engineering proceedings

MDPI

Proceeding Paper

If You Like It, GAN It—Probabilistic Multivariate Times Series Forecast with GAN †

Alireza Koochali [1,2,3,*], Andreas Dengel [2,3] and Sheraz Ahmed [2]

[1] Ingenieurgesellschaft Auto und Verkehr (IAV) GmbH, 10587 Berlin, Germany
[2] Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI) GmbH, 67663 Kaiserslautern, Germany; andreas.dengel@dfki.de (A.D.); sheraz.ahmed@dfki.de (S.A.)
[3] Department of Computer Science, University of Kaiserslautern, 67663 Kaiserslautern, Germany
* Correspondence: alireza.koochali@iav.de
† Presented at the 7th International Conference on Time Series and Forecasting, Gran Canaria, Spain, 19–21 July 2021.

Abstract: The contribution of this paper is two-fold. First, we present ProbCast—a novel probabilistic model for multivariate time-series forecasting. We employ a conditional GAN framework to train our model with adversarial training. Second, we propose a framework that lets us transform a deterministic model into a probabilistic one with improved performance. The motivation of the framework is to either transform existing highly accurate point forecast models to their probabilistic counterparts or to train GANs stably by selecting the architecture of GAN's component carefully and efficiently. We conduct experiments over two publicly available datasets—an electricity consumption dataset and an exchange-rate dataset. The results of the experiments demonstrate the remarkable performance of our model as well as the successful application of our proposed framework.

Keywords: time-series; generative adversarial networks; forecasting; probabilistic; prediction

Citation: Koochali, A.; Dengel, A.; Ahmed, S. If You Like It, GAN It—Probabilistic Multivariate Times Series Forecast with GAN. *Eng. Proc.* **2021**, *5*, 40. https://doi.org/10.3390/engproc2021005040

Academic Editors: Ignacio Rojas, Fernando Rojas, Luis Javier Herrera and Hector Pomare

Published: 8 July 2021

1. Introduction

Many sectors, such as health care, the automotive industry, the aerospace industry and weather forecasting, deal with time-series data in their operations. Knowledge about what will happen in the future is essential for making genuine decisions, and accurately forecasting future values is key to their success. A huge body of research is therefore dedicated to addressing the forecasting problem. An overview of various studies on the forecasting problem is provided in [1]. Currently, the field is dominated by point prediction methods, which are easy to understand. However, these deterministic models report the mean of possible outcomes and cannot reflect the inherent uncertainty that exists in the real world. The probabilistic forecast models are devised to rectify these shortcomings. These models try to quantify the uncertainty of the predictions by forming a probability distribution over possible outcomes [2].

In this paper, we propose ProbCast, a new probabilistic forecast model for multivariate time series based on Conditional Generative Adversarial Networks (GANs). Conditional GANs are a class of NN-based generative models that enable us to learn conditional probability distribution given a dataset. ProbCast is trained using a Conditional GAN setup to learn the probability distribution of future values conditioned on the historical information of the signal.

While GANs are powerful methods for learning complex probability distributions, they are notoriously hard to train. The training process is very unstable and quite dependent on careful selection of the model architecture and hyperparameters [3]. In addition to ProbCast, we suggest a framework for transforming an existing deterministic forecaster—which is comparatively easy to train—into a probabilistic one that exceeds the performance of its predecessor. By using the proposed framework, the space for searching the GAN's

architecture becomes considerably smaller. Thus, this framework provides an easy way to adapt highly accurate deterministic models to construct useful probabilistic models, without compromising the accuracy, by exploiting the potential of GANs.

In summary, the main contributions of this article are as follows:

- We introduce ProbCast, a novel probabilistic model for multivariate time-series forecasting. Our method employs a conditional GAN setup to train a probabilistic forecaster.
- We suggest a framework for transforming a point forecast model into a probabilistic model. This framework eases the process of replacing the deterministic model with probabilistic ones.
- We conduct experiments on two publicly available datasets and report the results, which show the superiority of ProbCast. Furthermore, we demonstrate that our framework is capable of transforming a point forecast method into a probabilistic model with improved accuracy.

