**Journal Report 13**

**11/17/23**

I spent class time researching the different parameters of the TimeGan model, which I think is the root cause for the problems I get with training it. My data is in a time series that consists of approximately 3,000,000 points, while the example code utilized a Yahoo stock dataset that consists of a time series that is approximately 800,000 points. These are some of the brief notes that I wrote down about each model parameter:

* + Batch size: It defines the number of data points used in each iteration of training. A larger batch size can lead to faster training, but it may require more memory.
  + Learning rate: It determines the step size at each iteration while moving toward a minimum of the loss function. A smaller learning rate can result in more accurate convergence but may slow down training.
  + Dimensionality of the noise vector: In GANs or VAEs, noise is often introduced to generate diverse and realistic outputs.
  + Dimensionality of hidden layers: It is relevant to the size and complexity of the model.
  + Latent dimensionality: This parameter is often associated with the dimensionality of the latent space in generative models. In VAEs or GANs, the latent space represents a lower-dimensional space where the model learns to encode meaningful representations of the input data.
  + Gamma: The role of the gamma parameter can vary depending on the context. In some models, it could be a regularization term or a hyperparameter that influences the overall behavior of the model. A gamma of 1 suggests equal importance, but the specific impact depends on the model's architecture.

**11/20/23**

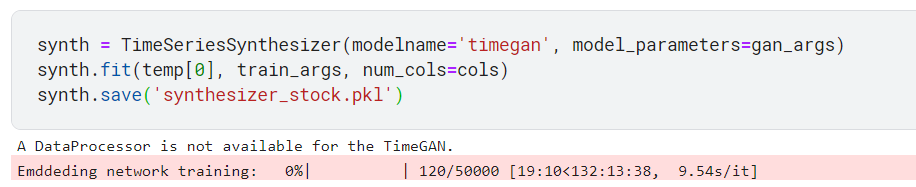
I was not at school, but worked on the project for 2 hours on my own time. I think the problem may lie in the “trainable parameters.” Specifically, the sequence length and the number of sequences are too small for my time series and will result in astronomically long training time.

train\_args = TrainParameters(epochs=50000,

sequence\_length=24,

number\_sequences=12)

I tried different values, but I haven’t been successful in finding a set of values that even start the training of the model. I continuously get the same error from last week:



**11/27/23**

I spent the class troubleshooting what could be going wrong. I graphed a random time series from my dataset and cut off the unimportant patterns. This includes the startup stage, where there is a period of all values being 0 in the startup of the Closed Cycle Refrigerator, and the end stage, where there is a period of all being zero after the neutron beam experiment is done. I want to see how GAN performs on my dataset even if it's a small time frame. As a result, I am thinking about cutting down my dataset and making the time series the exact size of the Yahoo Stock dataset that the example code uses. If this does not work, I may have to try to find another implementation. I am thinking about implementing a Variational Autoencoder (VAE) for next time.

**11/29/23**

I spent the class further processing my dataset to look like the one given in the example code. This time I took out all variables except for *Set B-field, Voltage,* and *B-field.* Then, I changed the corresponding values within the trainable parameters back to the state shown in the example code. After running it, I still get the same errors. I think Kaggle lacks the computational power to run this code, but I am not sure. As a result, I put my code into the server at the NIST, which has a plethora of GPUs and CPUs for training. I didn’t get a chance to run it yet and hope to do it next class.

Afterwards, I spent some time researching VAEs. It turns out that they don’t require labeled datasets, but are usually used for images. There have been some implementations for time series forecasting and creation of synthetic data, but I need to do further research as to whether its a viable option. As I am time-constrained, I might come back to this GAN training and set up a transformer model for anomaly detection with my current dataset first.