Exploring Generative Data Augmentation in Multivariate Time Series Forecasting : Opportunities and Challenges

Ankur Debnath
Mastercard AI Garage
Gurugram, Haryana, India
ankur.debnath@mastercard.com

Govind Waghmare
Mastercard AI Garage
Gurugram, Haryana, India
govind.waghmare@mastercard.com

Hardik Wadhwa
Mastercard AI Garage
Gurugram, Haryana, India
hardik.wadhwa@mastercard.com

Siddhartha Asthana
Mastercard AI Garage
Gurugram, Haryana, India
siddhartha.asthana@mastercard.com

Ankur Arora
Mastercard AI Garage
Gurugram, Haryana, India
ankur.arora@mastercard.com

ABSTRACT

In multivariate time series (MTS), each time point constitutes multiple time-dependent variables. Short-term and long-term correlation of these variables is a significant characteristic of MTS, and is a key challenge while modelling the same. While classical auto-regressive models are heavily used to model MTS, neural models are more flexible and efficient. However, neural models rely on a large amount of labelled data for training. Availability of labelled time series data could be a bottleneck in real-world scenarios. This scarcity of labelled data can be mitigated by data augmentation. In MTS, augmentation techniques need to realize short-term correlations and long-term temporal dynamics. In this work, we introduce a novel meta-algorithm for time-series data augmentation to address the data scarcity problem. Due to the intrinsic ordering of samples in time series, we argue that one cannot simply add synthetic samples to the real samples for augmentation. To this end, we generate synthetic MTS data preserving temporal dynamics using an off-the-shelf generative algorithm and frame augmentation in MTS as a transfer learning problem. In addition, we point out the drawbacks of generative model in MTS augmentation. We show the effectiveness of our method on publicly available MTS datasets in forecasting. We also perform qualitative and quantitative analysis of synthetic MTS data and its applicability in long-term forecasting. To the best of our knowledge, this is the first study on generative data augmentation for MTS forecasting.

CCS CONCEPTS

• **Applied computing** → **Forecasting**; • **Computing methodologies** → *Neural networks*; • **Mathematics of computing** → Time series analysis.

MileTS '21, August 14th, 2021, Singapore

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

ACM Reference Format:
Ankur Debnath, Govind Waghmare, Hardik Wadhwa, Siddhartha Asthana, and Ankur Arora. 2021. Exploring Generative Data Augmentation in Multivariate Time Series Forecasting : Opportunities and Challenges. In *MileTS '21: 7th KDD Workshop on Mining and Learning from Time Series, August 14th, 2021, Singapore.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

A multivariate time series(MTS) is a sequence of events measured at regular time intervals. Multivariate time series analysis is quintessential in forecasting, anomaly detection, and classification [4, 6, 12]. Everyday examples of MTS are the household electricity consumption, the solar-energy plant output, the currency exchange rates, the stock market prices, etc. In MTS, each time point constitutes multiple time-dependent variables. Real-world MTS exhibit complex temporal dynamics at each time point as well as in the short-term and long-term.

Auto-regressive models have been widely used to model MTS. But deep learning models are much more efficient and flexible, and can capture the complex characteristics of real world MTS data. Deep learning models have shown their capability in learning a function to accurately map inputs to outputs using a large amount of labelled data. Availability of labelled data is limited in many applications and hence, researchers proposed many data augmentation methods [18] to resolve data scarcity issues by increasing training data size and quality.

A few of the proposed data augmentation techniques attend to MTS temporal dynamics, which include generative methods such as [11, 19]. A good MTS generative framework should possess the following two properties: (a) it should model the distribution of each time point, and (b) as time evolves, it should capture temporal correlation of features [19]. Some augmentation methods like cropping, flipping, and warping may distort long-term temporal dynamics in task-dependent time series problems like classification, forecasting, and anomaly detection. Synthetic series based on such augmentations may not respect the original feature relationship across time, making it less similar and predictable concerning original MTS [18]. Time series data follows intrinsic time ordering. Thus, text and image-based augmentation methods may not work in the context of MTS. In order to alleviate the time ordering issue, transfer learning or fine-tuning is utilized for time-order preserving data augmentation [14].

In this work, we explore challenges associated with temporal dynamics-preserving augmentation in MTS forecasting using transfer learning. In line with the transfer learning procedure, we first train the forecasting model on real MTS (source domain) and use this pre-trained model as a starting point for synthetic MTS (target domain). Here, we use LSTNet [8] as a forecasting model , and TimeGAN [19] as a temporal dynamics-preserving generative framework (refer Figure 1). To summarize, we aim to improve the generalizability of the forecasting model using synthetic data. To this end, we present a novel and data-efficient meta-algorithm for augmentation to aid MTS forecasting. Also, we point out drawbacks of MTS augmentation in generative settings. Additionally, we perform qualitative and quantitative analysis of synthetic MTS data and its applicability in long-term forecasting. We believe, this is the first attempt to explore generative data augmentation in MTS forecasting.

