Data Augmentation techniques in time series domain: A survey and taxonomy

EDGAR TALAVERA*†, Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos, Spain
GUILLERMO IGLESIAS*, Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos, Spain
ÁNGEL GONZÁLEZ-PRIETO, Universidad Complutense de Madrid Instituto de Ciencias Matemáticas (CSIC-UAM-UCM-UC3M), Spain
ALBERTO MOZO, Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos, Spain
SANDRA GÓMEZ-CANAVAL, Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos, Spain

With the latest advances in deep learning generative models, it has not taken long to take advantage of their remarkable performance in the area of time series. Deep neural networks used to work with time series depend heavily on the breadth and consistency of the datasets used in training. These types of characteristic are not usually abundant in the real world, where they are usually limited and often with privacy constraints that must be guaranteed. Therefore, an effective way is to increase the number of data using Data Augmentation (DA) techniques, either by adding noise or permutations and by generating new synthetic data. It is systematically review the current state-of-the-art in the area to provide an overview of all available algorithms and proposes a taxonomy of the most relevant researches. The efficiency of the different variants will be evaluated; as a vital part of the process, the different metrics to evaluate the performance and the main problems concerning each model will be analysed. The ultimate goal of this study is to provide a summary of the evolution and performance of areas that produce better results to guide future researchers in this field.

CCS Concepts: • **Computing methodologies** → **Neural networks**; **Model verification and validation**.

Additional Key Words and Phrases: Deep Learning, Time series, Data Augmentation, Generative Models

ACM Reference Format:
Edgar Talavera, Guillermo Iglesias, Ángel González-Prieto, Alberto Mozo, and Sandra Gómez-Canaval. 2022. Data Augmentation techniques in time series domain: A survey and taxonomy. *ACM Comput. Surv.* 0, 0, Article 0 (2022), 25 pages. https://doi.org/XXXXXXX.XXXXXXX

*Both authors contributed equally to this research.
†Corresponding author.

Authors' addresses: Edgar Talavera, e.talavera@upm.es, Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos, Madrid, Spain; Guillermo Iglesias, guillermo.iglesias.hernandez@alumnos.upm.es, Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos, Madrid, Spain; Ángel González-Prieto, Universidad Complutense de Madrid, Instituto de Ciencias Matemáticas (CSIC-UAM-UCM-UC3M), Madrid, Spain; Alberto Mozo, Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos, Madrid, Spain; Sandra Gómez-Canaval, Universidad Politécnica de Madrid. Departamento de Sistemas Informáticos, Madrid, Spain.

0360-0300/2022/0-ART0 $15.00
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

From Deep Learning (DL) advent, an important part of the effort of the research and industrial community has focused on solving and improving supervised training tasks. Supervised learning requires a dataset with various features in which each sample must be labelled. The most representative problems to solve using supervised learning techniques are classification, regression, and structured pattern output problems.

Traditionally, Machine Learning (ML) models for supervised tasks fall into Discriminative model category. Discriminative modelling is synonymous with supervised learning, or learning a function that maps an input to an output using a labelled dataset. From a formal point of view, Discriminative modelling estimates $p(y|x)$, that is, the probability of a label y given observation x.

However, a major problem exists when attempting to train one of these models on an incomplete, unbalanced, or privacy-challenged dataset. Typically, these problems are addressed by preprocessing dataset techniques, such as subsampling, or in datasets that are not large enough, by DA techniques [22, 25].

However, as problems arise, technology evolves to address these boundaries. In recent years, Artificial Neural Networks (ANNs) and its application in the field of DL, has experienced a period of great advances. Although a multitude of models have contributed to this expansion, one of the most revolutionary that has been proposed appeared in 2014 by Ian Goodfellow [34],with his Generative Adversarial Networks (GANs).

GANs are certainly not the earliest generative architecture ever introduced; already in 1987 Yann Lecun [64] suggested in his thesis the Autoencoder (AE) architectures, which were capable of generating data modifications received as input. But it is not until the incorporation of directed probabilistic models into AE architectures, also known as Variational Autoencoder (VAE) [58], that the models begin to be presented as capable of generating synthesised data.

Although these networks show impressive results, the capabilities of GANs have been shown to be far ahead, with impressive results applied to the field of imaging. Nevertheless, this is not the unique area of application; synthetic data generation is a powerful boost to synthesise sensitive data, such as those in the world of telecommunications.

Thus, this paper aims to review all the existing technologies for DA and data generation, and to review the positive and negative aspects of each of them.

2 RELATED WORKS

Recently, a number of high-quality data augmentation review articles have been published [16, 27, 75]. However, most of them are focused on more popular areas such as imaging, video or Natural Language Processing (NLP). Although these techniques focus on correcting the imbalance or incompleteness of the dataset, there are other areas of application where these problems are more common. The scarcity of valid datasets is not as clear in all areas of deep learning applications as in time series.

In a first approach to the literature review, in [49] an approximation of DA algorithms is made for use in neural network algorithms for time series classification. In the survey, they evaluated 12 methods to enhance time series data in 128 time series classification datasets with six different types of neural network. Other recent studies focus more specifically on the use of GANs for data augmentation, as in [12], where they focus on the taxonomy of discrete-variant GANs and continuous-variant GANs, in which GANs deal with discrete time series and continuous time series data.

However, improving data sources to feed Artificial Intelligence (AI) algorithms is not limited to DA exclusively. Therefore, some studies have decided to take the path of building synthetic traffic generators to generate their datasets almost from scratch; some examples focus on this aspect [1, 33, 80]. In this way, they are able to abstract from the dataset itself, which is only necessary to understand the distribution of the data. Furthermore, in [82]

they set out a further study of the repercussions of these technologies, highlighting one of the major advantages of generating synthesised data, the abstraction of privacy issues, and the ease of obtaining datasets.

Despite the possibilities presented by new technologies in this area to improve the quality of datasets used for time series, there are not many studies that compile all technologies. Therefore, we want to contribute to reduce the existing gap in the area by trying to bring together all the time series DA and data generator algorithms that currently exist, contrasting their possible virtues, approaches, and differences to help future researchers position themselves in the area.

3 BACKGROUND

3.1 Traditional algorithms

DA has been a crucial task when the available data is unbalanced or insufficient. Traditionally, in fields such as image recognition, different transformations have been applied to data such as cropping[39, 61, 91, 94], scaling[39, 91], mirroring[45, 61, 94], color augmentation[39, 61, 70] or translation[45].

DA in time series cannot be done with these algorithms directly due to the particularity of the data itself. Due to the diversity of time series data, not all techniques can be applied to every dataset. Some of the previous algorithms used in computer vision can be adapted to a time series domain, but, in other cases, new specific algorithms must be designed to treat with time series data.

Another important factor when applying DA to the time series domain, specially in signal processing, is that manipulation of the data could distort the signal too much, leading to negative training.

We will define traditional algorithms for all the techniques based on taking data input samples and synthesising new samples by modifying these data and applying different transformations. The main difference between this technique and those that we reviewed in sections 3.2 and 3.3 is that, in the former algorithms, the transformations are applied directly to the data, while in the latter the objective is to learn the probability distribution of the data in order to generate completely new samples trying to imitate the data distribution.

3.2 Variational Autoencoder (VAE)

VAEs are neural generative models first introduced by Diederik P. Kingma and Max Welling[58], this algorithm is based on the AE architecture[64] proposed in 1987. AEs allow changing typical artificial intelligence problems, such as linear regression or classification, to domain-shifting problems. In order to perform this, AEs take an input, usually an image, and infer, as output modifications of that same input, that the most generalised fields are image denoising [8].

AE Network is composed of two components, an Encoder and a Decoder. The Encoder is in charge of reducing the input dimensionality of the data to a latent space, while the Decoder reconstructs the input information from this latent representation. This latent space is a lower-dimensional manifold of the input data. Then, synthetic data is generated, interpolating the values of the latent space and decoding them. However, this interpolation of the latent space does not generate completely new values; it just mixes the features of the learned probability distribution.

In order to avoid overfitting produced in AE, VAE regularises its training, generating more diverse samples. The main difference between both architectures is that VAE encodes the input information in a probability distribution rather than in a point. Then, from this distribution, it samples a point that is then decoded to synthesise new samples.

This intermediate step allows the network to map the input distribution to a lower-dimensional distribution from which new latent points can be generated. To do so, the latent distribution is normally defined by a normal distribution with a mean $\vec{\mu} = (\mu_1, \ldots, \mu_n)$ and a standard deviation $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$. These mean and standard deviation vectors define the latent distribution of the model.

Leaving the network to learn a distribution, instead of a set of points learned in AE, the decoder network associates the features of the input data with the probability areas with their respective mean and deviation. With this representation, the mean of the distribution defines the center point from which the synthetic samples will be generated and the standard deviation defines the variability in the output, that is, the diversity of the generated samples.

Figure 1 shows the architecture of a VAE network.

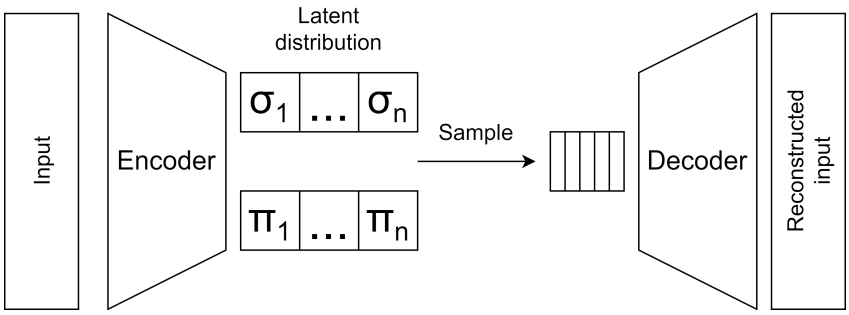


Fig. 1. VAE architecture.

3.3 Generative Adversarial Networks (GANs)

GANs are a generative neural model based on a competition between two Neural Networks (NNs), they were first introduced by Ian Goodfellow[34] in 2014. The objective of the architecture is to replicate a given data distribution in order to synthesise new samples of the distribution. To achieve this goal, the GAN architecture is composed of a Generator (G) model and a Discriminator (D) model. The former is in charge of generating the synthetic samples of the data distribution, while the latter tries to distinguish the real samples from the synthesised samples.

