Exploring the use of TimeGAN to Synthesize Ambient Vibrations for Earthquake Monitoring Systems

Ivana Koon Yee Lim and Fritz Kevin Flores
Advanced Research Institute for Informatics, Computing and Networking
ivana_koon_lim@dlsu.edu.ph

Abstract: Data accessibility has always been a present issue in the development of intelligent models, especially for domain-specific problems wherein the availability of data is limited. Synthetic data generation has been an emerging trend for use cases that require a large dataset for machine learning but available data is sparse, or have sensitive data and privacy issues, or access to the data is limited or unavailable. Traditional approaches for data augmentation and synthesis fail to fully replicate the underlying statistical properties of a given dataset; however, the emergence of a class of neural networks in deep learning called Generative Adversarial Networks (GAN) have been promising in terms of its capability to preserve the underlying statistical distributions of a dataset. This paper explores the possibility of using a novel GAN architecture for synthesizing time series data called TimeGAN on accelerometer-based monitoring systems. The model was evaluated using both qualitative and quantitative methods and results show that the generated synthetic ambient vibration data have similar distribution as that of the real data, showing some promising results using the TimeGAN model.

Key Words: Data Augmentation, Synthetic Data Generation, Time Series, and Generative Adversarial Networks, Earthquake Monitoring

1. INTRODUCTION

Data accessibility has always been a present issue in the development of intelligent models, especially for domain-specific problems wherein the availability of data is limited. Generative Adversarial Network (GAN) is a class of neural networks in deep learning that is able to produce and synthesize new data (Goodfellow et al., 2014). This deep learning algorithm has gained popularity over the recent years in the field of computer vision mainly due to its ability to synthesize realistic images (Alqahtani, Kavakli-Thorne, & Kumar, 2019). However, much is still yet to be explored when it comes to data outside of the image or video domain such as time series domain, specifically time series data from vibration or earthquake monitoring systems e.g. Palert system.

Existing works on time series synthesis have been explored and have focused on domains such as audio and music (Dong, Hsiao, Yang, & Yang, 2018; Donahue, McAuley, & Puckette, 2019; Engel et al., 2019), and medical field applications (Esteban, Hyland, & Rätsch, 2017; Hazra & Byun, 2020; Vaccari, Orani, Paglialonga, Cambiaso, & Mongelli, 2021). Additionally, developing a model that is capable of generating realistic data from earthquake monitoring systems implies modeling the process that generates such time series information. This can help represent the direction towards building innovative approaches for modeling predictive systems such as earthquake detection and early warning systems and also structural health monitoring systems. Moreover, this paper focuses on synthesizing ambient vibrations using a novel time series GAN architecture called TimeGAN (Yoon et al., 2019) for purposes such as differentiating ambient vibrations from earthquake vibrations. This leads to the research question: can GANs be trained to synthesize time series data from earthquake monitoring systems?

To address the research question, several factors need to be considered such as how the data is collected and how the data is prepared prior to modeling. Data preparation is important in terms of feeding into a generative model for synthesizing more quality data. In order to model the time element of the data, necessary preprocessing needs to be performed. This includes setting the window size and the range of time overlaps. Intrinsically, this work will be exploratory in terms of implementing different techniques in order to generate time series data from vibration or earthquake monitoring systems.

2. METHODOLOGY

2.1 Dataset Details

The earthquake monitoring system used in this study is the Palert system from Sanlien which is an industry-grade accelerograph used for detecting earthquake waves. It is embedded with micro-electromechanical system (MEMS) accelerometers (Figure 1a). Earthquake information such as trigger time, intensity, and acceleration are recorded and can be retrieved from the system. Figure 1b shows the interface of the Palert system.