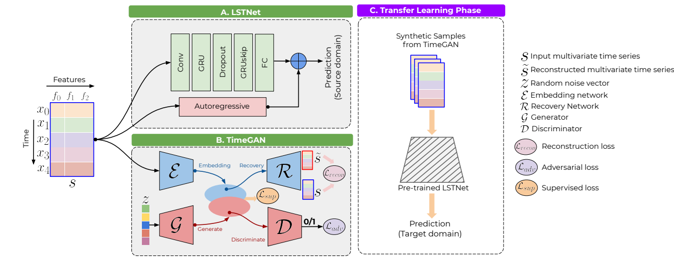


Figure 1: An overview of the proposed framework; based on TimeGAN [19] and LSTNet [8]. Input $\mathcal{S}$ is multivariate time series where x_0, x_1, x_3 and x_4 are time series instances each having features f_1, f_2 and f_3. (A)We train an LSTNet model using source data. (B)TimeGAN is trained using source data to generate synthetic MTS (target domain). (C) In the transfer learning phase, we use pre-trained LSTNet as an initial model and fine-tune on target data. (Zoom-in for best view)

2 RELATED WORK

As real-world multivariate time series evolve across time, they manifest complex mechanisms over the short-term and long-term. The intrinsic temporal dependency allows time-series transformation into frequency and time-frequency domains. Deep learning-based models are flexible and efficient in multiple time series tasks [18]. Due to multivariate and time evolution nature, augmentations from image, text, and speech modalities may yield poor results on different tasks in time series. Therefore, it is crucial to develop deep learning-based augmentation methods to strengthen training data size and quality. [7, 18] systematically review multiple task-specific time series augmentation methods. Specifically, [18] proposed a taxonomy of augmentation techniques ranging from simple to advanced approaches. Simple approaches involve time, frequency, and time-frequency domain augmentation techniques like cropping, warping, flipping, etc. On the other side, advanced methods include decompositional methods, generative, and learning-based techniques.

In MTS modelling, three things are crucial: (a) correlation between multiple variables at a given time step, (b) short-term correlation, and (c) long-term correlation. [5] explored encoder-decoder-based frameworks along with recurrent and convolutional modules. [9] demonstrates autoregressive methods for forecasting. LSTNet [8] proposed a hybrid model consisting of an autoregressive path, convolutional, and skip-connected recurrent layers. It focused on the periodic and non-periodic nature of real-world MTS datasets. TimeGAN [19] is a generative framework, which captures temporal dynamics of data across time. It involves joint training of supervised and unsupervised objectives via a learned embedding space. TimeGAN compares its approach with RCGAN [3] and C-RNN-GAN [13]. Other aspects of time series like data dependence, temporal modelling, and attention mechanisms are explored by [10, 15–17]. Although multiple augmentation methods are proposed in the literature, very few of them [1, 19] generate temporal dynamic-preserving synthetic data.

3 DATASETS

We used four public benchmark datasets[1]. Table 1 summarizes the dataset statistics. Stocks [2]: It contains daily historical Google stocks data from 2004 to 2019. Solar-Energy[1]: The solar power production records in the year of 2006 in Alabama State. Electricity[1]: The electricity consumption in kWh recorded every 15 minutes from 2012 to 2014, for N = 321 clients. Exchange-Rate[1]: The collection of the daily exchange rates of eight countries ranging from 1990 to 2016.

Datasets	N	L
Electricity	321	26,304
Exchange Rate	8	7,588
Stocks	6	3,685
Solar-Energy	137	52,560

Table 1: Dataset Statistics, where N is the number of variables, L is the length of the time series

We split all datasets into a training set (60%), validation set (20%) and test set (20%) in chronological order. Financial datasets such as Exchange Rate and Stocks are inherently aperiodic and have high temporal correlations across dimensions, as shown in Figure 4. On the other hand, Electricity and Solar are more periodic and contain seasonalities.