2. Related Work

Due to the lack of a standard evaluation method for GANs, initially they were applied to domains in which their results are intuitively assessable, for example, images. However, recently, they have been applied to time-series data. Currently, GANs are applied to various domains for generating realistic time-series data including health care [4–8], finance [9,10] and the energy industry [11–13]. In [14], the authors combine GAN and auto-regressive models to improve sequential data generation. Ramponi et al. [15] condition a GAN on timestamp information to handle irregular sampling.

Furthermore, researchers have used conditional GANs to build probabilistic forecasting models. Koochali et al. [16] used a Conditional GAN to build a probabilistic model for univariate time-series. They used Long Short Term Memory (LSTM) in the GAN's component and tested their method on a synthetic dataset as well as on two publicly available datasets. In [17], the authors utilized LSTM and Multi-Layer Perceptron (MLP) in a conditional GAN structure to forecast the daily closing price of stocks. The authors combined the Mean Square Error (MSE) with the generator loss of a GAN to improve performance. Zhou et al. [18] employed LSTM and a convolutional neural network (CNN) in an adversarial training setup to forecast the high-frequency stock market. To guarantee satisfying predictions, this method minimizes the forecast error in the form of Mean absolute error (MAE) or MSE during training in conjunction with the GAN value function. Lin et al. [19] proposed a pattern sensitive forecasting model for traffic flow, which can provide accurate predictions in unusual states without compromising its performance in its usual states. This method uses conditional GAN with MLP in its structure and adds two error terms to the standard generator loss. The first term specifies forecast error and the second term expresses reconstruction error. Kabir et al. [20] make use of adversarial training for quantifying the uncertainty of the electricity price with a prediction interval. This line of research is more aligned with the method we presented in this article; however, the methods suggested in [17–19] include a point-wise loss function in the GAN loss function. Minimizing suggested loss functions would decrease the statistical error values such as RMSE, MAPE and MSE. However, they encourage the model to learn the mean of possible outcomes instead of the probability distribution of future value. Hence, their probabilistic forecast can be misleading despite the small statistical error.

3. Background

Here, we work with a multivariate time-series $X = \{X_0, X_1, ..., X_T\}$, where each $X_t = \{x_{t,1}, x_{t,2}, ..., x_{t,f}\}$ is a vector with size f equal to the number of features. In this paper, $x_{t,f}$ refers to the data point at time step t of feature f and X_t points to the feature vector at time step t. The goal is to model $P(X_{t+1}|X_t, .., X_0)$, the probability distribution for X_{t+1} given historical information $\{X_t, .., X_0\}$.

3.1. Mean Regression Forecaster

To address the problem of forecasting, we can take the predictive view of regression [2]. Ultimately, the regression analysis aims to learn the conditional distribution of a response given a set of explanatory variables [21]. The mean regression methods are deterministic methods, which are concerned with accurately predicting the mean of the possible outcome, that is, $\mu(P(X_{t+1}|X_t, .., X_0))$. There is a broad range of mean regression methods available in the literature; however, all of them are unable to reflect uncertainty in their forecasts. Hence, their results can be unreliable and misleading in some cases.

3.2. Generative Adversarial Network

In 2014, Goodfellow et al. [22] introduced a powerful generative model called the Generative Adversarial Network (GAN). GAN can implicitly learn probability distribution, which describes a given dataset, that is, $P(data)$ with high precision. Hence, it is capable of generating artificial samples with high fidelity. The GAN architecture is composed of two neural networks, namely generator and discriminator. These components are trained simultaneously in an adversarial process. In the training process, first, a noise vector z is sampled from a known probability distribution $P_{noise}(z)$ and is fed into a generator. Then, the generator transforms z from $P_{noise}(z)$ to a sample, which follows P_{data}. On the other hand, the discriminator checks how well the generator is performing by comparing the generator's outputs with real samples from the dataset. During training, this two-player minimax game is set in motion by optimizing the following value function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data}(x)}[log(D(x))] + \mathbb{E}_{z\sim P_{noise}(z)}[log(1-D(G(z)))]. \quad (1)$$

However, GAN's remarkable performance is not acquired easily. The training process is quite unstable and the careful selection of GAN's architecture and hyperparameters is vital for stabilizing the training process [3]. Since we should search for the optimal architecture of the generator and discriminator simultaneously, it is normally a cumbersome and time-consuming task to find a perfect combination of structures in a big search space.