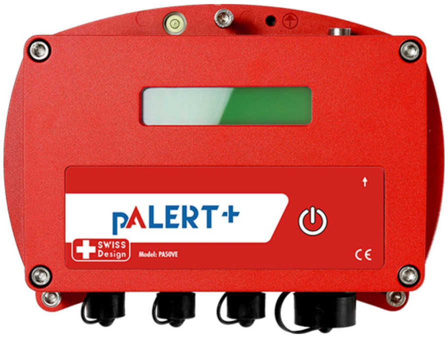
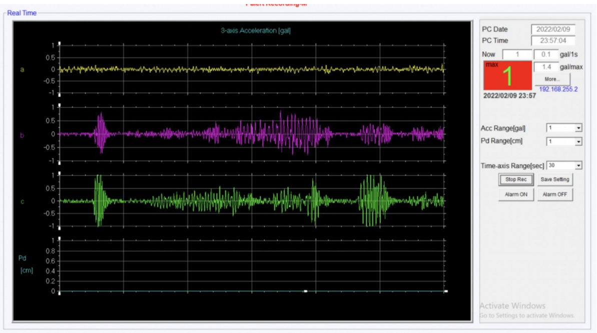


Fig. 1a. Palert accelerograph by Sanlien.



PC Date 2022/02/09
PC Time 23:57:04
Now 1 0.1 gal/1s
max 1 1.4 gal/max
192.168.255.2
2022/02/09 23:57
Acc Range[gal] 1
Pd Range[cm] 1
Time-axis Range[sec] 30

Fig. 1b. Palert system interface.

The Palert system uses a sampling rate of 100Hz, which records 100 readings per second. Table 1 shows sample values exported from the Palert system. In order to explore the capability of the model to synthesize ambient vibrations, which is the focus of this paper, a 30-minute duration of recorded ambient vibration data from the Palert system was first parsed into a compatible format for data analysis. The main information studied are the *a, b,* and *c* axes of the MEMS accelerometer. Figure 2a shows a plot of the ambient vibrations from the *a, b,* and *c* axes while Figure 2b shows a sample of actual vibrations collected during an earthquake. It must also be noted that the signals captured by the Palert system undergo quantization, meaning the input values are mapped to a prescribed smaller set of values.

Table 1. Sample Palert Data

Time	a-axis	b-axis	c-axis
16:44:28.00	-0.1196	0.0	-0.0598
16:44:28.01	-0.05987	0.0	-0.0598
16:44:28.02	-0.0598	0.0	-0.0598
16:44:28.03	0.0	0.0	-0.0598
16:44:28.04	0.0	0.0	0.0

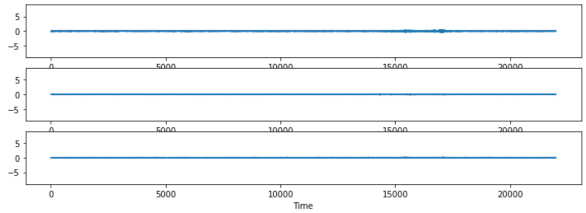


Fig. 2a. Ambient vibrations detected.

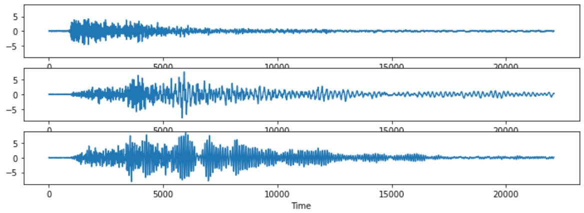


Fig. 2b. Actual vibrations from an earthquake.

2.2 Synthetic Data Generation with TimeGAN

TimeGAN was proposed in 2019 as a framework to synthesize time series data, and the TimeGAN model has 4 network components, namely the usual generator and discriminator networks (adversarial components), and the addition of an embedding and recovery networks as shown in Figure 3. The embedding and recovery networks act as an “auto encoder” to reduce the dimensions of the adversarial learning space and they are trained jointly with the adversarial components (Yoon et al., 2019). The readings from the Palert system were then preprocessed into sequence windows, and all numerical values were scaled accordingly prior to feeding it for training using the TimeGAN model. The hyperparameters of TimeGAN used are the following: the hidden dimensions were set to 24, batch size to 128, and learning rate to 5e-4. The sequence length was set to 24 which is the window size. Since GANs are notoriously difficult to train, this study experimented with 3 training sets, namely: 1) train on 1,000 steps, 2) 5,000 steps, and 3) 10,000 steps to observe the learning of the GAN model.