4 PROPOSED METHODOLOGY

Figure 1 shows the complete end-to-end pipeline of the proposed framework. The entire pipeline can be divided into three parts:

- Training LSTNet on original data
- Training TimeGAN to generate synthetic data
- Transfer learning on pretrained LSTNet model using synthetic data

[1]https://github.com/laiguokun/multivariate-time-series-data
[2]https://finance.yahoo.com/quote/GOOG/history?p=GOOG

In the first part, we train LSTNet on original data and create a baseline model for fine-tuning. In line with [8], architecture of LSTNet is shown in Figure 1A. The LSTNet has two main hyperparameters, commonly seen in any deep forecasting model, window size w and horizon length h. These two are crucial and need to be set properly to work in coherence with data augmentation used here *i.e.* TimeGAN. The other network parameters are chosen as suggested by the authors. The second part involves training TimeGAN to generate synthetic samples for the data augmentation process. We refer [19] for TimeGAN architecture and as shown in Figure 1B. TimeGAN generates synthetic data using subsequences. It uses the same subsequence for sampling. The most critical parameter of TimeGAN is the sequence length s. The choice of this sequence length, in this method, depends on the w and h of LSTNet. Diversity, fidelity and predictive utility are important factors to assess the quality of synthetic data. In line with [19], we perform this assessment based on two scores; predictive score and discriminative score, in addition to statistical measures described in Section 6.2. The training of TimeGAN is done on the same original dataset used for the creation of the base model.

The final step involves the feeding of the generated samples from TimeGAN to be utilized by LSTNet. Figure 1C provides details of the transfer learning procedure. Time series data follows intrinsic time ordering. Thus, text and image-based augmentation methods may not work in the context of time series. We employ transfer learning procedures to alleviate the time ordering issue. A forecasting model trained on original time series can improve its generalizability once fine-tuned on synthetic data [14].

5 EXPERIMENTAL SETUP

In this section, we describe the experimental setup for our proposed methodology in detail. At the end, we present a comprehensive evaluation and analysis of the proposed method. The supplementary code for the base experiments of LSTNet[3][8] and TimeGAN[4][19] provided by the authors are used in this work. Training is performed using a 60-20-20 train-validation-test split, where the 20% test or held out set is unseen by both LSTNet and TimeGAN, to ensure that there is no data leakage.

5.1 Implementation Details

5.1.1 Problem Formulation. The task of multivariate time series forecasting is carried out in this work. A given time series, represented by $X = \{x_1, x_2, ..., x_T\}$, where $x_t \in \mathbb{R}^n$, T is the length of the time series, n is the number of time series or dimensions. Forecasting is done in a rolling window style, where task is to predict $y = x_{T+h}$, where h is the desired horizon for the forecast ahead of the given timestamp, given that $\{x_1, x_2, ..., x_T\}$ is available, i.e. the input-output pair is given $(X_T, y) \equiv \{(x_1, x_2, ..., x_T), x_{T+h}\}$, where $X_T \in \mathbb{R}^{n \times T}$, is the input matrix and output $y \in \mathbb{R}^n$.

5.1.2 Preprocessing. Data preprocessing is a crucial step involved in modelling time series data. The general practices involve different normalizations and scaling of the data which are both data and model-dependent. Since datasets have different variances for each column, we chose to normalize the data for each time series in a dataset. The normalization procedures adopted in LSTNet [8] and TimeGAN [19] are used and given by Eqn. (1) and (2) respectively:

$$X_{i,j} = \frac{X_{i,j}}{max(X_j)} \tag{1}$$

$$X_{i,j} = \frac{X_{i,j} - min(X_j)}{max(X_j) - min(X_j)} \tag{2}$$

where i is the time step and j is a feature index.

5.1.3 Base Model Creation. The LSTNet is a deep learning architecture designed for the task of multivariate time series forecasting. The first task in this framework is to create base models on each dataset, which will act as the baseline for improvement. We train these base models using a similar procedure described in LSTNet. It includes normalization given by Eqn. 1. Training hyperparameters are shown in Table 5, and forecasting parameters are stated in Table 6. Early stopping is employed to obtain the best model, which will serve as a base model for fine-tuning. Base model results are discussed in results (Section 6).

Dataset	Discriminative Score	Predictive Score
Exchange Rate	0.2377	0.081
Stocks	0.1834	0.0406
Electricity	0.4999	0.0347
Solar-Energy	0.4626	0.0412

Table 2: Evaluation metrics and results for generated data from TimeGAN. Lower Discriminative and Predictive scores indicate better quality of synthetic data.

5.1.4 Preserving Temporal-dynamics. Temporal dynamics in the context of MTS is the property of such data having complex interactions across variables and time. It captures short-term and long-term patterns of the data. Preserving temporal dynamics is critical while generating synthetic MTS data for augmentation [11, 19]. Generative models such as TimeGAN [19] ensure that such characteristics of MTS are preserved. In this work, we generate synthetic MTS data through TimeGAN and evaluate the temporal dynamics of synthetic data through various metrics described in Section 6.2.