3.3. Conditional GAN

Conditional GAN [23] enables us to incorporate auxiliary information, called the condition, into the process of data generation. In this method, we provide an extra piece of information, such as labels, to both the generator and the discriminator. The generator must respect the condition while synthesizing a new sample because the discriminator considers the given condition while it checks the authenticity of its input. The new value function $V(G, D)$ for this setting is:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim P_{data}(x)}[log(D(x|y))] + \mathbb{E}_{z\sim P_{noise}(z)}[log(1-D(G(z|y)))]. \quad (2)$$

After training a Conditional GAN, the generator learns implicitly the probability distribution of the given condition of the data, that is, $P(data|condition)$.

4. Methodology

4.1. ProbCast: The Proposed Multivariate Forecasting Model

In this article, we consider Conditional GAN as a method for training a probabilistic forecast model using adversarial training. In this perspective, the generator is our probabilistic model (i.e., ProbCast) and the discriminator provides the required gradient for optimizing ProbCast during training. To learn $P(X_{t+1}|X_t, .., X_0)$, the historical information $\{X_t, .., X_0\}$ is used as the condition of our Conditional GAN and the generator is trained to generate X_{t+1}. Hence, the probability distribution, which is learned by the generator,

corresponds to $P(X_{t+1}|X_t, .., X_0)$, that is, our target distribution. The value function, which we used for training the ProbCast (indicated as PC), is:

$$\min_{PC} \max_{D} V(D, PC) = \mathbb{E}_{X_{t+1} \sim P_{\text{data}}(X_{t+1})}[\log D(X_{t+1}|X_t, .., X_0)] + \mathbb{E}_{z \sim P_z(z)}[\log (1 - D(PC(z|X_t, .., X_0)))]. \quad (3)$$

4.2. The Proposed Framework for Converting Deterministic Model to Probabilistic Model

By stepping into the realm of multivariate time-series, other challenges also need to be addressed. In the multivariate setting, we require more complicated architecture to figure out dependencies between features and to forecast the future with high accuracy. Furthermore, as previously mentioned, GANs require precise hyperparameter tuning to have a stable training process. Considering required network complexity for handling multivariate time-series data, it is very cumbersome, or in some cases impossible, to find suitable generator and discriminator architecture concurrently which performs accurately. To address this problem, we propose a new framework for building a probabilistic forecaster based on a deterministic forecaster using GAN architecture.

In this framework, we build the generator based on the architecture and hyperparameters of the deterministic forecaster and train it using appropriate discriminator architecture. In this fashion, we can perform the task of finding an appropriate generator and discriminator architecture separately, which results in simplification of the GAN architecture search process. In other words, by using this framework, we can transform an existing accurate deterministic model into a probabilistic model with increased precision and better alignment with the real world.

4.3. Train Pipeline

Figure 1 demonstrates the proposed framework, as well as the conditional GAN setup, for training ProbCast. First, we build an accurate point forecast model by searching for the optimal architecture of the deterministic model. In the case that a precise point forecast model exists, we can skip the first step and use an existing model. Then, we need to integrate the noise vector z into the deterministic model architecture. In our experiments, we obtain the best results when we insert the noise vector into the later layers of the network, letting earlier layers of the network learn the representation of the input window. Finally, we train this model using adversarial training to acquire our probabilistic forecast model, that is, ProbCast.

With the generator architecture at hand, we only need to search for an appropriate time-series classifier to serve as the discriminator during the training of GAN. By reducing the search space of GAN architecture to the discriminator only, we can efficiently find a discriminator structure that is capable of training the ProbCast with a superior performance in comparison to the deterministic model. The following steps summarize the framework:

1. Employ an accurate deterministic model;
   (a) Either use an existing model;
   (b) Or search for an optimal deterministic forecaster;
2. Structure the generator based on deterministic model architecture and incorporate the noise vector into the network, preferably into later layers;
3. Search for an optimal discriminator structure and train the generator using it.