To reach the goal of generating completely new data that are indistinguishable from the input data distribution, both models interact with each other. The G generates samples trying to replicate, without copying, the distribution, while the D discriminates the real samples from the fake samples. In this way, when D differentiates both distributions, it feedbacks G negatively; on the other hand, when D is not capable of differentiating each distribution, its positively feedbacks G. In doing so, G evolves to fool D. At the same time, D is positively rewarded when discrimination is done correctly.

This competition encourages both networks to evolve together. If D fails in its task, G will not evolve because it will always succeed, despite the quality of the synthesised samples. Although if D always perfectly differentiates both distributions, G will not be able to fool D, making it impossible to evolve.

The standard GAN architecture can be seen in Figure2.

From a mathematical perspective, this competitive behavior is based on Game Theory, where two players compete in a zero-sum game. The D estimates $p(y|x)$, where y is the label (true or fake) of the given sample x. And G generates synthetic samples from a latent vector z, which can be denoted as $G(z)$.

From a formal point of view, this competition is defined as a minimax game where D tries to maximize its accuracy when discriminating between both distributions and G tries to minimize this accuracy. The formulation of this process is denoted as follows:

$$\begin{aligned} \min_G \max_D L(D,G) = E_{x\sim p_r} \log[D(x)] \\ + E_{z\sim p_z} \log[1 - D(G(x))] \end{aligned} \tag{1}$$

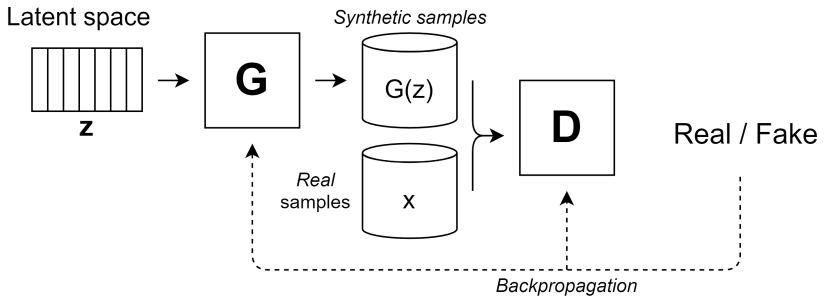


Fig. 2. GAN architecture.

where z is the latent vector, which is generated randomly by a uniform or Gaussian distribution and $x \sim p_r$ is the real distribution.

In the publication in which GANs was presented[34] it was proved that the architecture can converge to a unique solution. This point, known as Nash equilibrium, is in practice very difficult to achieve due to the instability behavior of GANs. The Nash equilibrium is characterized by the fact that none of the networks can reduce their respective losses.

4 EVALUATION METRICS

Due to the particularity of the time series field, there is no unique metric to evaluate the reliability of algorithms in all of their applications. Finding a measurement capable of evaluating the quality and diversity of the synthesised data is still an open issue.

For example, in GAN networks, there is no consensus on evaluation metrics between the different studies. In addition, most of the evaluation metrics designed are centred on computer vision, since it is the most popular field for this kind of network.

Therefore, we will describe the most commonly used metrics that have been used to evaluate the algorithms we discuss in this article. However, it should be noted that to choose a proper evaluation metric, one should adapt the metric to the specific data augmentation algorithm and the application field.

4.1 External performance evaluation

When applying DA to a dataset, the most common objective is to generate new data samples to improve the performance of certain models, reducing the imbalance of the data or the lack of data. One of the most popular ways to measure how the addition of new data changes the behavior of the models is simply to compare these models before and after DA. Then it is possible to compare whether each model has improved its performance after applying DA to the input data.

This approximation is purely practical and relies on the correlation between the performance of a defined model and not the quality of the synthetic samples themselves. Most traditional algorithms base their performance on this method because it is a straightforward method to evaluate an algorithm.

In [9] the performance of the DA algorithms they propose is achieved using *symmetric Mean Absolute Percentage Error* and *Mean Absolute Scaled Error*, which are the two most common evaluation metrics used in forecasting. Research compares the values of these metrics before and after applying DA to the dataset, then evaluates how the models improve their performance due to the addition of more data to the training set.

In [49] they used six different neural networks to evaluate how each DA algorithm affects the classification of the data. In particular, they evaluated VGG[91], Residual Network[39], Multilayer Perceptron[100], Long Short-Term

Memory[85], Bidirectional Long Short-Term Memory[89] and Long Short-Term Memory Fully Convolutional Network[54]. Then, the changes in the accuracy of the models are compared, observing how certain DA algorithms benefit the performance of the models, while in other cases it gets worse. The main drawback stated in the article is that each architecture has its particularities, given the different results for each algorithm and making it a difficult task to differentiate the best algorithm. Furthermore, because all of them are neural models, it is difficult to interpret some of the results.

The approximation followed in [62] is comparing different DA techniques by the increase in accuracy produced in each case of study. It is worth mentioning how this approximation adapts to each application without having to change anything. The authors of the article are able to compare very different techniques such as Noise Addition, GAN, Sliding Window, Fourier-Transform and Recombination of Segmentation under the same criteria for a specific domain purpose. This example shows how this approximation easily adapts to different DA techniques, making it possible to compare the results for a certain task.

A similar approximation to measure the quality of the new generated data is to compare different models using a defined loss function. This approximation was followed in GANs architectures such as [34, 48, 97, 103, 106], where a comparison between networks is possible using the same loss function to evaluate their training. Then, they correlate the quality of the synthetic data with this value. This approximation can be applied naturally to the time series domain, allowing comparison between different networks. However, the main drawback of this method is that it compares the performance of different neural models and cannot be applied to other models. It should be noted that, as in previous metrics, it correlates the quality of the generated data not with the data itself but rather with the performance of the model.

In [102] they compare the performance of different DA techniques with Mean Per-Class Error (MPCE). This metric, proposed in [101], measures the error per class in J datasets taking into account the number of classes in each dataset. The main particularity of MPCE is that it allows us to quantify the performance of an algorithm for different datasets. The MPCE is calculated as follows:

$$\text{MPCE} = \sum_{j \in [J]} \text{PCE}_j = \frac{e_j}{c_j} \tag{2}$$

where e_j is the error rate and c_j is the number of classes in each dataset. This metric is capable of taking into account the number of classes of each dataset, in order to normalise the comparison between different sources of data.

4.2 GAN related metrics

Since the introduction of GAN it has always been an open issue to measure the quality of the synthesised samples produced by the architecture[11]. One of the most important difficulties when designing a metric for GANs is the ability to capture both the quality and diversity of the generated data.

In addition to being still an open issue, there is consensus on some metrics and many articles measure their results with the same metrics[32, 55–57, 81, 107]. The main problem in the time series domain is that it is not always possible to adapt the metrics to the particularities of this field because most of the metrics are designed to be useful in computer vision-related tasks.

Over the past few years, some articles have been developed suggesting applying DA to time series data and treating it as if it were an image[38, 98]. These articles use GAN networks to synthesise new time series data, but to convert the signal data into an image. In these cases, traditional GAN metrics, such as the Inception Score[88], Mode Score[36] or Fréchet Inception Distance[40] are used to evaluate the results. These metrics are based on how the *Inception v3* neural classifier distinguishes the different samples. The idea is to measure the entropy of the synthetic dataset using an external classifier.

In addition to the field of computer vision, studies have been developed that apply GAN directly to time series. That is, in time series GAN (TimeGAN)[104] two new metrics are proposed to assess the quality of the generated samples. The Discriminative Score is based on the use of an external pre-trained model, as was done with the Inception Score, consisting of a 2-layer Long-Short Term Memory (LSTM). The Discriminative Score measures how this model distinguishes between real and fake samples and the classification error corresponds to the Discriminative Score. Furthermore, the Predictive Score also uses a 2-layer LSTM, but in this case, this model is trained with synthetic samples. The model is then evaluated using the original dataset. The Predictive Score corresponds to Mean Absolute Error (MAE) of the model trained with the synthetic samples evaluated with the real samples.

4.3 Similarity measurements

This set of metrics focuses on the comparison of two probability distributions. The idea is to measure how far from the original distribution the synthetic samples generated with DA. The main advantage of these metrics is that they focus on directly studying the quality of the data, in contrast to previous reviewed methods that measured the quality indirectly. Another advantage of these types of metrics is that they can be applied to synthetic data despite the algorithm used to generate them.

An empirical and qualitative approach to measuring the differences between two distributions is to reduce the dimensionality of the data and perform a visual comparison. The objective is to reduce the dimensionality of the data to plot the samples in a bidimensional space; an empirical comparison is then made by visualising the data. This approach was followed in [104] where they applied T-distributed Stochastic Neighbour Embedding (t-SNE) and Principal Component Analysis (PCA). Then they compared the distribution of the data in the two-dimensional space for TimeGAN[104], Recurrent Conditional GAN (RCGAN)[23], Continuous recurrent GAN (C-RNN-GAN)[72], T-Forcing[93], WaveNet[96] and WaveGAN[21]. This approach was also followed in [50] where they used PCA to compare different traditional algorithms for the GunPoint dataset from the 2018 UCR Time Series Archive[18].

Kullback-Leibler Divergence (KL Divergence) has been used in work such as [19, 30] to measure similarities between synthetic and real datasets. KL Divergence is defined as follows:

$$D_{KL}(P||Q) = \sum_{i=1}^{|v|} P_v(i) \frac{P_v(i)}{Q_v(i)} \tag{3}$$

where P and Q are the probability distributions whose distance is calculated and i is each sample of the distribution.

In [14] a novel measurement is proposed to quantify the distance between the time series distribution. It is based on calculating the Wasserstein distance between time series data. The metric is defined by measuring the Wasserstein distance of the energy between frequencies. The Wasserstein-Fourier distance between the probability distributions is computed as follows:

$$\mathrm{WF}([x], [y]) = W_2\left(s_x, s_y\right) \tag{4}$$

where s_x and s_y are the normalised power spectral densities of the distributions.

5 DATA AUGMENTATION ALGORITHMS REVIEW

During this section, different state-of-the-art algorithms will be reviewed. This section will explain the particularities, strengths and weaknesses of each algorithm. In addition, the different approximations to apply DA will be grouped and related between them. A taxonomy of the different trends and lines of research will be proposed, showing the different existing links between the works of the last years.

It should be noted that not all the algorithms can be applied to all types of time series data; in some cases, the algorithms proposed will be heavily focused on a certain application, while in others more general techniques will be studied.