5.1.5 Generating Synthetic Samples. In this work, data augmentation is incorporated through a generative model TimeGAN [19]. TimeGAN takes in the original sequence and breaks it down to windows of sequences, similar to LSTNet and randomly permute them, assuming each window of length s is iid. After training, the outputs are synthetic sequences corresponding to the input window sequences. The number of sequences in both real and synthetic data is the same. The quantitative evaluation of the synthetic samples is done using two metrics, Discriminative score and Predictive score, as defined by [19]. Synthetic sequences for data augmentation is obtained by training TimGAN on each dataset with sequence lengths as $w + h$ according to Table 6. The rest of the hyperparameters during training are stated in Table 7. These hyperparameters are not changed for any dataset and have been used throughout. The metrics obtained for each dataset upon training are shown in Table 2. We will revisit the impact of these metrics in our final discussion.

[3]https://github.com/fbadine/LSTNet
[4]https://github.com/jsyoon0823/TimeGAN

5.1.6 Knowledge Transfer. In this work, we propose to use transfer learning using generative (model-based) data augmentation using TimeGAN experimented on LSTNet, but can easily be extended to any deep forecasting models. The main challenge in this task is to connect the output of TimeGAN to LSTNet in a transfer learning framework. We propose the following 2-step approach; **Matching Hyperparameters :** The synthetic sequences generated from TimeGAN are sequences corresponding to slided window sequences of the original time series. Whereas, the inputs to the LSTNet are slided window sequences extracted from the original time series. Therefore, to feed TimeGAN generated output to LSTNet, the sequence length s of TimeGAN must be equal to $w + h$ of LSTNet so that, LSTNet can have the input-output pair covered by the synthetic sequence. **Fine-Tuning :** The base LSTNet model trained on the original time series contains learnt parameters. When LSTNet is to be trained using synthetic sequences generated from TimeGAN, the model is initialized by weights of the base model. Additionally, freezing the initial convolutional and recurrent layers ensure that the higher-order information of the time series can be transferred during augmentation. In this work, we present an extensive ablation study on how fine-tuning each layer impacts the end performance. While fine-tuning, there is a huge possibility of overfitting and non-convergence. Since the dataset size for the base model as well as the augmented one is the same, hence overfitting on a small dataset can be safely ruled out. However, the loss can easily overshoot if proper hyperparameter tuning is not done. Therefore, the obvious steps include, using an annealing and low learning rate, increasing regularization by increasing dropout probabilities. The common modified hyperparameters in this part of the experiment are listed in Table 8.

6 RESULTS & DISCUSSION

In this section, we present a summary of results obtained by data augmentation and justify them. We discuss the importance of the quality of synthetic data and point out the relevant gap in the literature and possible research direction. We also introduce the concept of selective data augmentation in the context of MTS data augmentation.

6.1 Analysis of Data Augmentation

This section discusses the quantitative results obtained and the evaluation of the proposed methodology. It includes evaluation metrics like Root Relative Squared Error(RSE), Relative Absolute Error(RAE), and Empirical Correlation Coefficient(CORR). Table 3 summarizes the results on each dataset under different settings. It contains results on the base model and the impact of different layers unfreezing. Data augmentation using our method provides a significant performance boost of 25.36% on the RSE for Stocks dataset and 0.05% on the Exchange Rate dataset, evident by the decrease in RSE and RAE scores. Thus, it indicates that transfer-learning on synthetic data is helping increase model generalizability. As the quality of synthetic data generated by TimeGAN for Stocks and Exchange Rate is good, forecasting accuracy is also improved. Contrary, performance tends to degrade in the case of Electricity and Solar-Energy datasets. These datasets have prominent periodic patterns and a large cardinality (refer Figure 4 and Table 4). TimeGAN is sensitive to periodicity and cardinality. Therefore, as shown in Table 2, the quality of synthetic data generated for Electricity and Solar-Energy is not good. We discuss in detail the impact of the quality of synthetic data in the next section. Another observation from the results is the contribution of unfreezing more and more layers has an overall positive impact, as far as the transfer-learning approach is concerned. It shows that unfreezing all the layers, except the convolutional layer, helps the model adapt to the augmented data better. It leads to better generalizability depending on the quality of synthetic data.