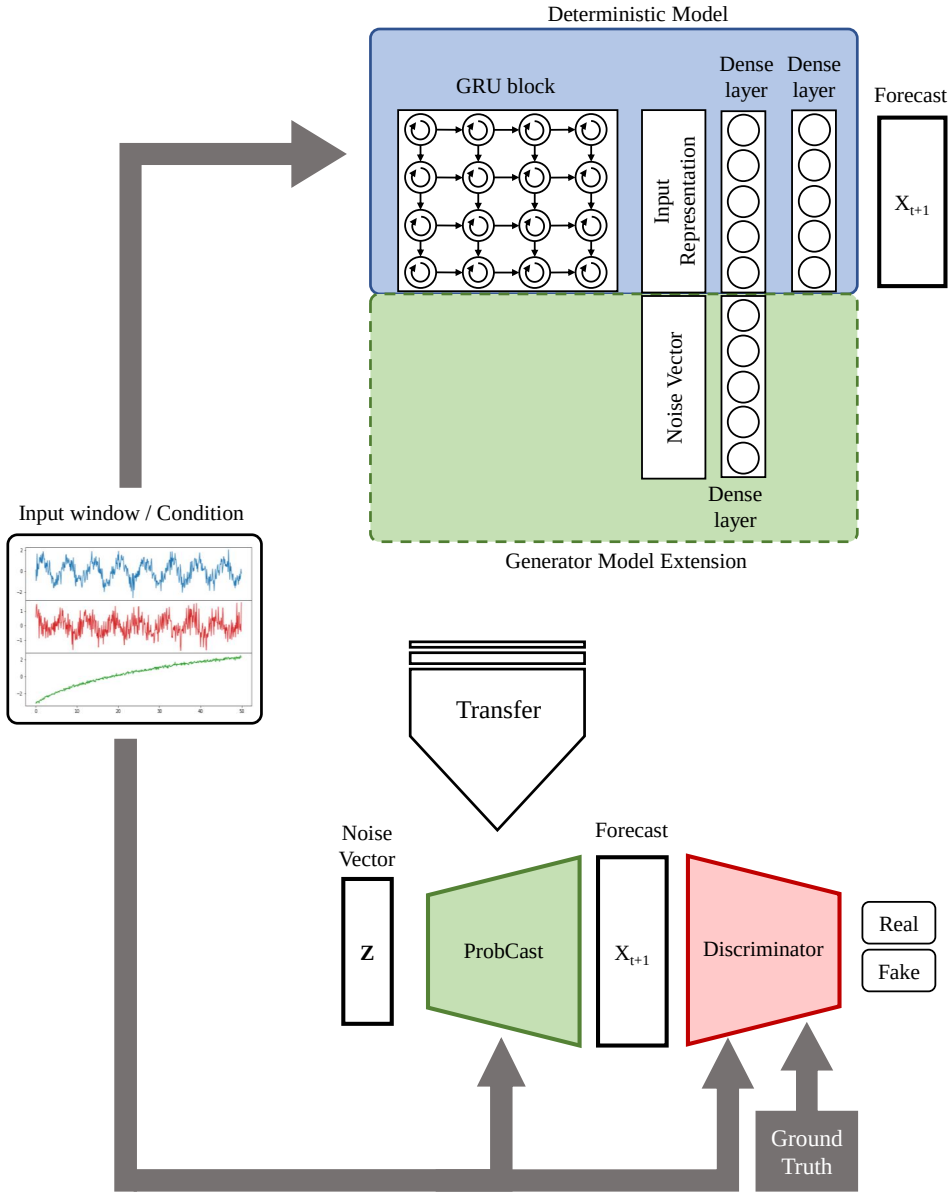

Figure 1. The demonstration of the proposed framework and adversarial training setup. The pipeline is followed from top to bottom. First, we search for the optimal architecture of the deterministic model. The deterministic model consists of a GRU block for the learning input window representation and two dense layers to map the representation to the forecast. Then, the noise vector z is integrated into the deterministic model to build the generator. Finally, the generator is trained using a suitable discriminator in a conditional GAN setup to obtain ProbCast.

5. Experiment

5.1. Datasets

We tested our method on two publicly available datasets—electricity and exchange-rate datasets (we used datasets from https://github.com/laiguokun/multivariate-time-series-data as they were prepared by the authors of [24]). The electricity dataset consists of the electricity consumption of 321 clients in KWh, which was collected every 15 min between 2012 and 2014. The dataset was converted to reflect hourly consumption. The exchange-rate dataset contains the daily exchange-rate of eight countries, namely Australia, Great Britain, Canada, Switzerland, China, Japan, New Zealand, and Singapore, which was collected between 1990 and 2016. Table 1 lists the properties of these two datasets.