5.1 Basic DA Methods

The basic DA algorithms that will be reviewed in this section are all techniques that use data manipulation to generate new synthetic data samples using existing samples and transforming the original samples. All these techniques have as their base the deformation, shortening, enlargement or modification of the data samples of the dataset. This group of techniques has been traditionally used in fields such as computer vision and, in some cases, the same algorithms can be adapted to process time series data, but in others, new algorithms must be designed specifically to use time series data as input.

Therefore, we will review and discuss the most important traditional algorithms that have been applied to time series data, outlining their particularities, advantages and disadvantages. Figure 3 shows the taxonomy proposed for the different algorithms reviewed.

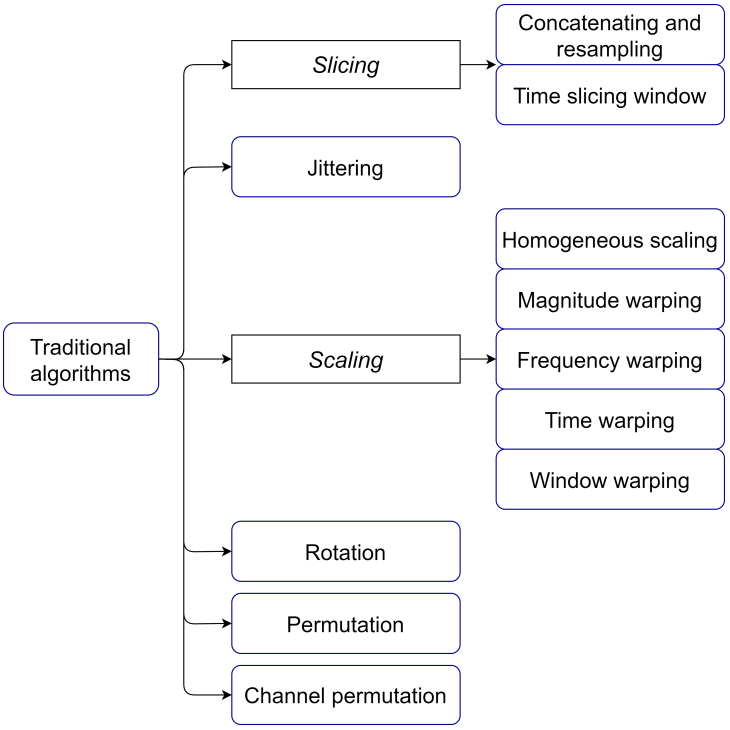


Fig. 3. Traditional DA algorithms taxonomy.

5.1.1 Time slicing window. Slicing, in time series, consists of cutting a portion of each data sample, to generate a different new sample. Normally, slicing is applied to the last steps of the sample, but the snippet of the original sample can be obtained from any step. When the original data is cropped, a different sample is produced, but unlike image processing, it is difficult to maintain all the features of the original data. The process of slicing time

series data can be denoted as:

$$x'(W) = \left\{x_{\varphi}, \cdots, x_t, \cdots, x_{\varphi+W}\right\} \tag{5}$$

where W is the slice window that defines the crop size and φ is the initial point from where the slicing is performed, such as $1 \leq \varphi \leq T - W$. One of the most important drawbacks of slicing the signal is that it can lead to invalid synthetic samples because it can cut off important features of the data.

A variation of the slicing method is proposed in [13], where the concatenating and resampling method is presented. This algorithm first detects features in the data, called characteristic points denoted with q, this is made by using the Pan-Tompkins QRS detector[77]. This algorithm detects the characteristic points in a heartbeat signal, so in order to apply the concatenating and resampling algorithm it must be defined and algorithm to detect these points. Then, after detecting the characteristic points, it is defined a subsequence SS that starts and ends in a characteristic point. This sequence is replicated k times and sliced in a window W to perform DA.

The concatenating and resampling algorithm tries to ensure the validity of the data, taking into account that the signal maintains its features. But the main disadvantage of this method is that it needs a detector of characteristic points that ensure the data validity.

5.1.2 Jittering. Jittering consists of adding noise to time series to perform DA. This technique, in addition to being one of the simplest forms of DA is one of the most popular in time series[28, 84]. Jittering assumes that the data are noisy which, in many cases, i.e., when dealing with sensor data, is true.

Jittering tries to take advantage of the noise of the data and simulate it to generate new samples. Typically, Gaussian noise is added to each time step; the mean and standard deviation of this noise define the magnitude and shape of the deformation, so it is different in each application. The jittering process can be defined as follows:

$$x'(\epsilon) = \{x_1 + \epsilon_1, \cdots, x_t + \epsilon_t, \cdots, x_T + \epsilon_T\} \tag{6}$$

where ϵ is refers to the noise addition at each step of the signal.

As mentioned above, the jittering process must be adapted to each case, because there are cases such as [95] where the effects of jittering lead to negative learning.

5.1.3 Scaling. Scaling consists of changing the magnitude of a certain step in time series domain. The idea is to maintain the overall shape of the signal while changing its values. With scaling, the new generated data change the range of values, but keep the shape of the changes. Homogeneous scaling can be denoted as:

$$x'(\alpha) = \{\alpha x_1, \cdots, \alpha x_t, \cdots, \alpha x_T\} \tag{7}$$

where α defines the scale of the change, this value can be defined by a Gaussian distribution with mean 1 and with σ as a hyperparameter[95], or it can be previously defined from a list of values[84].

Within scaling techniques, there are several different approximations for a specific time series domain. They take advantage of the specific properties of the signal data and adapt to perform DA.

Magnitude warping is a technique used in [95] that consists of applying a variable scaling to different points of the data curve. To define where to apply the transformation, a set of knots $\mathrm{u} = u_1, \ldots, u_i$ is defined; these represent the step in which the scaling is performed. Then the magnitude of the scaling is defined by a cubic spline interpolation of the knots, such as $S(\mathrm{x})$. Then the magnitude warping can be defined as follows:

$$x'(\alpha) = \{\alpha_1 x_1, \cdots, \alpha_t x_t, \cdots, \alpha_T x_T\} \tag{8}$$

where $\alpha = \alpha_1, \ldots, \alpha_i = S(\mathrm{x})$. With magnitude warping, the main particularity is that it applies a smoothened scaling to each point of the curve, multiplying the possibilities of the transformation while preserving the overall shape of the data. However, it still assumes that the synthetic data maintain validity after transformation.

Frequency warping is a variation of magnitude warping, mostly applied in speech processing[2, 17, 59]. The most popular version in speech recognition is Vocal Tract Length Perturbation, which can be applied in a deterministic way[17] or stochastically within a range[51].

Another scaling technique is Time warping, the idea is very similar to magnitude warping, but the main difference between both algorithms is that Time warping modifies the curve in the temporal dimension. That is, instead of fluctuating the magnitude of the signal in each step, it stretches and shortens the time slices of the signal. To define how to warp the signal, a smooth curve, as was done in magnitude warping, is defined by using a cubic spline for a set of knots. The time warping algorithm can be denoted as:

$$x'(\tau) = \{\tau(1)x_1, \cdots, \tau(t)x_t, \cdots, \tau(T)x_T\} \tag{9}$$

where *tau* is the function of the curve, such as $\tau = S(\mathrm{x})$. This algorithm has been used in several works, such as [52, 79]. There is yet another variation of this algorithm, known as window warping, followed in [63] that defines a slice in the time series data and speeds up or down the data by a factor of $1/2$ or 2. In this case, the warping is applied to a defined slice of the whole sequence; the rest of the signal is not changed.

5.1.4 Rotation. Rotation can be applied to multivariate time series data by applying a rotation matrix with a defined angle. In univariate time series, rotation can be applied by flipping the data. Rotation is defined as follows:

$$x'(R) = \{Rx_1, \cdots, Rx_t, \cdots, Rx_T\} \tag{10}$$

where R defines the degree of rotation of each step. This algorithm is not very usual in time series due to the fact that rotating a time series sample could make it loss the class information[24]. On the other hand, there have been articles[95] that demonstrate the benefits of applying rotation, specially combined with other data transformations.

5.1.5 Permutation. Shuffling different time slices of data in order to perform DA is a method that generates new data patterns. It was proposed in [95], where a fixed slice window was defined from which the data is rearranged, but it has also been applied with variable windows[78]. The main problem of applying permutation is that it does not preserve time dependencies; thus, it can lead to unvalid samples. The permutation algorithm can be denoted as follows:

$$x'(w) = \left\{x_i, \cdots, x_{i+w}, \cdots, x_j, \cdots, x_{j+w}, \cdots, x_k, \cdots, x_{k+w}\right\} \tag{11}$$

where i, j, k represents the first index slice of each window, so that each is selected exactly once, and w denotes the window size if the slices are uniform $w = T/n$ where n is the number of total slices.

5.1.6 Channel permutation. Changing the position of different channels in multi-dimensional data is a common practice. In computer vision, it is quite popular to swap the RGB channels to perform DA[66]. With respect to time series, channel permutation can be applied as long as each channel of the data is still valid. The channel permutation algorithm, for multidimensional data such as $x = \{\{x_{11}, \cdots, x_{1T}\}, \cdots, \{x_{c1}, \cdots, x_{cT}\}\}$ where c is the number of channels, can be denoted as:

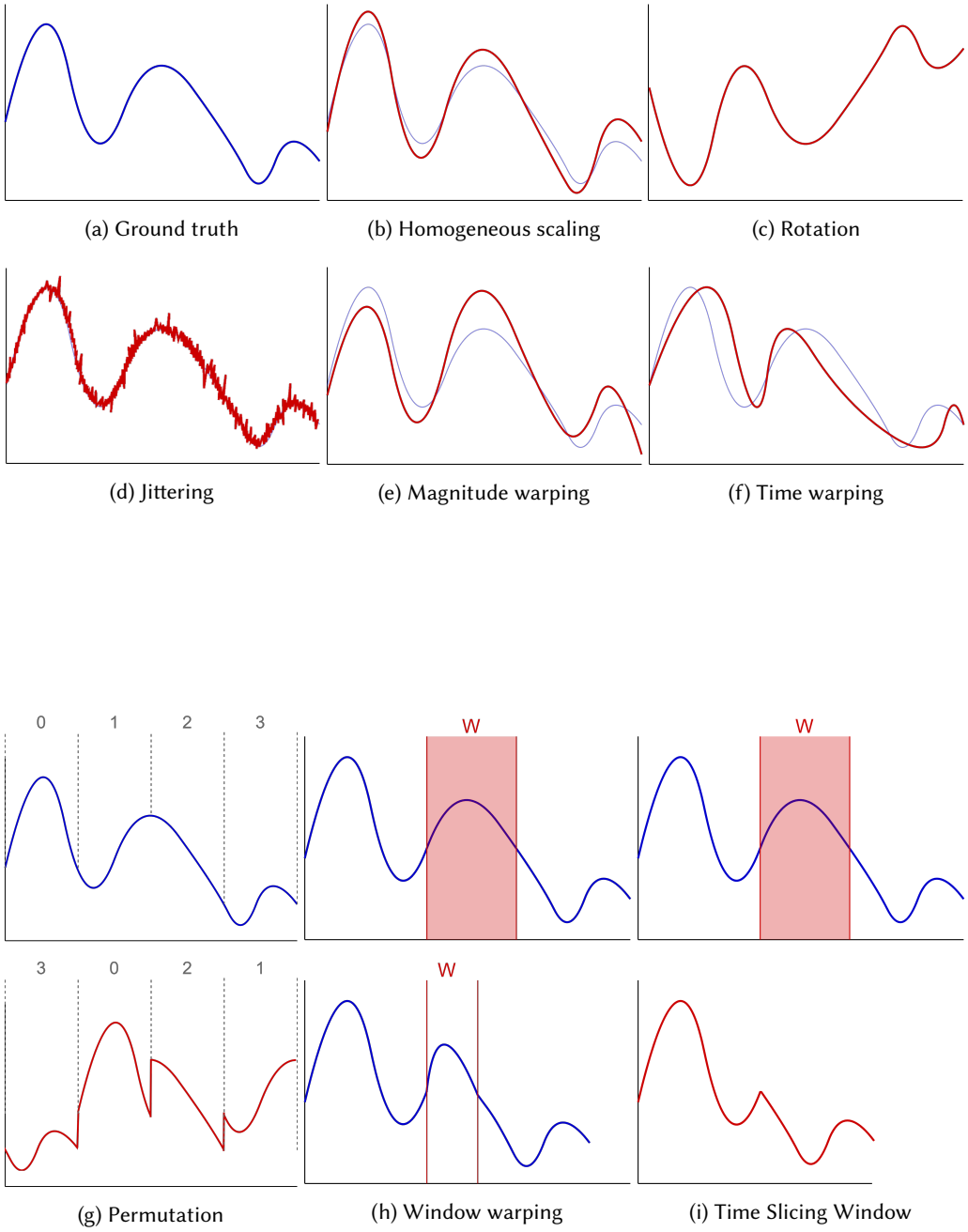
$$x = \left\{\{x_{i1}, \cdots, x_{iT}\}, \cdots, \left\{x_{j1}, \cdots, x_{jT}\right\}\right\} \tag{12}$$

where i and j represent the number of channels, so each index is selected exactly once.

In time series domain, this algorithm is not applicable to the application of the data, because permutation assumes that the channel information is independent of the channel itself. In other words, the information about the channels is not linked to the particular channel.

That is, in [30], they applied this algorithm by flipping the position of the sensors that recorded the data signals. In the article, the researchers used an exercise mat with eight proximity sensors that they flipped to generate new data. That is, in practice, changing the position of the signal channels.

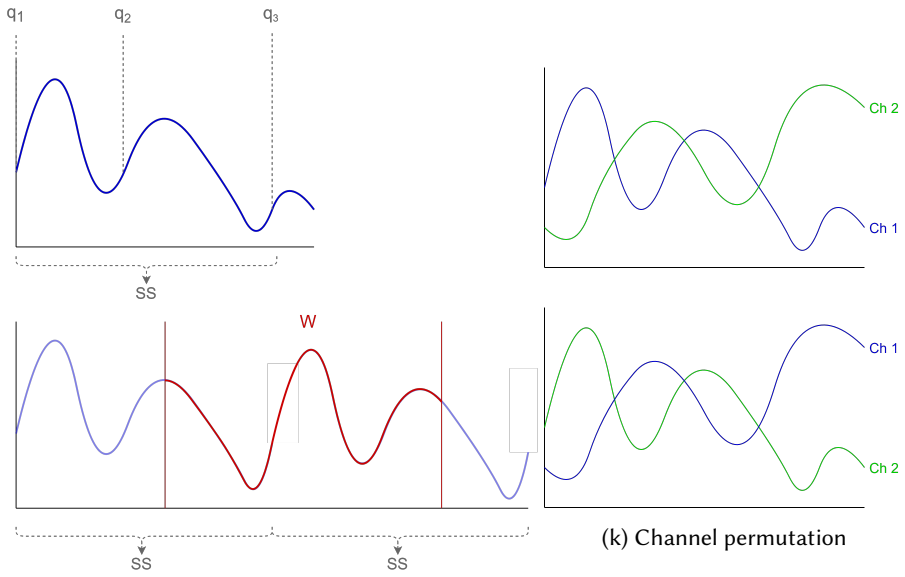
5.1.7 Summary of the traditional algorithms. The figure 4 shows an example of each algorithm reviewed:



(a) Ground truth (b) Homogeneous scaling (c) Rotation

(d) Jittering (e) Magnitude warping (f) Time warping

(g) Permutation (h) Window warping (i) Time Slicing Window



(j) Concatenating resampling

(k) Channel permutation

Fig. 4. Summary of traditional algorithms.

5.2 Data Augmentation through VAE

The use of AE architectures is nothing more than the evolution of data generation algorithms to produce more and better data, which means that, better, they are varied and therefore the standard deviation with respect to the original data is perfect. To precisely control the deviation of the data, VAE arises as the evolution of AE to generate better synthetic data, as shown in [4] where VAE is used to generate data for anomaly detection problems with LSTM. Or this other work [5], in which they use a dataset augmented with VAE to improve the recognition of human activity with LSTM. Even more exhaustive studies [26, 73], show the efficiency of these algorithms in increasing the size of datasets.

But the use of VAEs for DA is not only intended for neural network models, but can also improve results when traditional Machine Learning algorithms are applied [35]. However, they can also be used in applications with unsupervised training, that is, in [44], which applies them to unsupervised domain adaptation for robust speech recognition.

In [20] they point out that most data augmentation methods for time series use feature space transformations to artificially enlarge the training set; they propose a composition of autoencoders (AEs), variational autoencoders (VAEs) and Wasserstein generative adversarial networks with gradient penalty (WGAN-GPs) for time series augmentation.

In the end, each VAE model and its hyperparameter configuration make them specialise in the area or format of the dataset they want to work on; but above all to the type of problem for which it will be used afterwards. That is to say, what makes the difference between the models is what the generated data will be used for, problems such as: classification, forecasting, value imputation or prediction.

5.2.1 VAEs for anomaly detection. As mentioned before, VAEs are a DA architecture that has been widely used in the anomaly detection field. The main objective of using this models in anomaly detection tasks is to be able of generate data in order to avoid the lack of invalid data of the datasets. The most common scenario is that there is

not enough available anomalous samples in order to train machine learning models with the dataset, so the use of the VAE is focused on generating new data.

The work presented in [4] is centred in the classification of Electrocardiogram (ECG) signals, distinguishing between the ones with a cardiac dysfunction. In order to augment the data available it is used a Conditional VAE (CVAE)[92] which is able to learn which samples are normal and which are anomalous. This CVAE architecture is composed of LSTM layers[43] which process the temporal data of the ECG signals.

Other architecture based on the anomaly detection problem is the Smoothness-Inducing Sequential VAE (SISVAE)[68] which uses a VAE with recurrent layers to maintain temporal dependencies. This work focuses on the problem of abrupt changes between time steps, which leaded to non smooth reconstructions of the input data of the model, and therefore temporal abrupt changes in the synthesised samples. The mechanism to avoid this is to introduce a corrective bias for each time step of the signal, calculated using the KL Divergence between one point and the next one in the series.

5.2.2 VAEs for data imputation. One field where the VAE architecture has been widely used is in data imputation tasks. This process consists on generating new data in a sample where there is missing information. In the temporal series domain, this process is usually used to fill gaps in temporal spaces where there are not available data. In this sense, VAEs generates synthetic information on demand to fill this gaps, generating new information following the distribution of the original data.

The GlowImp architecture[69] was proposed as a combination of the Wasserstein GAN (WGAN)[7] architecture together with a VAE to impute missing data. The architecture is composed of the so-called Glow VAE, which incorporates a function that takes the latent distribution of the traditional VAE encoder and interpolates the missing values via the Glow Model. The other main part of the architecture is the GAN model where the generator corresponds with the decoder of the VAE and the discriminator forces the system to produce realistic samples. The architecture of the GlowImp can be seen in figure 5.

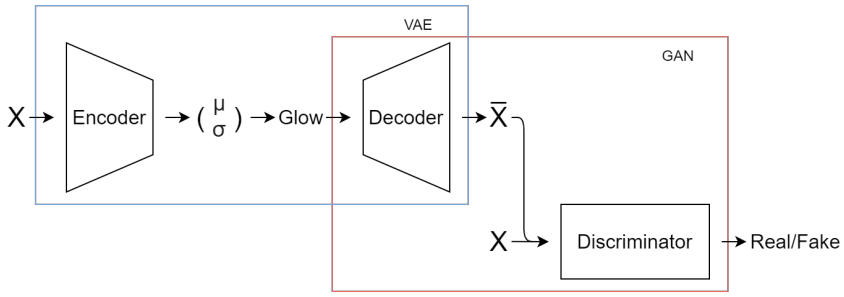


Fig. 5. GlowImp architecture. Based on the figure of [69].

The work of Li et al.[67] present a VAE architecture to impute temporal values using meteorologic datasets. In order to fill the missing values of the samples of data it is used shift correction, this correction tries to counteract the deviation caused by the missing values. This correction is used in the Gaussian latent distribution, where it is applied a shift hyperparameter λ which is set manually to center the latent distribution, thus correcting the possible bias produced by the missing values. The VAE architecture used in this work to impute the missing values is the β-VAE[41].