6.2 Synthetic Data Evaluation

In this section, we attempt to analyze the performance from a synthetic data quality standpoint. We evaluate the synthetic data generated by TimeGAN through different metrics and point out relevant gaps. Also, we propose a method of selective augmentation for mitigating quality issues impacting model generalizability. For evaluating the quality of synthetic data, we perform an exploratory data analysis and comparison of real and generated data through various statistical measures. TimeGAN[19] performs better where datasets have high temporal correlations and less periodicity. Whereas in datasets such as Electricity and Solar, TimeGAN seems to struggle in terms of capturing very long and very short periodic patterns shown by Figure 5. We perform extensive statistical analysis of the synthetic data through metrics like KL Divergence, Pairwise Correlation Difference, Discriminative and Predictive Scores[19]. Figure 6 and 7 show the PCA and t-SNE plots of the real and synthetic data generated by TimeGAN, which qualitatively indicates that the quality of Stocks and Exchange Rate synthetic data is good, whereas poor quality is obtained in Solar and Electricity. Table 4 summarizes different metrics computed across all datasets. Also for further analysis, the probability density functions of generated and real data are compared in Figure 2. This analysis shows that feature correlations are difficult to model with increasing dimensionality. Also, TimeGAN may be suffering from possible mode collapse (Figure 2). Fig 5 points out that both short and long term patterns are modeled incorrectly. And as the length of time series increases, capturing long term correlations gets even more challenging for TimeGAN. This can be a potential research direction in developing better generative models that are agnostic to mode-collapse, model long and short term patterns and correlations, as well as scalable in terms of cardinality. Some work has been done to address these issues [11], but the fidelity of such models is yet to be explored in cases of data augmentation.

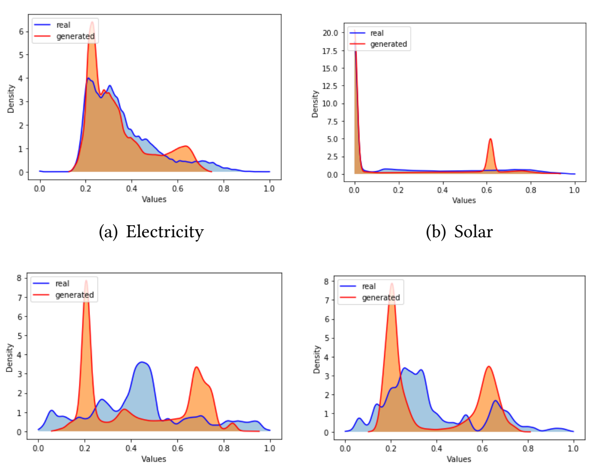
6.2.1 Selective Data Augmentation. In this section, we propose a mitigating strategy to tackle poor quality generated data motivated by [2]. We show the effects of selective augmentation and compare it against complete augmentation and its effects on the performance of the model. We deploy a random sampling strategy, where a fraction of total sequences is used to augment and end performance is evaluated. Figure 3 shows that if the quality of generated data is poor, increasing augmentation size degrades the performance. This implies that imputing poor quality data to the model corrupts its knowledge base. Hence sampling can become an effective way in such cases. But, in our case, sampling was random. An informed or strategic sampling may even provide a boost to the model.

Dataset	Metrics	Strategies				
		Base Model	With FC Layer unfreezed	FC+Highway	FC+Highway+Skip-GRU	Only CNN layer retained
Exchange Rate	RSE	0.0353	0.0353	0.0351	0.0351	**0.0351**
	RAE	0.0296	0.0292	0.0290	0.0291	**0.0290**
	CORR	**0.9539**	0.9538	0.9535	0.9536.	0.9537
Stocks	RSE	0.7055	0.6601	0.4792	0.457	**0.4519**
	RAE	0.3177	0.3536	0.2331	0.2112	**0.2008**
	CORR	0.7838	**0.8026**	0.7917	0.7918	0.7897
Electricity	RSE	**0.1032**	0.2157	0.3900	0.1626	0.1107
	RAE	**0.0563**	0.1301	0.1970	0.0902	0.0611
	CORR	**0.8985**	0.7400	0.6041	0.8684	0.8956
Solar-Energy	RSE	**0.3966**	0.5951	0.4257	0.5158	0.5566
	RAE	**0.2287**	0.4929	0.2783	0.3946	0.4251
	CORR	**0.9191**	0.8829	0.9158	0.917	0.9095

Table 3: Results summary (in RSE, RAE and CORR) of all strategies on four datasets: 1) each row has the results of a specific dataset in a particular metric across different strategies; 2) each column compares the results of all strategies on a particular dataset; 3) bold face indicates the best result of each row in a particular metric. Lower RSE and RAE scores correspond to better performance, whereas higher CORR score corresponds to better performance.

