Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 175 (2020) 64–71

Procedia Computer Science

www.elsevier.com/locate/procedia

The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC)
August 9-12, 2020, Leuven, Belgium

Time-Series Data Augmentation based on Interpolation

Cheolhwan Oh, Seungmin Han, Jongpil Jeong*

Department of Smart Factory Convergence, Sungkyunkwan University, Suwon, Jangan-gu, Gyeonggi-do, 16419, Republic of Korea

Abstract

In machine learning, data augmentation is the process of generating synthetic data samples that will be used to train the model to improve the performance of the machine learning model. Data augmentation has been shown to improve the generalization capabilities of models and is particularly popular in the field of computer vision, which is to deal with image data. In contrast, data augmentation is less widely used in the field of time-series data, such as time-series classification, than in computer vision. This is because time-series data is particularly vulnerable to the transformation of data that occurs while performing data augmentation. For example, flipping or rotating images to augment image data does not significantly undermine the meaning of the original data. However, in the case of time-series data, it is likely to distort the meaning of the data, and it is difficult to identify whether or not it has altered. Previously proposed time-series data augmentation methods performed well in many fields, but often did not consider trend information of time-series data such as slicing or reordering the original time-series. In this paper, we propose a time-series data augmentation method based on interpolation. The proposed method is robust against the impairment of trend information of the original time-series and has the advantage of not high complexity. To evaluate the performance of the proposed method, we experimented with time-series datasets from the UCR archive and showed that the performance of the model could be improved.

Keywords: Time-Series Classification; Data Augmentation; Interpolation; Machine Learning; Deep Learning

1. Introduction

Time-series data is growing rapidly and exists everywhere. Time-series data are constantly occurring in a wide range of areas such as climate, robotics, entertainment, finance, healthcare and transportation. The development of IOT sensors and communication technology has made it possible to store and use large amounts of time-series data. As a result, there has been an increasing number of attempts to extract latent patterns from large volumes of data to create new values. The importance and impact of time-series analysis and modeling techniques continue to increase, and machine learning is a representative example.

* Corresponding author. Tel: +082-031-290-4260; fax: +082-031-290-4260
E-mail address: jpjeong@skku.edu

10.1016/j.procs.2020.07.012

The general focus of machine learning is the representation of input data and generalization of learned patterns for use in future invisible data. The excellence of the data representation greatly affects the performance of the machine learning model on the data. That is, if the data representation is not good, even the advanced model may degrade the performance. If the data representation is good, the relatively simple model may have high performance. Therefore, feature engineering, which focuses on constructing shapes and data representations from raw data, is an important component of machine learning [1]. However, feature engineering consumes a lot of effort in machine learning tasks, and is generally quite domain specific and involves significant human input. For example, when using image data as input to machine learning models, feature extraction was performed using methodologies such as Histogram of Oriented Gradients (HOG) [2], Scale Invariant Feature Transform (SIFT) [3], Speeded Up Robust Features (SURF) [5] and Oriented FAST and Rotated BRIEF (ORB) [4]. When the data used as the input of the machine learning model is a time-series, the raw vector is used as the input of the model as it is, or when the high dimension of the time-series data is burdened, the shape features or statistical features are extracted and used [6]. In addition, nondata adaptive, data adaptive, and model based methodologies are used to obtain representations of time-series data [7]. Deep learning, on the other hand, is a methodology in which the model itself extracts features from raw data, that is, performs automatic extraction of features. Deep learning algorithms that mimic the way the human brain works have a hierarchical architecture that builds up layers. And stacked layers automatically extract features from raw data [8]. The eyes and ears are the main inputs of the biological brain. And like the human brain, deep learning mimics the brain's excellent performance in image processing and speech recognition even when it receives raw images and sounds [10, 9]. For the reasons described above, deep learning has been widely used in many areas, such as computer vision and signal processing. Among them, deep learning architectures such as CNN and RNN have been shown to be effective in many fields, and since these architectures consist of many parameters, an efficient and large data set is required for training these parameters [11]. However, for many applications, only small labeled data sets can be used to train deep learning algorithms, which often degrades the performance of the model [12].