For our experiments, we used 75% of the datasets for training, 5% for validation and 20% for testing.

Table 1. The properties of datasets.

	Electricity Dataset	Exchange-Rate Dataset
Dataset length	62,304	7588
Number of feature	321	8
Sample rate	1 h	1 day

5.2. Setup

In each of our experiments, first we ran an architecture search to find an accurate deterministic model. For training the deterministic model, we employed MAE as a loss function. In Figure 1, the architecture of the deterministic model is indicated. We used a Gated recurrent unit (GRU) [25] block to learn the representation of the input window. Then, the representation passed through two dense layers to map from the representation to the forecast. We adopted the architecture of the most precise deterministic model, which we found in order to build the ProbCast by concatenating the noise vector to the GRU block output (i.e., representation) and extending the MLP block as shown in Figure 1. Finally, we searched for the optimal architecture of the discriminator (Figure 2) and trained the ProbCast. The discriminator concatenated X_{t+1} to the end of the input window and constructed $\{X_{t+1}, X_t, .., X_0\}$. Then it utilized a GRU block followed by two layers of MLP to inspect the consistency of this window. We used the genetic algorithm to search for the optimal architecture. We coded our method using Pytorch [26].

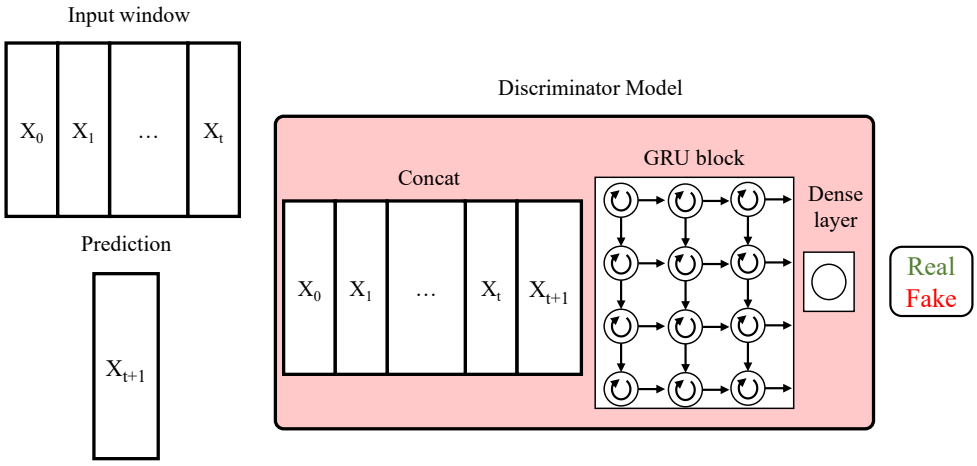

Figure 2. The discriminator architecture of our conditional GAN. The number of layers and cells in the GRU block are hyperparameters.

5.3. Evaluation Metric

To report the performance of the ProbCast, we used the negative form of the Continuous Ranked Probability Score [27] (denoted by $CRPS^*$) as the metric. The $CRPS^*$ reflects the sharpness and calibration of a probabilistic method. It is defined as follows:

$$CRPS^*(F, x) = E_F|X - x| - \frac{1}{2}E_F|X - X'|, \quad (4)$$

where X and X' are independent copies of a random variable from probabilistic forecaster F and x is the ground truth. The $CRPS^*$ provides a direct way to compare deterministic and probabilistic models. In the case of the deterministic forecaster, the $CRPS^*$ reduces to Mean Absolute Error (MAE), which is a commonly used point-wise error metric.

In other words, in a deterministic setting, the $CRPS^*$ is equivalent to MAE:

$$MAE(x, \hat{x}) = E|\hat{x} - x|, \tag{5}$$

where x is the ground truth and $\hat{x}$ is the point forecast. After the GAN training concluded, we calculated the $CRPS^*$ of the ProbCast and the deterministic model. To calculate $CRPS^*$ for ProbCast using Equation (4), we sampled it 200 times (100 times for each random variable).

6. Results and Discussion

Table 2 presents the optimal hyperparameters we found for each dataset using our framework during the experiments, and Table 3 summarizes our experiments' results presenting the $CRPS^*$ of the best deterministic model and the ProbCast for each dataset.

Table 2. List of the hyperparameters alongside their optimal values for each experiment.