Details	Electricity	Stocks	Exchange Rate	Solar
Size	26,304	3,685	7,588	52,560
Cardinality	321	6	8	137
Rel. perf. (↑)	-1.00%	25.36%	0.05%	-10.03%
Disc. score (↓)	0.4999	0.1834	0.2377	0.4626
Pred. score (↓)	0.0347	0.0406	0.0981	0.0412
KLD (↓)	64992.33	6847.39	8256.62	32319.53
PCD (↓)	48.44	1.073	1.397	10.587

Table 4: Different metrics computed across all the datasets which include Relative performance (Rel. perf), Discriminative Score (Disc. score), Predictive Score(Pred. score), KL Divergence(KLD), Pairwise correlation differnce(PCD). (↑) means higher, the better. (↓) means lower, the better.



(a) Electricity (b) Solar

(c) Exchange Rate (d) Stock

Figure 2: Probability density functions of synthetic and real data (zoom-in for best view)

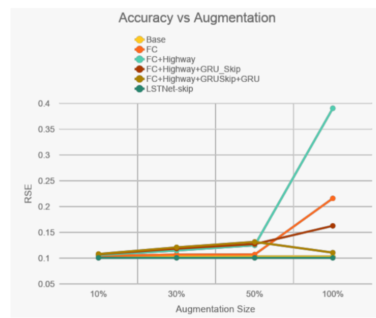


Figure 3: Comparison of varying augmentation size and performance at each layer for Electricity dataset. Increasing augmentation size in case of poor quality synthetic data leads to degradation in performance (zoom-in for visibility).

7 CONCLUSION

This paper presents a novel and data-efficient meta-algorithm for augmentation to aid MTS forecasting. In this, we explore challenges associated with temporal dynamics-preserving augmentation. Also, we perform quantitative and qualitative analysis on synthetic MTS data and its usability in long-term forecasting. To the best of our knowledge, this is the first attempt to explore and examine augmentation using GAN generated time-series. The proposed approach shows improvements on datasets that are aperiodic and have high temporal correlations. Also, it highlights the shortcomings of generative modelling of MTS data. Empirically, it is shown that augmentation heavily relies on quality of synthetic data. Consistent with literature, the utility of synthetic data in long-term forecasting depends on the ability of such generative models to capture short and long-term correlations in the time series and across variables. This can be a potential research direction in developing better generative models that are agnostic to mode-collapse, model long and short term patterns and correlations, as well as scalable in terms of cardinality. Though current work is focused on forecasting task, it would be interesting to see an extension of the above work to other time-series tasks such as classification and anomaly detection.

REFERENCES

[1] Kasun Bandara, Hansika Hewamalage, Yuan-Hao Liu, Yanfei Kang, and Christoph Bergmeir. 2020. Improving the Accuracy of Global Forecasting Models using Time Series Data Augmentation. *arXiv preprint arXiv:2008.02663* (2020).

[2] Binod Bhattarai, Seungryul Baek, Rumeysa Bodur, and Tae-Kyun Kim. 2020. Sampling strategies for GAN synthetic data. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2303–2307.

[3] 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).

[4] 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).

[5] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[6] Zhongyang Han, Jun Zhao, Henry Leung, King Fai Ma, and Wei Wang. 2019. A review of deep learning models for time series prediction. *IEEE Sensors Journal* (2019).

[7] Brian Kenji Iwana and Seiichi Uchida. 2020. An empirical survey of data augmentation for time series classification with neural networks. *arXiv preprint arXiv:2007.15951* (2020).

[8] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. 2018. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 95–104.

[9] Bryan Lim and Stefan Zohren. 2020. Time Series Forecasting With Deep Learning: A Survey. *arXiv preprint arXiv:2004.13408* (2020).

[10] Bryan Lim, Stefan Zohren, and Stephen Roberts. 2019. Recurrent Neural Filters: Learning Independent Bayesian Filtering Steps for Time Series Prediction. *arXiv preprint arXiv:1901.08096* (2019).

[11] Zinan Lin, Alankar Jain, Chen Wang, Giulia Fanti, and Vyas Sekar. 2020. Using GANs for Sharing Networked Time Series Data: Challenges, Initial Promise, and Open Questions. In *Proceedings of the ACM Internet Measurement Conference*. 464–483.

[12] Xinrui Lyu, Matthias Hueser, Stephanie L Hyland, George Zerveas, and Gunnar Raetsch. 2018. Improving clinical predictions through unsupervised time series representation learning. *arXiv preprint arXiv:1812.00490* (2018).

[13] Olof Mogren. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904* (2016).

