header indicates the tuned hyperparameter value used for that particular augmentation technique in the corresponding experiment. For example, a jittering value of 0.03 corresponds to the standard deviation value of the added Gaussian noise, a scaling value of 0.7 corresponds to the scaling factor multiplier, etc. For the generative models, the default hyperparameters recommended by the original model authors were used without tuning.

More implementation details can be found in the public source code page of the project: https://github.com/imics-lab/time-series-augmentation

Table 1. Results for Human Activity Recognition dataset.

Augmentation	**Par.**	**LSTM**	**Par.**	**CNN**	**Par.**	**Transformer**
Original	Null	80.54% ± 0.1	Null	88.68% ± 0.61	Null	90.85% ± 0.71
Jittering	0.03	85.91% ± 0.6	0.05	89.58% ± 0.43	0.8	91.09% ± 0.43
Scaling	0.7	86.01% ± 0.58	3	91.23% ± 0.45	3	93.22% ± 0.47
Mang. Warp.	0.1	84.8% ± 0.5	0.1	88.62% ± 0.34	0.3	90.04% ± 0.29
Time Warp.	0.1	83.38% ± 1.19	0.1	89.02% ± 0.64	0.1	91.38% ± 0.52
Window Warp.	0.01	85.62% ± 0.62	0.9	90.28% ± 0.46	0.01	91.43% ± 0.54
Window Slic.	0.9	88.14% ± 1.13	0.9	91.62% ± 0.44	0.9	93.37% ± 0.22
TTS GAN	Null	74.21% ± 1.05	Null	86.22% ± 0.57	Null	89.59% ± 0.25
VAE	Null	79.37% ± 0.45	Null	85.41% ± 0.56	Null	88.06% ± 0.44
Diffusion	Null	80.14% ± 0.67	Null	84% ± 0.47	Null	87.6% ± 0.54

Table 2. Results for Sleep Event Detection dataset.

Augmentation	**Par.**	**LSTM**	**Par.**	**CNN**	**Par.**	**Transformer**
Original	Null	68.72% ± 2.35	Null	63.43% ± 4.2	Null	75.49% ± 1.12
Jittering	0.01	66.04% ± 1	0.03	63.51% ± 2.3	0.07	77.39% ± 0.86
Scaling	11	78.04% ± 0.29	11	73.15% ± 4.09	9	77.94% ± 1.37
Magnitude Warping	3	77.39% ± 2.44	11	77.92% ± 0.02	9	75.62% ± 1.42
Time Warping	3	73.06% ± 2.04	11	74.91% ± 2.6	0.1	76.63% ± 1.33
Window Warping	0.9	68.36% ± 1.69	0.7	69.69% ± 3.94	0.09	74.7% ± 2.3
Window Slicing	0.2	75.48% ± 1.21	0.1	74.55% ±1.88	0.7	72.89% ± 1.93
TTS GAN	Null	63.3% ± 3.68	Null	68.8% ± 2.8	Null	71.09% ± 3.18
VAE	Null	67.87% ± 2.19	Null	75.25% ± 0.77	Null	76.47% ± 1.6
Diffusion	Null	62.29% ± 2.89	Null	65.23% ± 4.6	Null	72.57% ± 2.43

Table 3. Results for Heart Disorder Detection dataset.

Augmentation	**Par.**	**LSTM**	**Par.**	**CNN**	**Par.**	**Transformer**
Original	Null	94.32% ± 0.1	Null	94.5 % ± 0.09	Null	95.01% ± 0.08
Jittering	0.01	94.91% ± 0.07	0.01	94.87% ± 0.05	0.03	95.23% ± 0.1
Scaling	0.1	94.97% ± 0.09	0.1	94.92% ± 0.08	0.1	95.28% ± 0.08
Mang. Warp.	0.1	94.92% ± 0.14	0.1	94.93% ± 0.13	0.1	95.26% ± 0.09
Time Warp.	0.1	93.84% ± 0.12	0.1	94.08% ± 0.08	0.1	94.82% ± 0.1
Window Warp.	0.01	95 % ± 0.08	0.03	95.09% ± 0.15	0.03	95.47% ± 0.07
Window Slic.	0.01	93.88 % ± 0.12	0.01	94.24% ± 0.15	0.01	95.26% ± 0.09
TTS GAN	Null	94.2% ± 0.09	Null	94.33% ± 0.12	Null	94.87% ± 0.08
VAE	Null	93.61% ± 0.06	Null	94.23% ± 0.12	Null	95.03% ± 0.11
Diffusion	Null	93.73% ± 0.08	Null	94.69% ± 0.07	Null	95.09% ± 0.08

Table 4. Results for Epileptic Seizure Detection dataset.

Augmentation	**Par.**	**LSTM**	**Par.**	**CNN**	**Par.**	**Transformer**
Original	Null	97.16% ± 0.17	Null	97.21 % ± 0.13	Null	96.53% ± 0.2
Jittering	0.2	97.32% ± 0.07	0.4	97.14% ± 0.2	1	96.67% ± 0.2
Scaling	0.1	97.27% ± 0.11	0.1	97.12% ± 0.11	0.1	96.81% ± 0.24
Magnitude Warping	0.1	97.16% ± 0.17	0.1	97% ± 0.1	0.1	96.81% ± 0.15
Time Warping	7	97.13% ± 0.16	3	97.07% ± 0.14	0.1	96.69% ± 0.17
Window Warping	0.07	97.36 % ± 0.13	0.7	97.26% ± 0.14	0.1	96.85% ± 0.23
Window Slicing	0.3	97.39 % ± 0.12	0.3	97.34% ± 0.05	0.9	96.48% ± 0.27
TTS GAN	Null	96.71% ± 0.22	Null	96.42% ± 0.12	Null	97.06% ± 0.12
VAE	Null	97.3% ± 0.09	Null	97% ± 0.13	Null	96.03% ± 0.31
Diffusion	Null	96.95% ± 0.2	Null	96.78% ± 0.17	Null	95.81% ± 0.37