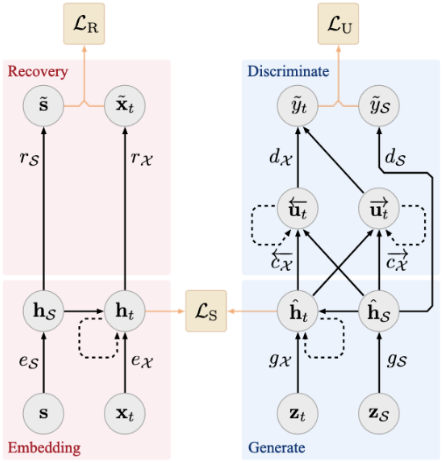


Fig. 3. TimeGAN network components (Yoon et al., 2019).

2.3 Model Validation

Both qualitative and quantitative metrics were used to evaluate the TimeGAN model synthesizer. PCA and t-SNE were used for dimensionality reduction, in order to visualize the data. The result of the visualization is used as a qualitative metric by observing the visual similarities between the real data and synthetic data. On the other hand, a train-on-synthetic, test-on-real (TSTR) approach was implemented using a separate recurrent neural network (RNN) regression model that was trained on both real and synthetic data. Both real and synthetic models were tested using data from the real test set to get the mean absolute error (MAE) and mean squared log error (MSLE) of each.

3. RESULTS AND DISCUSSION

Figure 4 presents the PCA and t-SNE visualizations of the real data (black) and the synthetic data (red) when the GAN model was trained on 1,000 steps with a window size of 24. It can be observed that the synthetic data is distinct from the real data in that the points of the synthetic data form very close points resembling “lines” as opposed to the sparseness observed from the real data; thus it can be said that the model has yet to fully learn the distribution. Moreover, Figure 5 shows a sample plot of how the synthetic values compare to the real values with a window size of 24 for the *a*, *b*, and *c* axes. It can be observed that the synthetic values show static data points close to the mid level. It is also necessary to mention that the values from the real data are quantized into a smaller set of values hence the gathered real data values from the aforementioned accelerograph portray a more discrete characteristic than a continuous one.

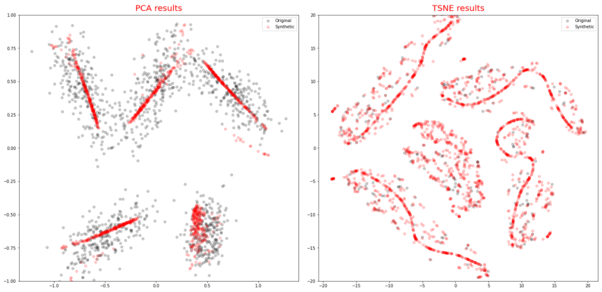


Fig. 4. PCA and t-SNE plots when trained on 1,000 steps.

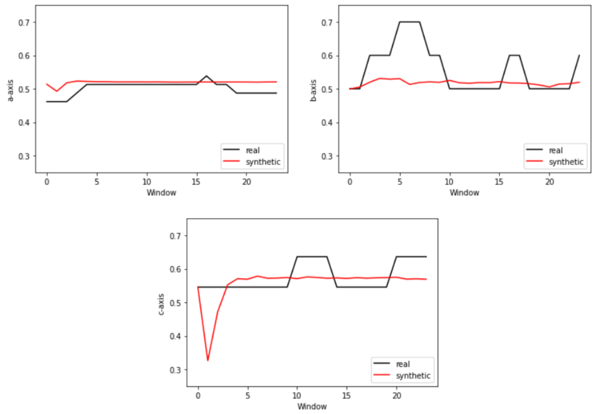


Fig. 5. Sample plot of real versus synthetic on *a, b,* and *c* axes (1,000 steps).