Generator Hyperparameters	**Electricity**	**Exchange-Rate**
Input windows size	174	170
Noise size	303	183
Number of GRU layers	1	1
Number of GRU cells in each layer	119	119
Discriminator Hyperparameters		
Number of GRU layers	3	1
Number of GRU cells in each layer	146	149

In the experiment with the electricity dataset, the ProbCast was more accurate than the deterministic model despite having an almost identical structure. Furthermore, this experiment showed that our model can provide precise forecasts for multivariate time-series even when the number of features is substantial. In the exchange-rate experiment, the ProbCast outperforms its deterministic predecessor, again despite structural similarities. We can also observe that our method works well even though the dataset is considerably smaller in comparison to that of the previous experiment.

Furthermore, it confirms that our framework is capable of transforming a deterministic model to a probabilistic model that is more accurate than its predecessor. The question now arises: Considering the sensitivity of GAN to the architecture of its components, why does employing the deterministic model architecture to define the ProbCast work well, when it is borrowed from a totally different setup? We think that the deterministic model provides us an architecture that is capable of learning a good representation from the input time window. Since the model is trained to learn the mean of possible outcomes, these representations contain a distinctive indicator of where the target distribution is located. With the help of these indicators, the MLP block learns to accurately transform the noise vector z to the probability distribution of future values.

Table 3. The results of the experiments for the deterministic model and the ProbCast reported in $CRPS^*$.

Dataset	**Deterministic Model**	**ProbCast**
Electricity	235.96	232.00
Exchange-rate	1.04×10^{-2}	8.66×10^{-3}

7. Conclusion and Future Works

In this paper, we present ProbCast, a probabilistic model for forecasting one step ahead of a multivariate time-series. We employ the potential of conditional GAN in a

learning conditional probability distribution to model the probability distribution of future values given past values, that is, $P(X_{t+1}|X_t, .., X_0)$.

Furthermore, we propose a framework to efficiently find the optimal architecture of GAN's components. This framework builds the probabilistic model upon a deterministic model to improve its performance. Hence, it enables us to search for the optimal architecture of a generator and a discriminator separately. Furthermore, it can transform an existing deterministic model into a probabilistic model with increased precision and better alignment with the real world.

We assess the performance of our method on two publicly available datasets. The exchange-rate dataset is a small dataset with few features, while the electricity dataset is bigger with a considerably larger number of features. We compare the performance of the ProbCast with its deterministic equivalent. In both experiments, our method outperforms its counterpart. The results of the experiments demonstrate that the ProbCast can learn patterns precisely from a small set of data and at the same time, it is capable of figuring out the dependencies between many features, and can forecast future values accurately in the presence of a big dataset. Furthermore, the results of the experiments indicate the successful application of our framework, which paves the way for a systematic and straightforward approach to exchanging currently used deterministic models with a probabilistic model to improve accuracy and obtain realistic forecasts.

The promising results of our experiments signify great potential for probabilistic forecasting using GANs and suggest many new frontiers for further pushing the research in this direction. For instance, we employ vanilla GAN for our research and there have been a lot of modifications suggested for improving GANs in recent years. One possible direction is to apply these modifications and inspect the improvement in the performance of the ProbCast. The other direction is experimenting with more sophisticated architectures for the generator and the discriminator. Finally, we only use the knowledge from the deterministic model to shape the generator. It would be interesting to push this direction and try to incorporate more knowledge from the deterministic model into the GAN training process to improve and optimize the probabilistic model.

Data Availability Statement: The datasets used in this study can be found at https://github.com/laiguokun/multivariate-time-series-data.