5.2.3 Taxonomy for the VAE algorithms reviewed. The figure 6 shows a scheme to group the different reviewed researches:

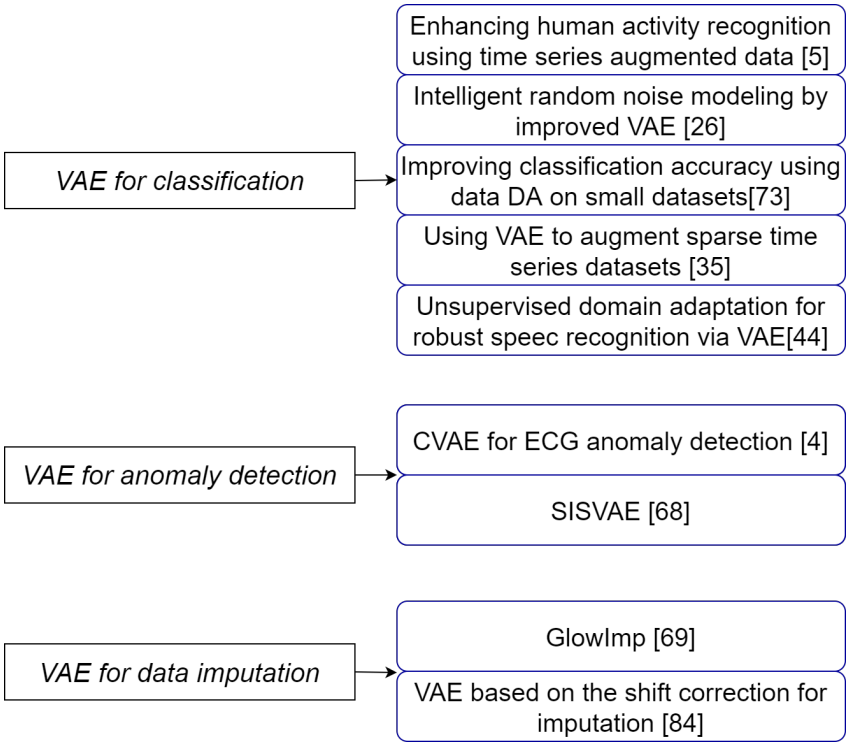


Fig. 6. Taxonomy of the presented VAE architectures.

5.3 Data Augmentation through GAN

GANs are one of the most popular generative models of the last decade, since its introduction in 2014 by Ian Goodfellow[34] this generative architecture has been established as one the main algorithms for DA. The main strength of the GAN architecture is that it learns the distribution of the data by extracting the main features of the samples, without copying the distribution directly. This empowers the generalisation and creativity of the synthetic data generated by the model. It is also an important factor that the training of the networks is unsupervised, not necessarily having labelled data to learn the distribution.

5.3.1 Long-Short Term Memory (LSTM) based GANs. One of the approximations of adapting the GAN architecture to time series is to use recurrent networks as the base of ANN. These GANs substitute the regular fully connected or convolutional layers by recurrent layers, capable of having memory that links temporal features of the data. The main strength of this set of architectures is that they are able of processing this temporal information that the input data have, similar to the spatial information processing of a convolutional neural network.

C-RNN-GAN[72] is one of the first GAN architectures proposed specifically for time series data. In particular, learning and synthesizing music tracks was proposed. This GANs uses LSTM blocks[43] as its main learning structure. The learning algorithm is the same as standard training GAN, where the network generator concatenates each input with the output of the previous cells and the discriminator is made up of a bidirectional recurrent network[89]. The internal composition of the discriminator is based on the work of Horchreiter[42] and Bengio et al.[10] that avoids the gradient vanishing and strengths the temporal dependencies.

The work presented by Haradal et al.[37] also proposes a GAN architecture based on the implementation of LSTM cells in both the generator and discriminator networks to adapt to time series data. The discriminator output is generated by applying an average pooling to the outputs generated by each layer, averaging the whole data sample into a unique scalar output which corresponds to the probability of the sample being generated by the generator network. This architecture was used to generate ECG[76] and Electroencephalogram (EEG)[6] data to improve the classification accuracy of a ANN classifier.

The LSTM and GAN combination has also been used for anomaly detection in the work of Zhu et al.[105] where the LSTM layers are used in the discriminator to extract temporal information from the data, while the GAN architecture provides the system with the ability to extract the most important features of the data. Training for detecting anomalies in the data has two phases. The first phase, known as the training phase, is a standard GAN training in which the discriminator learns to distinguish between real and synthetic data. In the second phase, so-called testing phase, the training consists of a feature extraction that generates and embedding of the dataset samples, these features are then reconstructed by the generator and compared with the original data, the task of the discriminator is to distinguish the real and the reconstructed data, which is anomalous.

The work presented by Shi et al.[90] uses the GAN architecture to generate sequences of faulty data from two different types, different models are trained for each type. The generator and discriminator of each GAN is composed by many-to-many LSTM model, which processes the voltage signal data and the sampling length of each step of the sequence. In this way, the generator output is composed of two vectors, one for the voltage and the other for the length, while the discriminator processes these data and its output is generated by averaging the classification of each step and generating an unique binary output.

5.3.2 Convolutional GANs applied to time domain series. In order to apply GANs to the time series domain, one of the most popular techniques used is to treat the time data as an image. Different approximations have been used in this field, where the focus is on how to transform the data into an image format, rather than adapting the GAN architecture to process time series information. One of the main advantages of this technique is that it does not have to deal with the design of GAN, which is a complex process due to the particularities of the architecture. The adaptation of the original data to an image is different in each case; different works published during the last years will be reviewed in order to study different approximations to this transformation.

An example of this use is the one proposed with SpecGAN[21] which tries to operate with sound spectograms that represent audio samples. This approach uses Deep Convolutional GAN (DCGAN)[81] as the main algorithm for DA, but prior to that, it processes the audio signal to generate images for each audio track. The process of transforming audio into image "can be approximately inverted" in the author's own words. First, a Fourier transform is applied to each audio to generate a matrix of the frequencies of the data. Then the scale of the data is adapted logarithmically and normalised to a normal distribution for a better understanding. Finally, the images are clipped to 3 standard deviations and rescaled within the $[-1, 1]$ range. As mentioned above, this process is reversible, so once the new data are generated using DCGAN they can be transformed to audio data using the reverse process. One advantage of using this process is that it opens up the possibility of comparing different audio generation algorithms treating the results as images; in the original paper, the results of the SpecGAN are compared with the WaveGAN, which is proposed in the same article.

The work presented by Jiang et al.[53] uses the GANomaly architecture[3] to process different time series data. The GANomally is used for anomaly detection in industrial tasks; it introduces a feature extraction into the network, which preprocesses the input data of both the generator and the generator. The generator is composed of an encoder-decoder-encoder network, which makes it possible to learn the latent representations generated by the feature extraction part. Regarding the data used for training, rolling bearing data was used to detect

anomalies, using signal data as spectograms, thus converting the time series data into the image domain. In particular, they used Bearing Data from Case Western Reserve University[1].

The Traffic Sensor Data Imputation GAN (TSDIGAN)[46] is an architecture proposed for missing data reconstruction, in particular traffic data is used. In this work, GAN is in charge of generating synthetic data that fill in the missing data gaps with realistic information. The approach used in the paper to treat time series traffic data is to transform them into an image format using the proposed method called Gramian Angular Summation Field (GASF). The GASF algorithm is focused on maintaining the time dependency of the traffic data; this algorithm is capable of transforming the data into a matrix by representing each time data point to polar coordination system, within the range $[-1, 1]$. Then, each point is encoded by its angular cosine and radius. This generates a matrix with the temporal correlation between each point, which is then fed to the networks. Finally, the data are processed using a convolutional based GAN that uses its generator to generate new data and reconstruct the missing values.

5.3.3 1D convolutional GANs. Temporal Convolutional Neural Networks (CNNs) are CNNs[65] where the convolutional operation is calculated in 1D instead of traditional 2D convolution. These networks adapt the geometric information captured by the 2D CNNs to a temporal domain, lowering the dimensions of the learnt filters to 1D. These networks have been used in works such as [50] to classify data from temporal series.

In recent years, different GAN architectures have been proposed that use these 1D convolutional layers as a base, replacing the traditional 2D convolutions of GANs applied to computer vision tasks. This approximation is very straightforward to adapt traditional GAN architectures to time series domain, making it very plausible for use in time series-related tasks.

The Temporal-Conditional GAN (T-CGAN)[83] is a GAN architecture based on the idea of transforming the Conditional GAN (CGAN)[71] architecture to time series domain by replacing the 2D convolutional layers with the 1D convolutional layers.

EmotionalGAN[15] also applies these 1D convolutional layers to create a GAN architecture to augment an ECG dataset improving the classification of Support Vector Machine (SVM) and Random Forest models when classifying the emotions of each subject.

The work published by Donahue et al.[21] presents the WaveGAN architecture, which is based on the application of 1D convolutional layers to sound data. This GAN uses the DCGAN architecture, but changes the convolutions to 1D. As suggested, these 1D convolutions should have a wider receptive field respecting the 2D convolutions of image processing; this is based on the particularities of the audio data, in which each cycle of a musical note sampled at 16kHz may take 36 samples to complete. Therefore, it is necessary to use wider filters to capture the distanced temporal dependencies of the data. This feature of the sound data was previously taken into account with solutions such as the dilated convolutions proposed in WaveNet[96]. This enlargement of the receptive field is compensated for by reducing one dimension, changing from 5x5 convolutions to 25 one-dimensional convolutions, maintaining the number of parameters of the network. The rest of the architecture maintains the standard GAN architecture, allowing the synthesis of audio tracks with unsupervised training GAN.

This approximation has also been followed by Sabir et al.[86] for augment DC current signals. The proposed work used the DCGAN architecture as a base and changes the original convolutions to 1D convolutions. In particular, this work has 2 different GANs, one that generates healthy signals and the other is in charge of generating faulty data.

There are also hybrid implementations that combine 1D convolutions with other techniques, such as in [105] where LSTM-GAN is proposed. This architecture combines the LSTM cell in the discriminator network with the 1D convolutional layers used in the generator network.

[1]https://engineering.case.edu/bearingdatacenter

5.3.4 Time series Generative Adversarial Networks (TimeGAN). The TimeGAN architecture[104] tries to implement a GAN model to perform DA on time series data, but differentiates itself from other previous alternatives by adding a new loss function that tries to capture the stepwise dependencies of the data. Previous implementations of GANs in data sequences were based on the use of recurrent networks for the generator and discriminator networks of GAN[23, 72], but this approximation may not be sufficient to accurately replicate the temporal transitions of the original data.