Figure 6 shows the PCA and t-SNE visualizations of the real data versus the synthetic data when the GAN model was trained on 5,000 steps with a window size of 24. It can be observed that the synthetic data presents a more distributed set of points and is closer to the distribution of the real data; qualitatively, there seems to be much improvement in the synthetic data when the number of steps is increased. Similarly, Figure 7 shows a sample plot of the synthetic versus the real values. It is interesting to note that all of the plots seem to be more dynamic as evidenced by a visually distinct set of peaks and valleys.

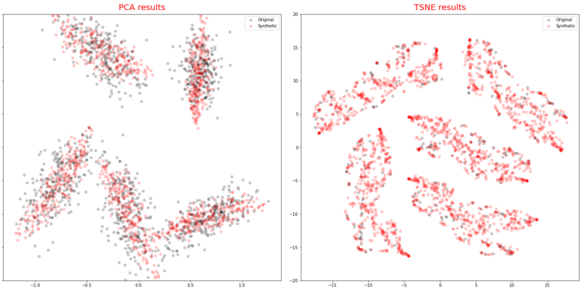


Fig. 6. PCA and t-SNE plots when trained on 5,000 steps.

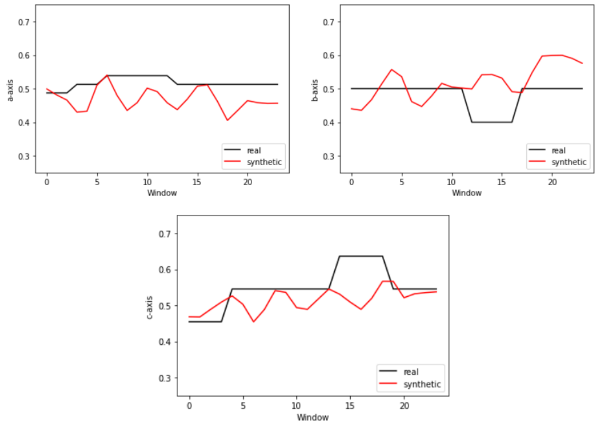


Fig. 7. Sample plot of real versus synthetic on *a, b,* and *c* axes (5,000 steps).

Figure 8 shows the PCA and t-SNE visualizations of the real data versus the synthetic data when the GAN model was trained on 10,000 steps with a window size of 24. It is interesting to see that the PCA visualization shows a strange phenomenon of “dense boundaries.” This can be attributed to the fact that since the data is solely from ambient vibrations, most of the data would be very similar in terms of minimal quantized vibrations. Moreover, Figure 9 presents the synthetic versus real sample values. Compared to the real samples, the synthetic samples show narrower peaks and valleys compared to the plot from the 5,000 steps, and this could be attributed to the fact that there is twice the amount of training steps than there were before.

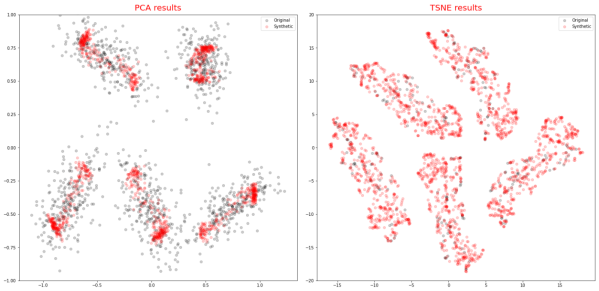


Fig. 8. PCA and t-SNE plots when trained on 10,000 steps.