Insufficient training sets are not enough to learn many of the parameters of deep learning algorithms and are known to cause overfitting [13]. In order to solve this problem, more data needs to be collected, but in actual applications, additional data collection is often difficult for various reasons such as time and cost limitations. Data augmentation can be used to address this. This method provides the opportunity to generate variations of the training sample without changing the semantics of the raw data, and in practice it is known to be effective in solving overfitting [14, 15]. The main challenge of data augmentation is whether the newly created data maintains the correct labels. For this reason, it is particularly widely used in the field of image data because of the advantage that the meaning does not change significantly even if the data is reversed or rotated. In addition, it is very useful in that it is possible to verify whether the meaning of the data has been changed by directly identifying the image change of the generated data with the human eye. For the foregoing reasons, there are so many ways to augment image data [15]. And it is important to note that these various methods can all be used generically for most image data. On the other hand, in case of time-series data, there are various difficulties in performing data augmentation. It does not have any of the advantages found in the image data augmentation described above, and for this reason, the methods used in image data augmentation are not well generalized to time-series data [12]. For this reason, when data augmentation is necessary due to insufficient number of time-series data, the researcher should develop and use a new data augmentation method that works well in his application. In fact, there were researchers who proposed augmentation method for time-series data generated in various fields, and they proposed a method for augmentation of time-series data generated in fields such as speech recognition, electroencephalogram (EEG), and wearable sensor and showed good performance [16, 17, 18, 19, 20, 21, 22, 23].

In this paper, we propose a time-series data augmentation method based on interpolation. The proposed method aims to:

- Robustness that does not compromise the original meaning of time-series data.
- Generality applicable to most time-series data.
- Convenience that is not complicated to use

The rest of this paper is organized as follows. Section 2 discusses related work and explains the background of the proposed method. Section 3 describes the proposed time-series data augmentation method. Section 4 conducts experiments to evaluate the performance of the proposed approach and describes the experiment result. Section 5 discusses conclusions and future work.

2. Related Work

Unlike image data, time-series data is vulnerable to label changes due to the deformation of original data that occurs when data augmentation is applied, and it is hard to use due to the difficulty of identifying them. To solve this problem, previous researchers have proposed a data augmentation methodology that is well applied to time-series data. This section introduces the proposed data augmentation methodologies and introduces the motivations for proposed method.

2.1. Domain-Specific Method

Researchers in the field of speech recognition have proposed data enhancement techniques in consideration of features such as human vocal organs and human speech speed and tempo changes. Taking advantage of the fact that the length of vocal tract, one of the vocal organs, influences the change in voice, some researchers have proposed a method for data enhancement based on Vocal Tract Length Perturbation (VTLP). They also proposed a method for augmenting data using Stochastic Feature Mapping (SFM), which is performed by probabilistic mapping of the effect of changing one speaker to another [16, 17]. Some researchers have proposed data augmentation techniques using information such as human speech speed and tempo to slowly or rapidly transform them [28, 18].

In addition to voice and audio, many researchers have presented time-series data augmentation methods that are well suited to their applications.

In the EEG field, Fabien Lotte [19] generated new data by changing the order of transformed data through Fourier transform or extracting signal power through principal component analysis in order to enhance the data. Krell et al. [22] proposed a method for augmenting data using temporal and spatial / rotational distortions for rare stimuli and movement prediction applications.

Steven et al. [23] presented a data augmentation workflow that extracts features from data collected from wearable sensors and performs local averaging in the extracted feature space. Um et al. [20] proposed a large number of data augmentation methods to generate new data from data collected from wearable sensors worn by patients with Parkinson's Disease. The methods are as follows.

- Jittering : A way of simulating additive sensor noise.
- Scaling : Changes the magnitude of the data in a window by multiplying by a random scalar.
- Rotations : Upside-down placement of the sensor. This changes the position of the sensor you are wearing.
- Permutation : Slice the data into N same length segments, and randomly permute the segments to create a new window.
- Time-warping : Smoothly distorting the time intervals between samples to change the temporal locations of the samples.
- Magnitude-warping : changes the magnitude of each sample by convolving the data window with a smooth curve varying around one.
- Cropping : Cut and delete collected data after a certain time.

2.2. Time Distortion Based Method

Not all of them, but the methods customized for the specific applications mentioned above are generally hard to use universally. For example, it is difficult to apply the VTLP-based augmentation method used in the speech recognition field to time-series data in other financial fields.Similarly, among the wearable sensor data augmentation methods, scaling, time-warping, etc. can be applied to other applications, but the rotation methods cannot be used in other fields.