[14] Sinno Jialin Pan and Qiang Yang. 2009. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2009), 1345–1359.

[15] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. 2018. Deep state space models for time series forecasting. In *Advances in neural information processing systems*. 7785–7794.

[16] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting* 36, 3 (2020), 1181–1191.

[17] Yuyang Wang, Alex Smola, Danielle C Maddix, Jan Gasthaus, Dean Foster, and Tim Januschowski. 2019. Deep factors for forecasting. *arXiv preprint arXiv:1905.12417* (2019).

[18] Qingsong Wen, Liang Sun, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. 2020. Time Series Data Augmentation for Deep Learning: A Survey. *arXiv preprint arXiv:2002.12478* (2020).

[19] Jinsung Yoon, Daniel Jarrett, and Mihaela van der Schaar. 2019. Time-series generative adversarial networks. In *Advances in Neural Information Processing Systems*. 5508–5518.

A HYPERPARAMETERS

The different sets of hyperparameters used in the experimental setup are shown below. These are very specific to the datasets used in this paper and need to be tuned according to the dataset under consideration.

B DATA ANALYSIS

In this, we compare the real and generated data on various parameters. Autocorrelation plots of the real datasets are shown in Figure 4. Figure 5 compares the real and generated data with the help of autocorrelation plots with lags 1000 and 5000, for both Electricity and Solar datasets. This figure shows how much of the short and long-term patterns in the data have been modeled by TimeGAN. Figure 6 and 7 show the PCA and t-SNE plots of real and generated data and qualitatively indicate how TimeGAN performs across all datasets.

Hyperparameter	Value
Epochs	100
Batch Size	128
Optimizer	Adam
Learning Rate	0.001

Table 5: Hyperparameters for Base Model

Dataset	w	h
Electricity	168	24
Exchange Rate	168	12
Stocks	168	12
Solar-Energy	168	12

Table 6: Forecasting parameters for each dataset

Hyperparameter	Value
Module	GRU
Batch Size	128
Iterations	10000
Hidden Dimensions	24
Number of Layers	3

Table 7: Hyperparameters for training TimeGAN

Hyperparameter	Value
Epochs	20
Batch Size	128
Optimizer	Adam
LR Scheduler	StepLR
Learning Rate	10^{-5}

Table 8: Hyperparameters for Fine-Tuning LSTNet with TimeGAN generated data

C EVALUATION METRICS

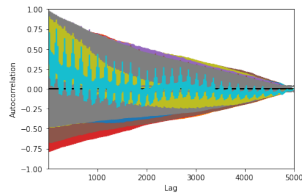
The various evaluation metrics used in this work include Root Relative Squared Error(RSE), Relative Absolute Error(RAE), Empirical Correlation Coefficient(CORR), KL Divergence and Pairwise Correlation Difference(PCD). These are defined as follows:

$$RSE = \frac{\sqrt{\sum_{(i,t)\in\Omega_{test}} (Y_{it} - \hat{Y_{it}})^2}}{\sqrt{\sum_{(i,t)\in\Omega_{test}} (Y_{it} - mean(Y))^2}} \tag{3}$$

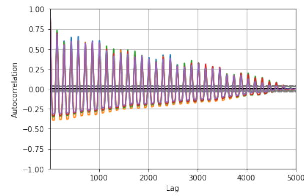
$$RAE = \frac{\sum_{(i,t)\in\Omega_{test}} (Y_{it} - \hat{Y_{it}})}{\sum_{(i,t)\in\Omega_{test}} (Y_{it} - mean(Y))} \tag{4}$$

$$CORR = \frac{1}{n}\sum_{i=1}^{n} \frac{\sum_t (Y_{it} - mean(Y_i))(\hat{Y}_{it} - mean(\hat{Y}_i))}{\sqrt{\sum_t (Y_{it} - mean(Y_i))^2 (\hat{Y}_{it} - mean(\hat{Y}_i))^2}} \tag{5}$$

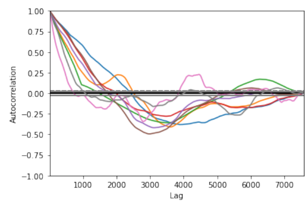
$$D_{KL}(P||Q) = \sum_{i=1}^{|v|} P_v(i) \frac{P_v(i)}{Q_v(i)} \tag{6}$$