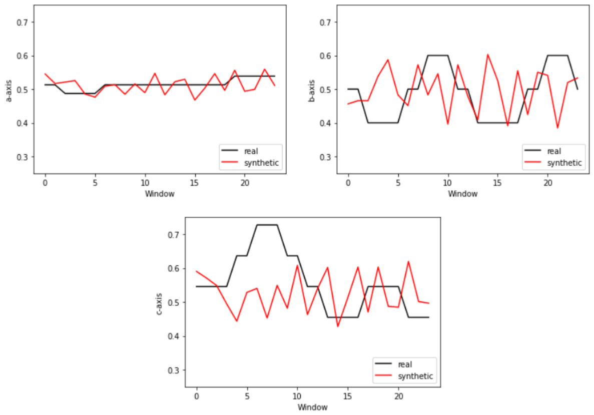


Fig. 9. Sample plot of real versus synthetic on *a, b,* and *c* axes (10,000 steps).

Other than the number of training steps, other hyperparameters may also be adjusted to determine the effectiveness in improving the model. It is worth exploring a larger window size as the comparison of real versus synthetic samples show that a window size of 24 is relatively small to have sufficient quantized values over a given sequence. This due to the fact that a small amount of data points makes it more difficult for the model to determine the trend in the time series data. Therefore, another GAN model was trained over a window size of 100, and it must be mentioned as well that 100 data points correspond to a 1-second reading in the context of the Palert system. Figure 10 presents the PCA and t-SNE visualizations when the GAN model was trained on 1,000 steps with a window size of 100. As opposed to having a window size of 24, it seems that the visual clusters that were formed previously were reduced as seen in the PCA and t-SNE plots; hence data points are distributed evenly based on the new data obtained from the result of the dimensionality reduction. Moreover, Figure 11 shows the sample plots of the real data versus the synthetic data, and it can be observed that the synthetic samples overlap closer with the real samples in terms of the peaks and valleys but with less quantized features.

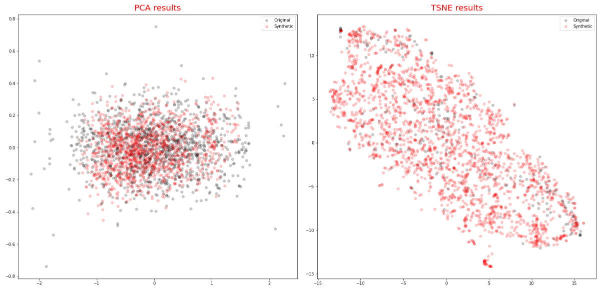


Fig. 10. PCA and t-SNE plots when trained on 1,000 steps and window size of 100.

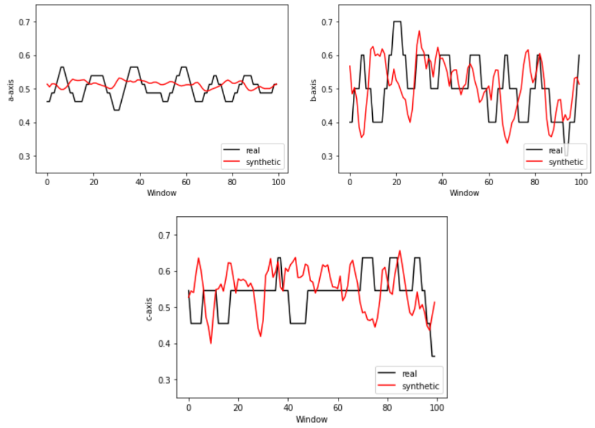


Fig. 11. Sample plot of real versus synthetic on *a, b,* and c axes (1,000 steps window size of 100).

Quantitatively, the respective MAE and MSLE were computed as shown in Table 2. It is intriguing to note that the values are close to 0, however it can be observed that the synthetic MAE and MSLE have larger values than that of the real MAE and MSLE. This could be attributed to the fact that the synthetic data does not use quantization, thus becoming more erratic in general. It is also worth noting that increasing the window size yields a slightly lower error rate score than increasing the training steps as seen with the synthetic MAE and MSLE, having smaller values on a window size of 100 than that of the real MAE and MSLE on a window size of 24.