References

1. Mahalakshmi, G.; Sridevi, S.; Rajaram, S. A survey on forecasting of time series data. In Proceedings of the 2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16), Kovilpatti, India, 7–9 January 2016; pp. 1–8.
2. Gneiting, T.; Katzfuss, M. Probabilistic forecasting. *Annu. Rev. Stat. Its Appl.* **2014**, *1*, 125–151. [CrossRef]
3. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: http://www.deeplearningbook.org (accessed on 1 June 2019).
4. Esteban, C.; Hyland, S.L.; Rätsch, G. Real-valued (medical) time series generation with recurrent conditional GANs. *arXiv* **2017**, arXiv:1706.02633.
5. Golany, T.; Radinsky, K. PGANs: Personalized Generative Adversarial Networks for ECG Synthesis to Improve Patient-Specific Deep ECG Classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 557–564.
6. Haradal, S.; Hayashi, H.; Uchida, S. Biosignal data augmentation based on generative adversarial networks. In Proceedings of the 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 18–21 July 2018; pp. 368–371.
7. Nikolaidis, K.; Kristiansen, S.; Goebel, V.; Plagemann, T.; Liestøl, K.; Kankanhalli, M. Augmenting Physiological Time Series Data: A Case Study for Sleep Apnea Detection. *arXiv* **2019**, arXiv:1905.09068.
8. Ye, F.; Zhu, F.; Fu, Y.; Shen, B. ECG Generation With Sequence Generative Adversarial Nets Optimized by Policy Gradient. *IEEE Access* **2019**, *7*, 159369–159378. [CrossRef]
9. Wiese, M.; Bai, L.; Wood, B.; Buehler, H. Deep Hedging: Learning to Simulate Equity Option Markets. 2019. Available online: https://ssrn.com/abstract=3470756 (accessed on 1 December 2019).
10. Wiese, M.; Knobloch, R.; Korn, R.; Kretschmer, P. Quant GANs: deep generation of financial time series. *Quant. Financ.* **2020**, *20*, 1–22. [CrossRef]

11. Chen, Y.; Wang, Y.; Kirschen, D.; Zhang, B. Model-free renewable scenario generation using generative adversarial networks. *IEEE Trans. Power Syst.* **2018**, *33*, 3265–3275. [CrossRef]
12. Fekri, M.N.; Ghosh, A.M.; Grolinger, K. Generating Energy Data for Machine Learning with Recurrent Generative Adversarial Networks. *Energies* **2020**, *13*, 130. [CrossRef]
13. Zhang, C.; Kuppannagari, S.R.; Kannan, R.; Prasanna, V.K. Generative Adversarial Network for Synthetic Time Series Data Generation in Smart Grids. In Proceedings of the 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), Aalborg, Denmark, 29–31 October 2018.
14. Yoon, J.; Jarrett, D.; van der Schaar, M. Time-series Generative Adversarial Networks. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019; pp. 5509–5519.
15. Ramponi, G.; Protopapas, P.; Brambilla, M.; Janssen, R. T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. *arXiv* **2018**, arXiv:1811.08295.
16. Koochali, A.; Schichtel, P.; Dengel, A.; Ahmed, S. Probabilistic Forecasting of Sensory Data with Generative Adversarial Networks–ForGAN. *IEEE Access* **2019**, *7*, 63868–63880. [CrossRef]
17. Zhang, K.; Zhong, G.; Dong, J.; Wang, S.; Wang, Y. Stock market prediction based on generative adversarial network. *Procedia Comput. Sci.* **2019**, *147*, 400–406. [CrossRef]
18. Zhou, X.; Pan, Z.; Hu, G.; Tang, S.; Zhao, C. Stock market prediction on high-frequency data using generative adversarial nets. *Math. Probl. Eng.* **2018**. [CrossRef]
19. Lin, Y.; Dai, X.; Li, L.; Wang, F.Y. Pattern sensitive prediction of traffic flow based on generative adversarial framework. *IEEE Trans. Intell. Transp. Syst.* **2018**, *20*, 2395–2400. [CrossRef]
20. Kabir, H.M.D.; Khosravi, A.; Nahavandi, S.; Kavousi-Fard, A. Partial Adversarial Training for Neural Network-Based Uncertainty Quantification. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, 1–12. [CrossRef]
21. Hothorn, T.; Kneib, T.; Bühlmann, P. Conditional transformation models. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2014**, *76*, 3–27. [CrossRef]
22. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the 28th Annual Conference on Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
23. Mirza, M.; Osindero, S. Conditional generative adversarial nets. *arXiv* **2014**, arXiv:1411.1784.
24. Lai, G.; Chang, W.C.; Yang, Y.; Liu, H. Modeling long-and short-term temporal patterns with deep neural networks. In Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018; pp. 95–104.
25. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
26. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic Differentiation in PyTorch. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
27. Gneiting, T.; Raftery, A.E. Strictly proper scoring rules, prediction, and estimation. *J. Am. Stat. Assoc.* **2007**, *102*, 359–378. [CrossRef]