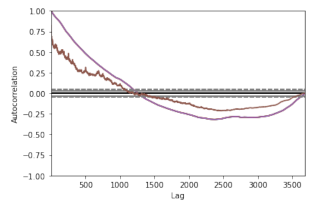
(a) Electricity



(b) Solar

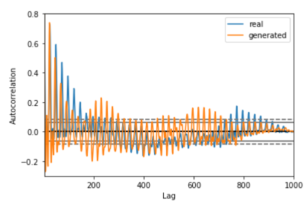


(c) Exchange Rate

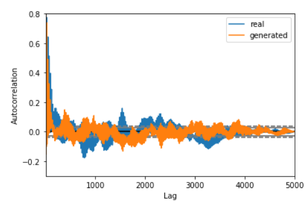


(d) Stock

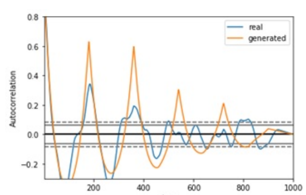
Figure 4: Auto-correlation graphs of sampled variables (zoom-in for visibility)



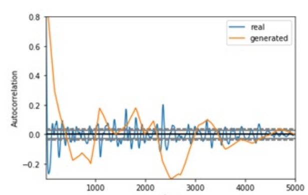
(a) Electricity short term



(b) Electricity long term



(c) Solar short term



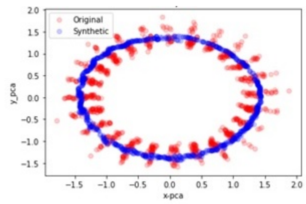
(d) Solar long term

Figure 5: Auto-correlation graphs of real and synthetic data for Electricity and Solar datasets. Figure 5(a) and 5(c) are autocorrelations for lag = 1000, which show short term periodicity. Figure 5(b) and 5(d) are autocorrelations for lag = 5000, which show long term periodicity. (zoom-in for visibility)

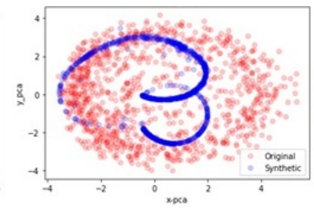
$$PCD(X_R, X_S) = \|Corr(X_R) - Corr(X_S)\|_F \tag{7}$$

where, $Y, \hat{Y} \in \mathbb{R}^{n \times T}$ are the true and predicted samples respectively. X_R, X_S are real and synthetic data matrices respectively. P_v is the pdf of variable v of real data and Q_v is the pdf of variable v of synthetic data.

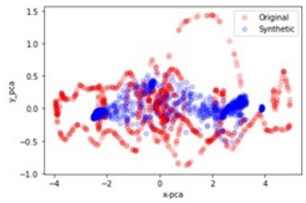
RAE and RSE are the scaled version of Relative Absolute Error and Root Mean Squared Error, used for better interpretation. For RSE and RAE, a lower score is better, whereas a higher CORR score is better. KL Divergence (KLD) measures similarity between two



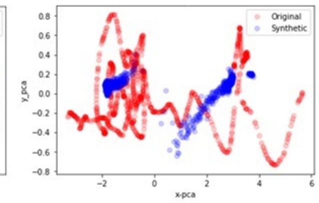
(a) Electricity



(b) Solar

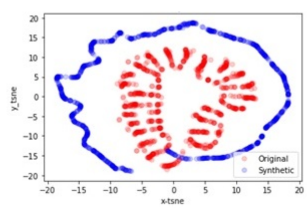


(c) Exchange

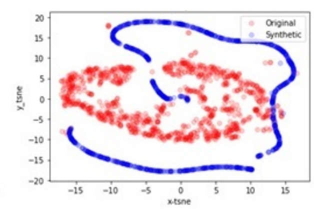


(d) Stock

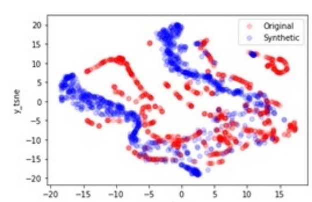
Figure 6: PCA plots of real and synthetic data. (zoom-in for visibility)



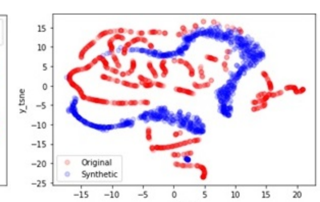
(a) Electricity



(b) Solar



(c) Exchange



(d) Stock

Figure 7: t-SNE plots of real and synthetic data. (zoom-in for visibility)

pdf/pmf, but only computed for each variable independently. However, it doesn't capture interactions(dependencies) among variables. KLD is a variable level metric and lower the distance, closer the synthetic data to the real data. Pairwise Correlation Difference (PCD) measures correlation among variables. Lower the PCD, the closer the synthetic data is to real data in terms of linear correlations. PCD metric is defined at the dataset level.