This work divides the data features into two different classes: static features and temporal features. Static features **S** do not vary over time, e.g., gender, while temporal features **X** change. In other words, the static features are characteristics of the data that are not directly related to the time series sample but define characteristics of it.

The proposed architecture adds, in addition to the generator and discriminator networks, two new networks: the encoder and recovery networks. These networks are responsible for embedding the input data in the latent space, as an autoencoder[60] would traditionally do. This system learns the so called embedding and recovery functions to take the static and temporal features to two separate latent codes $\mathbf{h}_s$ and $\mathbf{h}_t$ and recover the input information S and X.

The generator and discriminator part of the network does the same work as it would do in a traditional GAN, using the discriminator to differentiate between real and synthetic samples. But in this case, the generator generates the data for the embedding space, while the discriminator also takes this embedding as input for its classification.

The main innovation of TimeGAN is implemented in the generator, which, in addition to the normal generation of synthetic samples, is also forced to learn the stepwise dependencies of the data. To do so, the generator receives as input the synthetic embedding $\mathbf{h}_s, \mathbf{h}_{t-1}$ and computes the next vector $\mathbf{h}_s, \mathbf{h}_t$. This new function is learned by a new supervised loss function that compares the generator forecast with the real data.

Therefore, the training objectives of the presented architecture can be divided into 3 different loss functions.

- **Reconstruction loss** ($\mathcal{L}_R$): This loss is used in the reversible mapping part of the network, composed by the encoder and recovery networks. It can be denoted as follows:

$$\mathcal{L}_R = \mathbb{E}_{\mathbf{s},\mathbf{x}_{1:T}\sim p}[\|\mathbf{s} - \tilde{\mathbf{s}}\|_2 + \sum_t \|\mathbf{x}_t - \tilde{\mathbf{x}}_t\|_2] \tag{13}$$

  where the tilde denotes the reconstructed samples.
- **Unsupervised loss** ($\mathcal{L}_U$): This function is the equivalent loss function of a normal GAN that tries to distinguish real and fake samples. It is denoted as follows:

$$\mathcal{L}_{\mathrm{U}} = \mathbb{E}_{\mathrm{s},\mathrm{x}_{1:T}\sim p}[\log y_{\mathcal{S}} + \sum_t \log y_t] + \mathbb{E}_{\mathrm{s},\mathrm{x}_{1:T}\sim \hat{p}}[\log(1 - \hat{y}_{\mathcal{S}}) + \sum_t \log(1 - \hat{y}_t)] \tag{14}$$

  where y_s and y_t are the classification of the discriminator for static and temporal features and the accent denotes synthetic samples.
- **Supervised loss** ($\mathcal{L}_S$): To encourage the generator to learn the conditional transitions of the data, this function is designed that measures the similarity between the real and the synthetic samples created by the generator when applied the forecasting. The loss function is denoted as follows:

$$\mathcal{L}_{\mathrm{S}} = \mathbb{E}_{\mathrm{s},\mathrm{x}_{1:T}\sim p}[\sum_t \|\mathrm{h}_t - g_{\mathcal{X}}(\mathrm{h}_{\mathcal{S}}, \mathrm{h}_{t-1}, \mathrm{z}_t)\|_2] \tag{15}$$

  where $g_{\mathcal{X}}$ denotes the sample synthesised by the generator, taking as input the embedded anterior sample $\mathrm{h}_{\mathcal{S}}, \mathrm{h}_{t-1}, \mathrm{z}_t$.

An overview of the learning scheme of TimeGAN can be seen in figure 7.

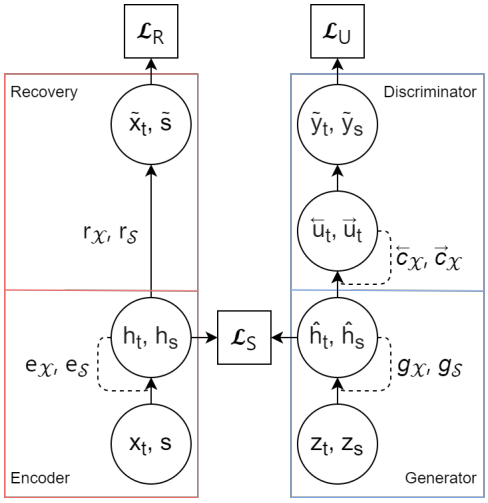


Fig. 7. TimeGAN architecture. Based on the figure of [104].

5.3.5 Taxonomy for the GAN algorithms reviewed. The figure 8 shows a scheme to group the different reviews of the researches:

5.4 Dynamic Time Warping (DTW) based on DA

5.4.1 DTW Barycenter Averaging. DTW[87] is a classical algorithm that measures the similarity between two data sequences. This method was used as a base in [29] to calculate the manifold of the original data. Once they have calculated this manifold, they use it to generate new data. The idea is to manipulate the manifold to generate infinite new samples of data, they achieve this by changing the weights of a set of time series, such as the set $D = \{(T_1, w_1), \ldots, (T_N, w_n)\}$ is embedded in a space E and the average of the DTW is denoted as follows:

$$\arg\min \bar{T} \in E \sum_{i=1}^{N} w_i \cdot DTW^2\left(\bar{T}, T_i\right) \tag{16}$$

where w is the weight of each sample.

To calculate $\bar{T}$ they use the Expectation-Maximization algorithm and to decide the weight values, three different methods are proposed:

- Average All: This method generates the weight vector values using a flat Dirichlet distribution. The main problem with this method is that it tends to fill in data spaces where it should not.
- Average Selected: This method focusses on selecting a subset of close samples. Thus, it prevents empty spaces from being filled with information because the subsets of samples are close together in the manifold.
- Average Selected with Distance: The difference between this method and the previous one is that this method calculates the relative distance between the near samples of data.

5.4.2 Suboptimal element alignment averaging. SuboPtimAl Warped time series geNEratoR (SPAWNER) is a DA method based on the DTW algorithm[87]. The DTW algorithm is used in this DA method to align different multidimensional signals X_1, X_2, giving the so called *warping path* W which is a sequence of points that minimises the distance between these input signals.

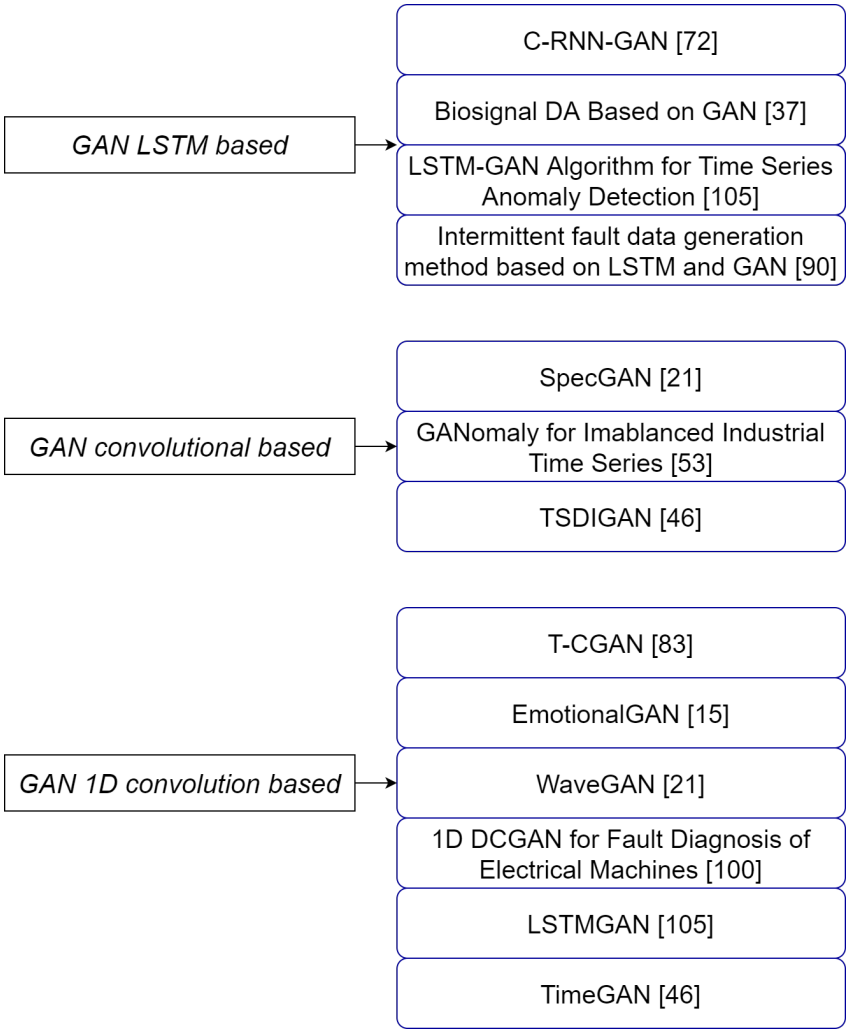


Fig. 8. Taxonomy of the presented GAN architectures.

The SPAWNER algorithm takes the warping path calculated with the DTW algorithm and introduces a new aleatory element to the sequence, known as w_p. This new point is generated using a uniformly distributed random number within the range $(0, 1)$. Then the new optimal path is forced to contain the new generated element, obtaining the new warping paths W_1^*, W_2^*. Both sequences are aligned using a parameter called ξ, which reduces the flexibility of the path. Finally, both warp paths are concatenated, generating the path $W_{1,2}^*$ from which the new time series signals X_1^*, X_2^* are obtained.

It is observed that for some multivariate signals, the variation of DA is not enough; therefore, a random variance is also applied to each point of the signal using a normal distribution such as $x^* \sim N(\mu, \sigma^2), \mu = 0.5(x_1^* + x_2^*), \sigma = 0.05|x_1^* - x_2^*|$.

The use of different alignment methods for text or image data is also proposed, instead of using DTW which is proposed when signals are used. Therefore, the overall algorithm can be easily translated to other domains, with the need for an alignment method between two samples.

6 OPEN ISSUES AND CHALLENGES

Some authors [99] tend to differentiate between DA and data generation due to the great advances produced in NN models. Traditional algorithms are always framed in the area of DA since the data they produce are always based on existing data; as an open problem, they generate less varied data but more control over what is generated. In addition, data generation algorithms produce new data so aggressively that much of the generated data is not possible, degenerating the quality of the augmented dataset [74].

