Time series Synthesis using Generative Adversarial Networks – A take on DoppelGANger

Auke Schaap, under supervision of Lydia Chen, Zilong Zhao, and Aditya Kunar

Abstract

With a growing need for data comes a growing need for synthetic data. In this work we reproduce the results of DoppelGANger [2] in synthesising time series data with metadata. We identify a key issue in the comparison made in [2] of DoppelGANger to TimeGAN, RNNs, AR and HMM models, which creates a new avenue of time series synthesis using GANs. We show that not all results of [2] can be reproduced. We furthermore find that DoppelGANger does not adequately capture measurement-metadata correlations of our dataset. Sample size reduction is shown to be an effective tool to reduce training time while still attaining accurate results, and the key parameter S is tuned further. Finally we show that execution on CPU has similar training times as execution on GPU by [2], suggesting that the original code can be improved, and we release our version of the models ourselves, to enable easy reproduction. In closing points we shine light on possible future improvements that we were unable to test ourselves, and conclude that DoppelGANger is a promising model that opens the door to new unseen applications of GANs for time series synthesis.

I. Introduction

Generative Adversarial Networks [1] (GANs) are a type of neural network that can be used to generate synthetic data. This research reproduces the results of [2] that uses a GAN named DoppelGANger to generate synthetic time series data with metadata.

II. Aims

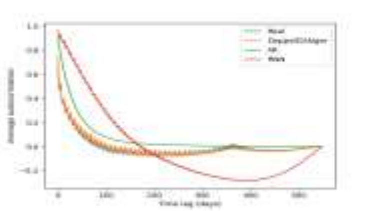
In this reproduction we want to evaluate DoppelGANger on:

- Temporal correlations
- Cross-measurement correlations
- Metadata distribution

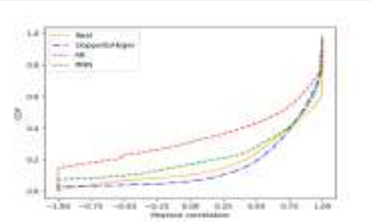
We compare DoppelGANger to an RNN and an AR model. We do analyse differential privacy.
We also evaluate DoppelGANger on the measurement-meta correlation, which [2] does not.

III. Results

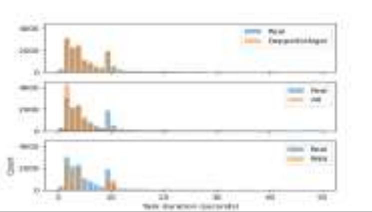
Temporal correlations
We find that the average autocorrelation of DoppelGANger on the WWT dataset is similar to the results of [2], and DoppelGANger successfully captures the annual and weekly correlations. The image on the right plots this.



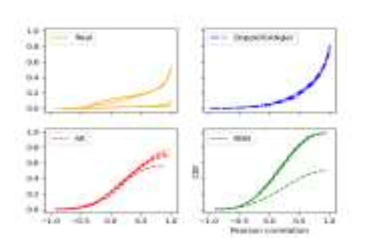
Cross-measurement correlation
We find that we capture the cross-measurement correlation worse than [2], but still better than the other models. Notably, increasing the sample size does not necessarily improve the results. The CDF of the Pearson correlation can be seen on the right.



Metadata distribution
DoppelGANger captures the metadata distribution as expected, better than the other models. A histogram of the task duration of the GCUT dataset can be seen on the right.

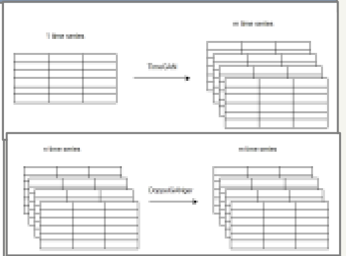


Measurement-metadata correlation
DoppelGANger does not capture the measurement-metadata correlation on the GCUT dataset adequately. On the right we see that the top left subplot shows 2 distinct modes; DoppelGANger on the top left subplot captures only one of these modes.

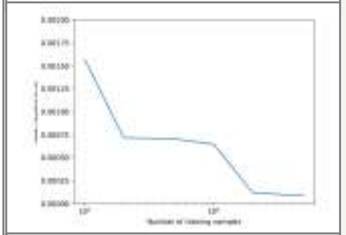


IV. Improvements

Comparing DoppelGANger
Comparing DoppelGANger with RNNs and AR models is imprecise, as these models use a single spatial dimension while DoppelGANger uses multiple time series and hence is multi-spatial dimensional, as shown here.

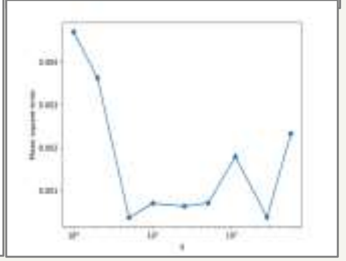


Sample size
Decreasing the sample size is an effective tool to reduce training time, for the WWT dataset. The MSE is plotted against the number of samples.



GPU vs CPU
Training times on CPU and GPU are shown to be similar, so improvements can be made to the implementation on GPU of the model. We have provided the CPU code.

Parameter tuning
We tune the special batch parameter S and find that S = 5 gives good results; we also find that S = 275 gives good results, which is not what we expected.



V. Conclusion

While some of the results attained by [2] have been reproduced, others have yielded dierent results. Notably, DoppelGANger is compared to models that are not expected to perform well at the tasks at hand, and hence we believe that they do not offer sufficient support to establish the promise of [2]. This also means that in essence, [2] offers a new insight to time series synthesis using GANs that has not been seen yet, which opens the door to new unseen applications of GANs for time series synthesis.

References

[1] Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. 2017. arXiv: 1701 . 00160

[2] Zinan Lin et al. Using GANs for Sharing Networked Time Series Data". In: Proceedings of the ACM Internet Measurement Conference (Oct. 2020). doi: 10.1145/3419394.3423643.