Of course, there are not only methods customized for the specific applications mentioned above, but also commonly used data augmentation methods for time-series data. The data augmentation method commonly used in time-series is the time slicing window method [12]. This method was introduced to train Deep CNN's numerous parameters [29], and the authors propose window slicing for the data augmentation. For a time-series T={$t_1, t_2, \cdots, t_n$}, a slice is a snippet of the original time-series, defined as $S_{i:j}=\left\{t_i, t_{i+1}, \cdots, t_j\right\}$, $1 \leq i \leq j \leq n$. Suppose a given time-series T is of length n, and the length of the slice is s, our slicing operation will generate a set of $n - s + 1$ sliced time-series:

$$Slicing(T, s) = \{S_{1:s}, S_{2:s+1}, \cdots, S_{n-s+1:n}\} \tag{1}$$

where all the time-series in $Slicing(T, s)$ have the same label as their original time-series T does. the authors apply window slicing on all time-series in a given training dataset.

Starting with time slicing, the methods for transforming and distorting time information to generate new data are the mainstays of the data augmentation method for time-series classification. To augmentation the time-series data, Le Guennec et al. [11] proposed a window warping method that stretches or reduces part of the window of a time-series. The authors applied it to CNN and showed that the performance of the model is improved.

In fact, how to generate new time-series by distorting time information is also already described in section 2.1. The method described in section 2.1 is also a way of rearranging the original time-series in a new order. The method of converting a person's speaking speed or tempo is also a method of slowing down or speeding up time. The only difference is whether or not additional processes are included that can reflect the nature of the domain in order to suggest augmentation that works well for that domain.

2.3. Manifold Based Method

The time warping-based method is simple to use, effective for model improvement, and can be useful in many applications. However, in the case of distorting the time information of the original time-series, in particular, a representative time distortion method, slicing and rearrangement, does not consider trend, cyclical, and seasonal information, which are major components of the time-series. This may cause the label of the original data to lose its meaning. This means that some data augmentation methods do not preserve the manifold of the time-series. Time-series analysis has been studied for a long time by many researchers in wide domains. In the traditional research field, time-series are largely divided into four components: trend, cyclical, seasonal, and error. An important purpose of time-series analysis is to show how each component affects the Y value over time, including future predictions and the separation of each of the three components [24].

The time distortion that occurs during the augmentation process can also damage the important manifolds contained by the components of the time-series. Concerned about this, previous researchers have proposed ways to preserve the manifold.

Forestier et al. [25] proposed a data augmentation method based on DTW Barycenter Averaging (DBA) [26], an averaging method using dynamic time warping. They calculate a weighted average that reflects the manifold of the original data and use it as new data. The authors note that their contribution lies in the proposed weighted averaging technique that well reflects the manifold of the original data. DeVries et al. [21] trains the manifold of the original time-series through the seq2seq autoencoder, the authors augment the data by extracting the features of each sample in the trained model and transforming them in feature space. Ramponi et al. [12] noted that the process of slicing time-series tends to remove temporal correlations from the data, and was concerned that the data generated in this way could have very different interpretations. To remedy their concerns, they trained the manifold of the original time-series data using the conditional GAN architecture [27] and generated new time-series data from the trained model. However, these methods are cumbersome to train another model to augmentation new data. Also, additional work is needed to ensure the reliability of the generative model, and in some cases, the computing power consumed can be a burden.

3. Proposed Method : Time-Series Data Augmentation based on Interpolation

Interpolation is a method of estimating unknown values using known data. more precisely, if the function $f(x)$ of the real variable x is unknown and the function value $f(x_i)$ for the value $x_i (i = 1, 2, \cdots, n)$ of two or more variables with a certain interval is known, estimating a function value for any x in between is called interpolation [30]. In time-series analysis, methods of interpolation are largely divided into deterministic or stochastic methods, which are

Algorithm 1: Augmentation based on Interpolation

Input: Time-Serie $T = \{t_1, t_2, \cdots, t_n\}$
Input: $RandomIndexing = true$ or $false$
Input: R, the ratio of data collection cycle
Output: Synthetic Time-Series $\hat{T}$
$\hat{T} = \{0, \cdots, 0\}$, length L
Let $X = \{1, 2, \cdots, n\}$, length L
Let $X^* = \left\{1, \frac{R+1}{R}, \frac{R+2}{R}, \cdots, n\right\}$, length $L \cdot R - R + 1$
$Interpolator = InterpolationFunction(X, T)$
$T^* = Interpolator(X^*)$
generate random integer K, where $0 < K < R$
for $i = 0$ **to** $L - 1$ **do**
  if $RandomIndexing == false$ **then**
    $\hat{T}[i] = T^*[i \cdot R + K]$
  else
    generate random integer K^*, where $0 < K^* < R$
    $\hat{T}[i] = T^*[i \cdot R + K^*]$
  end if
end for

further subdivided according to the theory on which each method is based. In addition, the interpolation method can vary depending on the data type and source.