Unlike the limitation of the scarcity of data generated with traditional models, AEs and VAEs are born to cover the deficiency of the generated data. In [31], they demonstrate the capability of generative NN models to add more diversity to the dataset. Additionally, traditional algorithms tend not to be flexible in taking a trained model and applying it to another problem, forcing a rethink of the algorithm. Neural networks, in this aspect, tend to be more flexible, able to use the same trained model in different problems. In [83], T-CGAN (Section 5.3.3) where different datasets are exposed with the same architecture, or in [105], LSTM-GAN that uses as inputs datasets as disparate as one of electrocardiograms and another of taxi statistics.

However, although generative models offer great advantages, GANs have significant additional problems, especially in training. Typical problems such as modal collapse, nash equilibria, gradient vanishing or instability are suffered in every training of these models, making their optimisation a very complex process [47].

In general, all generative models share the same open problem that often complicates their validation process. And, as shown in Section 4, despite the existence of some evaluation metrics, there is no consensus in the community on which should be used. For example, in [104] use empirical evaluation for data generation, for visualisation they use PCA and a discriminative and a predictive model to see how they have improved after adding the synthetic images, in [21] uses the Inception Score, a measure of Nearest Neighbour and empirical measurement by humans and in [46] use traditional measures of deep learning (MAE, RMSE and MRE) to compare the generation of future values. If we also focus on GAN models, we must take into account that to this problem we must add that there is no method for these architectures to define what the stop condition is in a training.

7 DISCUSSION

Data augmentation algorithms in the time series domain are really important for improving the available datasets, whose creation is not always easy. In general terms, all the methods presented in this work are algorithms specifically designed for DA in time series, but in other cases they are usually adaptations of architectures that were originally designed for other domains, such as image processing. However, the GAN-based algorithms themselves have their beginnings in the field of imaging and have gradually been integrated into other areas. In this section, we will analyse the main advantages and disadvantages for each type of algorithm.

7.1 Advantages

Traditional algorithms are widely developed and studied in such a way that their results can be fairly. In DA they allow you to work by modifying the examples already present, which allows you to control variations. In addition, the simplicity of the algorithms themselves by greatly reducing the number of hyperparameters to be configured results in less time to set them up and the need for fewer data to train them.

Secondly, the VAE generative algorithms allow to control to a greater extent the variability of the generated data by directly influencing the standard deviation of the latent distribution of the original dataset. This feature

allows, among all algorithms, the greatest control of the variability of the generated data. VAE is commonly used for anomaly detection cases due to its better performance.

Finally, the most current generative models are a breakthrough in the area due to their great results. GANs, like VAEs, allow synthetic data to be generated and, at the cost of losing some control over data generation, they are algorithms capable of much better generalisation. All this is due to the training scheme itself, which allows GANs models to learn the distribution that follows the original dataset and, through it, generate synthetic data according to the distribution of the dataset.

Furthermore, since GANs are relatively recent algorithms, they benefit from greater attention from the scientific community, which means that there is more recent research focused on improving their results than other algorithms.

7.2 Disadvantages

In terms of limitations, the use of traditional algorithms is quite limited because they are based on making modifications to elements of the real dataset. Therefore, they can often produce invalid examples. In general, they are limited to generate examples of lower quality and never to generate new elements.

Although VAEs are algorithms capable of generating synthetic data, as opposed to traditional algorithms that only modify the original data, new NN models such as GANs have mitigated their use in the field because by nature they are capable of generating fewer data than the most current generative networks. Despite this, because they can very precisely control the variability of the generated data, there are fields of application that still continue to use them.

Regarding GANs, it can be said that despite their great results, there are certain difficulties that slow their progress. GANs are by far the most complex models and, due to the particularities of the way they are trained, they are extremely difficult to train and obtain results.

However, due to their training based on the Nash equilibrium search, putting the Generator and the Discriminator networks in competition, once we have a trained model, half of the data it is able to generate must be discarded, which makes it necessary to control the generation of new data in a more supervised way.

8 CONCLUSION

Due to the significant evolution in the area of DA that has been undergone in recent years, more and more fields are beginning to appear in which to apply and improve the results that have been given. In this article, we focus on giving a comprehensive overview of the main algorithms used for DA in the field of time series. We have organised the review method in a taxonomy consisting of basic and advanced approaches, summarise representative methods in each algorithm (traditional, VAEs and GANs) to compare them empirically, disaggregate by application areas and highlight advantages/disadvantages for future research.

REFERENCES

[1] Mohammad Abufadda and Khalid Mansour. 2021. A Survey of Synthetic Data Generation for Machine Learning. In *2021 22nd International Arab Conference on Information Technology (ACIT)*. IEEE, 1–7.

[2] Seiji Adachi, Hironori Takemoto, Tatsuya Kitamura, Parham Mokhtari, and Kiyoshi Honda. 2007. Vocal tract length perturbation and its application to male-female vocal tract shape conversion. *The Journal of the Acoustical Society of America* 121, 6 (2007), 3874–3885.

[3] Samet Akcay, Amir Atapour-Abarghouei, and Toby P Breckon. 2018. Ganomaly: Semi-supervised anomaly detection via adversarial training. In *Asian conference on computer vision*. Springer, 622–637.

[4] Zabir Al Nazi, Ananna Biswas, Md Abu Rayhan, and Tasnim Azad Abir. 2019. Classification of ECG signals by dot residual LSTM network with data augmentation for anomaly detection. In *2019 22nd International Conference on Computer and Information Technology (ICCIT)*. IEEE, 1–5.

[5] Luay Alawneh, Tamam Alsarhan, Mohammad Al-Zinati, Mahmoud Al-Ayyoub, Yaser Jararweh, and Hongtao Lu. 2021. Enhancing human activity recognition using deep learning and time series augmented data. *Journal of Ambient Intelligence and Humanized*

Computing 12, 12 (2021), 10565–10580.

[6] Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger. 2001. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E* 64, 6 (2001), 061907.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*. PMLR, 214–223.

[8] Andri Ashfahani, Mahardhika Pratama, Edwin Lughofer, and Yew-Soon Ong. 2020. DEVDAN: Deep evolving denoising autoencoder. *Neurocomputing* 390 (2020), 297–314.

[9] Kasun Bandara, Hansika Hewamalage, Yuan-Hao Liu, Yanfei Kang, and Christoph Bergmeir. 2021. Improving the accuracy of global forecasting models using time series data augmentation. *Pattern Recognition* 120 (2021), 108148.

[10] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks* 5, 2 (1994), 157–166.

[11] Ali Borji. 2019. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding* 179 (2019), 41–65.

[12] Eoin Brophy, Zhengwei Wang, Qi She, and Tomas Ward. 2021. Generative adversarial networks in time series: A survey and taxonomy. *arXiv preprint arXiv:2107.11098* (2021).

[13] Ping Cao, Xinyi Li, Kedong Mao, Fei Lu, Gangmin Ning, Luping Fang, and Qing Pan. 2020. A novel data augmentation method to enhance deep neural networks for detection of atrial fibrillation. *Biomedical Signal Processing and Control* 56 (2020), 101675.

[14] Elsa Cazelles, Arnaud Robert, and Felipe Tobar. 2020. The Wasserstein-Fourier distance for stationary time series. *IEEE Transactions on Signal Processing* 69 (2020), 709–721.

[15] Genlang Chen, Yi Zhu, Zhiqing Hong, and Zhen Yang. 2019. EmotionalGAN: Generating ECG to enhance emotion state classification. In *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*. 309–313.

[16] Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. 2021. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology* 65, 5 (2021), 545–563.

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

[18] Hoang Anh Dau, Eamonn Keogh, Kaveh Kamgar, Chin-Chia Michael Yeh, Yan Zhu, Shaghayegh Gharghabi, Chotirat Ann Ratanamahatana, Yanping, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, Gustavo Batista, and Hexagon-ML. 2018. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

[19] Ankur Debnath, Govind Waghmare, Hardik Wadhwa, Siddhartha Asthana, and Ankur Arora. 2021. Exploring Generative Data Augmentation in Multivariate Time Series Forecasting: Opportunities and Challenges. *Solar-Energy* 137 (2021), 52–560.

[20] Sumeyra Demir, Krystof Mincev, Koen Kok, and Nikolaos G. Paterakis. 2021. Data augmentation for time series regression: Applying transformations, autoencoders and adversarial networks to electricity price forecasting. *Applied Energy* 304 (2021), 117695. https://doi.org/10.1016/j.apenergy.2021.117695

[21] Chris Donahue, Julian McAuley, and Miller Puckette. 2018. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208* (2018).

[22] Huu-Thanh Duong and Tram-Anh Nguyen-Thi. 2021. A review: preprocessing techniques and data augmentation for sentiment analysis. *Computational Social Networks* 8, 1 (2021), 1–16.

[23] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633* (2017).

[24] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2018. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv preprint arXiv:1808.02455* (2018).

[25] Ebubeogu Amarachukwu Felix and Sai Peck Lee. 2019. Systematic literature review of preprocessing techniques for imbalanced data. *IET Software* 13, 6 (2019), 479–496.

[26] Qiankun Feng, Yue Li, and Hongzhou Wang. 2021. Intelligent random noise modeling by the improved variational autoencoding method and its application to data augmentation. *Geophysics* 86, 1 (2021), T19–T31.

[27] Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075* (2021).

[28] Anibal Flores, Hugo Tito-Chura, and Honorio Apaza-Alanoca. 2021. Data augmentation for short-term time series prediction with deep learning. In *Intelligent Computing*. Springer, 492–506.

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

[30] Biying Fu, Florian Kirchbuchner, and Arjan Kuijper. 2020. Data augmentation for time series: traditional vs generative models on capacitive proximity time series. In *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*. 1–10.

[31] Biying Fu, Florian Kirchbuchner, and Arjan Kuijper. 2020. Data Augmentation for Time Series: Traditional vs Generative Models on Capacitive Proximity Time Series. In *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to*

Assistive Environments (Corfu, Greece) *(PETRA '20)*. Association for Computing Machinery, New York, NY, USA, Article 16, 10 pages. https://doi.org/10.1145/3389189.3392606

[32] Lianli Gao, Daiyuan Chen, Zhou Zhao, Jie Shao, and Heng Tao Shen. 2021. Lightweight dynamic conditional GAN with pyramid attention for text-to-image synthesis. *Pattern Recognition* 110 (2021), 107384.

[33] Nan Gao, Hao Xue, Wei Shao, Sichen Zhao, Kyle Kai Qin, Arian Prabowo, Mohammad Saiedur Rahaman, and Flora D Salim. 2022. Generative adversarial networks for spatio-temporal data: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 2 (2022), 1–25.