Table 2. MAE and MSLE scores

	Real MAE	Synthetic MAE	Real MSLE	Synthetic MSLE
1,000 steps; win size=24	0.009850	0.026223	0.000314	0.000765
5,000 steps; win size=24	0.010157	0.036305	0.000322	0.001346
10,000 steps; win size=24	0.010221	0.035479	0.000316	0.001383
1,000 steps; win size=100	0.009968	**0.026139**	0.000314	**0.000738**

4. CONCLUSIONS AND FUTURE WORK

In this study, the use of TimeGAN for accelerometer-based monitoring systems was explored using the ambient vibration readings from an industry-grade Palert accelerograph. The data collected was segmented into windows and trained with different sets of iterations. The model was evaluated qualitatively using PCA and t-SNE visualizations, and quantitatively using the train-on-synthetic, test-on-real (TSTR) approach by training an RNN classifier for both real and synthetic data.

The goal for this paper is to synthesize ambient vibration data that is plausible, and qualitatively, the results show that synthetic data from TimeGAN captures the distribution of the real data. On the other hand, the MAE and MSLE values of both real and synthetic models from the TSTR validation show near-zero values; however there is still a difference in terms of the MAE and MSLE values of the synthetic model compared to the real model. It is also observed that increasing the window size seems to yield a better overall visual and numerical result, but this may be caused by over-generalizing the data due to the large amount of data that is summarized.

This study is part of an ongoing work and further experiments will be explored in terms of the introduction of actual and acted-out earthquake vibrations. This research will also utilize multiple accelerographs for monitoring smaller tremors and its effect on infrastructure.

6. REFERENCES

Alqahtani, H., Kavakli-Thorne, M., & Kumar, G. (2019, mar). Applications of Generative Adversarial Networks (GANs): An Updated Review. Archives of Computational Methods in Engineering, 28(2), 525–552.

Donahue, C., McAuley, J., & Puckette, M. (2019, feb). Adversarial audio synthesis. In the 7th international conference on learning representations, ICLR 2019. Retrieved from http://arxiv.org/abs/1802.04208

Dong, H. W., Hsiao, W. Y., Yang, L. C., & Yang, Y. H. (2018, sep). Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In 32nd aaai conference on artificial intelligence, AAAI 2018 (pp. 34–41). Retrieved from http://arxiv.org/abs/1709.06298

Engel, J., Agrawal, K. K., Chen, S., Gulrajani, I., Donahue, C., & Roberts, A. (2019). Gansynth: Adversarial neural audio synthesis. Retrieved from https://openreview.net/pdf?id=H1xQVn09FX

Esteban, C., Hyland, S. L., & Rätsch, G. (2017, jun). Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. Retrieved from http://arxiv.org/abs/1706.02633

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. Bengio, Y. (2014, jun). Generative Adversarial Networks. Retrieved from http://arxiv.org/abs/1406.2661

Hazra, D., & Byun, Y. C. (2020, dec). Synsiggan: Generative adversarial networks for synthetic biomedical signal generation. Biology, 9(12), 1–20. doi: 10.3390/biology9120441

Torres, D. G. (2018). Generation of Synthetic Images with Generative Adversarial Networks. , 57. Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:bth-15866

Vaccari, I., Orani, V., Paglialonga, A., Cambiaso, E., & Mongelli, M. (2021, jun). A generative adversarial network (GAN) technique for internet of medical things data. *Sensors*, 21(11). doi: 10.3390/s21113726

Yoon, J., Jarrett, D., & Van Der Schaar, M. (2019). Time-series Generative Adversarial Networks (Tech. Rep.). Vancouver: NeurIPS.