The main goal of the proposed method is that it should be simple to use and robust while being able to preserve the manifold of the main components of the original time-series data. In order to achieve our goal, we make some assumptions and proceed with our ideas.

- The data collection cycle of the original time-series is never short but rather long.
- The original time-series obtained through long data collection cycles do not reflect the real world information originally intended to be observed.
- If additional cost and higher collection techniques can be used to shorten the data collection cycle, it can reflect more information from the real world.

In fact, in the signal processing field, it is very important to select an appropriate sampling rate in order to obtain discrete data points that sufficiently contain the components of the original signal. Discrete data points collected at a low sampling rate may not have enough information from the original signal. Of course, there are differences in the domains, but this can be an example to demonstrate that more value can be created when the data collection cycle is shorter than the original cycle. For this reason, we assume that there are real world data points that are considered to have failed to collect due to cost or technology limitations. The data points that are considered failed to be collected are reconstructed using interpolation, and they are considered new data and used to train the model.

First, each collection point for the original time-series T is called X, where T is the, in Section 3.1, function value $f(x_i)$. Here, the virtual collection points between X to be filled by interpolation are defined as X^*, where X^* is in line with real value x in section 3.1, the density of the virtual collection points is determined by R, the input to Algorithm 1. X^* is used as the input to an interpolator based on an arbitrary interpolation function, which obtains the interpolated time-series T^*. Finally, we generate a new time-series $\hat{T}$ by extracting random data points from T^*. When extracting $\hat{T}$, whether or not to extract data points at regular intervals from T^* depends on $RandomIndexing$.

4. Experiment and Performance Analysis

4.1. Experimental Setup

We experimented with 85 datasets provided by UCR Time Series Classification Archive to evaluate the performance of the proposed methodology. The datasets provided by UCR Archive have been collected from various applications, and there is a label for each data, which is suitable for conducting experiments and performance analysis [31]. We perform an experiment to check whether the data samples generated by the proposed method affect the performance of the classification model. In this paper, the most commonly used cubic interpolation is used to augmentation the original time-series. This corresponds to the interpolation function of Algorithm 1. Additionally, in Algorithm 1, we set R to 10 and $RandomIndexing$ to true. The classification model used residual network, the latest model for time-series classification [32] and doubled the size of the training set using the proposed data augmentation.

4.2. Experiment result

Our experimental results show that the data generated by data augmentation based on interpolation affects the performance of deep learning algorithm. In Figure 1, each point is a set of data provided by the UCR archive, and points in the upper left area represent data sets in which the proposed augmentation technique has a positive effect on model performance. In contrast, the points in the lower right area indicate that augmentation was negatively affected by model performance. In order to check the performance difference by data type, different colors were plotted for each type. W / T / L results show that the proposed method has a positive effect on 54 datasets, but does not affect 9 datasets. For 22 data sets, we found that the proposed method had a rather negative impact.

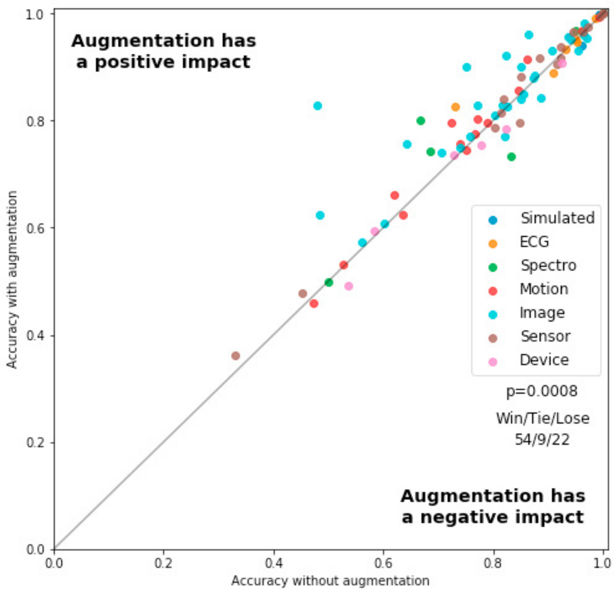

Fig. 1: Accuracy of residual network with/without augmentation

We performed a one-sided wilcoxon signed rank test to confirm whether the data augmentation technique using the proposed technique has a statistically positive effect on model performance. As a result, the P-value was 0.0008, which is strong statistical evidence that the proposed method has a positive effect on performance improvement.