[34] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. arXiv:1406.2661 [stat.ML]

[35] Maxime Goubeaud, Philipp Joußen, Nicolla Gmyrek, Farzin Ghorban, Lucas Schelkes, and Anton Kummert. 2021. Using Variational Autoencoder to augment Sparse Time series Datasets. In *2021 7th International Conference on Optimization and Applications (ICOA)*. IEEE, 1–6.

[36] Swaminathan Gurumurthy, Ravi Kiran Sarvadevabhatla, and Venkatesh Babu Radhakrishnan. 2017. DeLiGAN : Generative Adversarial Networks for Diverse and Limited Data. arXiv:1706.02071 [cs.CV]

[37] Shota Haradal, Hideaki Hayashi, and Seiichi Uchida. 2018. Biosignal data augmentation based on generative adversarial networks. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 368–371.

[38] Kay Gregor Hartmann, Robin Tibor Schirrmeister, and Tonio Ball. 2018. EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals. *arXiv preprint arXiv:1806.01875* (2018).

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[40] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

[41] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. (2016).

[42] Sepp Hochreiter. 1998. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6, 02 (1998), 107–116.

[43] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[44] Wei-Ning Hsu, Yu Zhang, and James Glass. 2017. Unsupervised domain adaptation for robust speech recognition via variational autoencoder-based data augmentation. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 16–23.

[45] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[46] Tongge Huang, Pranamesh Chakraborty, and Anuj Sharma. 2021. Deep convolutional generative adversarial networks for traffic data imputation encoding time series as images. *International Journal of Transportation Science and Technology* (2021).

[47] Guillermo Iglesias, Edgar Talavera, and Alberto Díaz-Álvarez. 2022. A survey on GANs for computer vision: Recent research, analysis and taxonomy. https://doi.org/10.48550/ARXIV.2203.11242

[48] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.

[49] Brian Kenji Iwana and Seiichi Uchida. 2021. An empirical survey of data augmentation for time series classification with neural networks. *Plos one* 16, 7 (2021), e0254841.

[50] Brian Kenji Iwana and Seiichi Uchida. 2021. An empirical survey of data augmentation for time series classification with neural networks. *Plos one* 16, 7 (2021), e0254841.

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

[52] Chi Yoon Jeong, Hyung Cheol Shin, and Mooseop Kim. 2021. Sensor-data augmentation for human activity recognition with time-warping and data masking. *Multimedia Tools and Applications* 80, 14 (2021), 20991–21009.

[53] Wenqian Jiang, Yang Hong, Beitong Zhou, Xin He, and Cheng Cheng. 2019. A GAN-based anomaly detection approach for imbalanced industrial time series. *IEEE Access* 7 (2019), 143608–143619.

[54] Fazle Karim, Somshubra Majumdar, Houshang Darabi, and Shun Chen. 2017. LSTM fully convolutional networks for time series classification. *IEEE access* 6 (2017), 1662–1669.

[55] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017).

[56] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-free generative adversarial networks. *Advances in Neural Information Processing Systems* 34 (2021).

[57] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4401–4410.

[58] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. arXiv:1312.6114 [stat.ML]
[59] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. 2015. Audio augmentation for speech recognition. In *Sixteenth annual conference of the international speech communication association.*
[60] Mark A Kramer. 1991. Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal* 37, 2 (1991), 233–243.
[61] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).
[62] Elnaz Lashgari, Dehua Liang, and Uri Maoz. 2020. Data augmentation for deep-learning-based electroencephalography. *Journal of Neuroscience Methods* 346 (2020), 108885.
[63] Arthur Le Guennec, Simon Malinowski, and Romain Tavenard. 2016. Data augmentation for time series classification using convolutional neural networks. In *ECML/PKDD workshop on advanced analytics and learning on temporal data.*
[64] Yann Lecun. 1987. *PhD thesis: Modeles connexionnistes de l'apprentissage (connectionist learning models).* Universite P. et M. Curie (Paris 6).
[65] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.
[66] H Lee, SJ Hwang, and J Shin. 2019. Rethinking data augmentation: Self-supervision and self-distillation. arXiv 2019. *arXiv preprint arXiv:1910.05872* (2019).
[67] Junying Li, Weijie Ren, and Min Han. 2021. Variational auto-encoders based on the shift correction for imputation of specific missing in multivariate time series. *Measurement* 186 (2021), 110055.
[68] Longyuan Li, Junchi Yan, Haiyang Wang, and Yaohui Jin. 2020. Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder. *IEEE transactions on neural networks and learning systems* 32, 3 (2020), 1177–1191.
[69] Caizheng Liu, Houquan Zhou, Zhi Sun, and Guangfan Cui. 2021. GlowImp: combining GLOW and GAN for multivariate time series imputation. In *International Conference on Algorithms and Architectures for Parallel Processing*. Springer, 50–64.
[70] Agnieszka Mikołajczyk and Michał Grochowski. 2018. Data augmentation for improving deep learning in image classification problem. In *2018 international interdisciplinary PhD workshop (IIPhDW)*. IEEE, 117–122.
[71] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).
[72] Olof Mogren. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904* (2016).
[73] Francisco J Moreno-Barea, José M Jerez, and Leonardo Franco. 2020. Improving classification accuracy using data augmentation on small data sets. *Expert Systems with Applications* 161 (2020), 113696.
[74] Jakub Nalepa, Michal Marcinkiewicz, and Michal Kawulok. 2019. Data augmentation for brain-tumor segmentation: a review. *Frontiers in computational neuroscience* 13 (2019), 83.
[75] Humza Naveed. 2021. Survey: Image mixing and deleting for data augmentation. *arXiv preprint arXiv:2106.07085* (2021).
[76] Robert Thomas Olszewski. 2001. *Generalized feature extraction for structural pattern recognition in time-series data.* Carnegie Mellon University.
[77] Jiapu Pan and Willis J Tompkins. 1985. A real-time QRS detection algorithm. *IEEE transactions on biomedical engineering* 3 (1985), 230–236.
[78] Qing Pan, Xinyi Li, and Luping Fang. 2020. Data augmentation for deep learning-based ECG analysis. In *Feature engineering and computational intelligence in ECG monitoring*. Springer, 91–111.
[79] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779* (2019).
[80] Abhishek G Patil, AR Surve, Anil Kumar Gupta, Amarjeet Sharma, and Sweta Anmulwar. 2016. Survey of synthetic traffic generators. In *2016 International Conference on Inventive Computation Technologies (ICICT)*, Vol. 1. IEEE, 1–3.
[81] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).
[82] Trivellore E Raghunathan. 2021. Synthetic data. *Annual Review of Statistics and Its Application* 8 (2021), 129–140.
[83] Giorgia Ramponi, Pavlos Protopapas, Marco Brambilla, and Ryan Janssen. 2018. T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. *arXiv preprint arXiv:1811.08295* (2018).
[84] Khandakar M Rashid and Joseph Louis. 2019. Window-warping: a time series data augmentation of IMU data for construction equipment activity identification. In *ISARC. Proceedings of the International Symposium on Automation and Robotics in Construction*, Vol. 36. IAARC Publications, 651–657.
[85] Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799* (2017).
[86] Russell Sabir, Daniele Rosato, Sven Hartmann, and Clemens Gühmann. 2021. Signal Generation using 1d Deep Convolutional Generative Adversarial Networks for Fault Diagnosis of Electrical Machines. In *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 3907–3914.

[87] Hiroaki Sakoe and Seibi Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing* 26, 1 (1978), 43–49.
[88] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. 2016. Improved Techniques for Training GANs. arXiv:1606.03498 [cs.LG]
[89] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45, 11 (1997), 2673–2681.
[90] Junyou Shi, Yufei Ding, and Zhenyang Lv. 2021. An Intermittent Fault Data Generation Method Based on LSTM and GAN. In *2021 Global Reliability and Prognostics and Health Management (PHM-Nanjing)*. IEEE, 1–4.
[91] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).
[92] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. 2015. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems* 28 (2015).
[93] Ilya Sutskever, James Martens, and Geoffrey E Hinton. 2011. Generating text with recurrent neural networks. In *ICML*.
[94] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1–9.
[95] Terry T Um, Franz MJ Pfister, Daniel Pichler, Satoshi Endo, Muriel Lang, Sandra Hirche, Urban Fietzek, and Dana Kulić. 2017. Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. In *Proceedings of the 19th ACM international conference on multimodal interaction*. 216–220.
[96] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W Senior, and Koray Kavukcuoglu. 2016. WaveNet: A generative model for raw audio. *SSW* 125 (2016), 2.
[97] Lidan Wang, Vishwanath Sindagi, and Vishal Patel. 2018. High-quality facial photo-sketch synthesis using multi-adversarial networks. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*. IEEE, 83–90.
[98] Shuo Wang, Carsten Rudolph, Surya Nepal, Marthie Grobler, and Shangyu Chen. 2020. PART-GAN: Privacy-Preserving Time-Series Sharing. In *International Conference on Artificial Neural Networks*. Springer, 578–593.
[99] Xiang Wang, Kai Wang, and Shiguo Lian. 2020. A survey on face data augmentation for the training of deep neural networks. *Neural computing and applications* 32, 19 (2020), 15503–15531.
[100] Zhiguang Wang, Weizhong Yan, and Tim Oates. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*. IEEE, 1578–1585.
[101] Zhiguang Wang, Weizhong Yan, and Tim Oates. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International joint conference on neural networks (IJCNN)*. IEEE, 1578–1585.
[102] Chao-Han Huck Yang, Yun-Yun Tsai, and Pin-Yu Chen. 2021. Voice2series: Reprogramming acoustic models for time series classification. In *International Conference on Machine Learning*. PMLR, 11808–11819.
[103] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*. 2849–2857.
[104] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. 2019. Time-series generative adversarial networks. *Advances in Neural Information Processing Systems* 32 (2019).
[105] Guangxuan Zhu, Hongbo Zhao, Haoqiang Liu, and Hua Sun. 2019. A novel LSTM-GAN algorithm for time series anomaly detection. In *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*. IEEE, 1–6.
[106] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*. 2223–2232.
[107] Minfeng Zhu, Pingbo Pan, Wei Chen, and Yi Yang. 2019. Dm-gan: Dynamic memory generative adversarial networks for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5802–5810.