Table 1: Accuracy of residual network with/without augmentation (average by data type)

Data Type	$Accuracy_{augmentation}$	$Accuracy_{non-augmentation}$
Simulated	97.8%	97.7%
ECG	93.4%	92.2%
Spectro	81.5%	79.9%
Motion	76.3%	74.7%
Image	84.0%	80.2%
Sensor	85.5%	84.9%
Device	71.1%	72.9%
Total	83.5%	81.7%

We also checked the accuracy of each data type to see whether the data generated by interpolation based data augmentation have different effects depending on the data type. According to Table 1, the proposed method has at least no negative effect except when the data type is not device.

5. Conclusion and Future Work

In this paper, we show how to generate enough data to train the deep learning model even when the training sample is insufficient by using time-series data augmentation based on interpolation.

The proposed method does not slicing and rearranging the original time-series, so it has the advantage of being robust against label variations even if the major components of the time-series (trend, cyclical, and seasonal) change over time. We used cubic splines as the interpolation function, which is known to be the most general, to perform the experiment, but it is a kind of hyperparameter. Interpolation is subdivided in a variety of ways, which means that a variety of interpolation methods suitable for different applications are already studied beforehand. In other words, the proposed method can achieve more positive results if the appropriate interpolation function is selected and used according to the user's application. This means that the time-series data augmentation problem, which has not been studied much, can be extended to time-series interpolation, which has already been studied. Therefore, in future work, we can find or develop interpolation function that is well suited to the method proposed in this paper in the field of chronic data sparsity problem.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2020-2018-0-01417) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation), and the Smart Factory Technological R&D Program S2727115 funded by Ministry of SMEs and Startups (MSS, Korea)

References

[1] Domingos, Pedro. (2012) "A few useful things to know about machine learning." *Communications of the ACM.* **55** (10): 78–87.

[2] Dalal, Navneet and Triggs, Bill. (2005) "Histograms of oriented gradients for human detection." *IEEE computer society conference on computer vision and pattern recognition.* **1** 886–893.

[3] Lowe, David G. (1999) "Object recognition from local scale-invariant features." *Proceedings of the seventh IEEE international conference on computer vision.* **2** 1150–1157.

[4] Rublee, Ethan and Rabaud, Vincent and Konolige, Kurt and Bradski, Gary. (2011) "International conference on computer vision." 2564–2571.

[5] Bay, Herbert and Ess, Andreas and Tuytelaars, Tinne and Van Gool, Luc. (2008) "Speeded-up robust features (SURF)." *Computer vision and image understanding.* **110** (3): 346–359.

[6] Nanopoulos, Alex and Alcock, Rob and Manolopoulos, Yannis. (2001) "Feature-based classification of time-series data." *International Journal of Computer Research* **10** (3) 49–61.

[7] Esling, Philippe and Agon, Carlos. (2012) "Time-series data mining." *ACM Computing Surveys (CSUR).* **45** (1) 1–34.

[8] Bengio, Yoshua and Courville, Aaron and Vincent, Pascal. (2013) "Representation learning: A review and new perspectives." *IEEE transactions on pattern analysis and machine intelligence.* **35** (8) 1798–1828.

[9] Hinton, Geoffrey and Deng, Li and Yu, Dong and Dahl, George E and Mohamed, Abdel-rahman and Jaitly, Navdeep and Senior, Andrew and Vanhoucke, Vincent and Nguyen, Patrick and Sainath, Tara N and others. (2012) "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups." *IEEE Signal processing magazine.* **29** (6) 82–97.

[10] Krizhevsky, Alex and Sutskever, Ilya and Hinton, Geoffrey E. (2012) "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems.* 1097–1105.

[11] Le Guennec, Arthur and Malinowski, Simon and Tavenard, Romain. (2016) "Data augmentation for time series classification using convolutional neural networks." *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data.*

[12] Ramponi, Giorgia and Protopapas, Pavlos and Brambilla, Marco and Janssen, Ryan. (2018) "T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling." *arXiv preprint arXiv:1811.08295.*

[13] Brain, Damien and Webb, Geoffrey I. (1999) "On the effect of data set size on bias and variance in classification learning." *Proceedings of the Fourth Australian Knowledge Acquisition Workshop, University of New South Wales.* 117–128.

[14] Baird, Henry S. (1992) "Document image defect models." *Structured Document Image Analysis.* 546–556.

[15] Shorten, Connor and Khoshgoftaar, Taghi M. (2019) "A survey on image data augmentation for deep learning." *Journal of Big Data.* **6** (1) 60.

[16] Jaitly, Navdeep and Hinton, Geoffrey E. (2013) "Vocal tract length perturbation (VTLP) improves speech recognition." *ICML Workshop on Deep Learning for Audio, Speech and Language.* **117**.

[17] Cui, Xiaodong and Goel, Vaibhava and Kingsbury, Brian. (2015) "Data augmentation for deep neural network acoustic modeling." *IEEE/ACM Transactions on Audio, Speech, and Language Processing.* **23** (9) 1469–1477.

[18] Ko, Tom and Peddinti, Vijayaditya and Povey, Daniel and Khudanpur, Sanjeev. (2015) "Audio augmentation for speech recognition." *Sixteenth Annual Conference of the International Speech Communication Association.*

[19] Lotte, Fabien. (2015) "Signal processing approaches to minimize or suppress calibration time in oscillatory activity-based brain–computer interfaces." *Proceedings of the IEEE.* **103** (6) 871–890.

[20] Um, Terry T and Pfister, Franz MJ and Pichler, Daniel and Endo, Satoshi and Lang, Muriel and Hirche, Sandra and Fietzek, Urban and Kulić, Dana. (2017) "Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks." *ACM International Conference on Multimodal Interaction.* 216–220.

[21] DeVries, Terrance and Taylor, Graham W. (2017) "Dataset augmentation in feature space." *arXiv preprint arXiv:1702.05538.*

[22] Krell, Mario Michael and Seeland, Anett and Kim, Su Kyoung. (2018) "Data Augmentation for Brain-Computer Interfaces: Analysis on Event-Related Potentials Data." *arXiv preprint arXiv:1801.02730.*

[23] Steven Eyobu, Odongo and Han, Dong Seog. (2018) "Feature representation and data augmentation for human activity classification based on wearable IMU sensor data using a deep LSTM neural network." *Sensors.* **18** (9) 2892.

[24] Hyndman, Rob J and Athanasopoulos, George. (2018) "Forecasting: principles and practice." *OTexts.*

[25] Forestier, Germain and Petitjean, François and Dau, Hoang Anh and Webb, Geoffrey I and Keogh, Eamonn. (2017) "Generating synthetic time series to augment sparse datasets." *IEEE international conference on data mining.* 865–870.

[26] Petitjean, François and Ketterlin, Alain and Gançarski, Pierre. (2011) "A global averaging method for dynamic time warping, with applications to clustering." *Pattern Recognition.* **44** (3) 678–693.

[27] Mirza, Mehdi and Osindero, Simon. (2014) "Conditional generative adversarial nets." *arXiv preprint arXiv:1411.1784.* 2672–2680

[28] Hannun, Awni and Case, Carl and Casper, Jared and Catanzaro, Bryan and Diamos, Greg and Elsen, Erich and Prenger, Ryan and Satheesh, Sanjeev and Sengupta, Shubho and Coates, Adam and others. (2014) "Deep speech: Scaling up end-to-end speech recognition." *arXiv preprint arXiv:1412.5567.*

[29] Cui, Zhicheng and Chen, Wenlin and Chen, Yixin. (2016) "Multi-scale convolutional neural networks for time series classification." *arXiv preprint arXiv:1603.06995.*

[30] Lunardi, Alessandra. (2018) "Interpolation theory." *Springer.* **16**.

[31] Chen, Yanping and Keogh, Eamonn and Hu, Bing and Begum, Nurjahan and Bagnall, Anthony and Mueen, Abdullah and Batista, Gustavo. (2015) "The UCR Time Series Classification Archive." `www.cs.ucr.edu/~eamonn/time_series_data/`.

[32] Fawaz, Hassan Ismail and Forestier, Germain and Weber, Jonathan and Idoumghar, Lhassane and Muller, Pierre-Alain. (2019) "TiDeep learning for time series classification: a reviewtle." *Data Mining and Knowledge Discovery.* **33** (4) 917–963.