**Journal Report 15**

**12/8/23**

I spent some more time researching Self Organizing Maps. Then, I pickled the preprocessed data from last time and looked into more preprocessing for anomaly detection. I don’t do normalization, but I realized that it may be necessary as my various time series do not share the same time axis. For example, some series start at 8:00 AM, while others start at 2:00 PM. I plan to normalize it so that the maximum value is 1 for each of the training series. I plan to do this by using the Darts scaler to divide each of them by their largest absolute value. Doing this type of scaling does not impact the sMAPE, so for simplicity I will work with the scaled series only here. Later, I will have to call scaler.inverse\_transform() on our forecasts to translate them back to the original domain.

**12/11/23**

During my class session, I dedicated my time to researching the creation of a model and determining the type of model to develop. I opted for the "classical" approach outlined by Darts, wherein I plan to fit a distinct model for each time series. Subsequently, I will devise two concise functions: "eval\_forecast()" will calculate the median sMAPE error for all forecasts and display the distribution of errors, while "eval\_local\_model" will iterate through all series, construct a (local) model, train it on the series' training segment, and store the forecast. This function will then invoke "eval\_forecasts()" to showcase the sMAPE errors across all series. The output will include both the list of errors and the total elapsed time, providing a baseline for evaluation. It is customary to assess the performance of a (very) simplistic model that merely repeats the last value of the training series. I plan to implement this using a NaiveSeasonal model in Darts, with the implementation scheduled for next week.

**12/13/23**

I created a 0.6/0.4 train-test split with my dataset. Specifically, the first 0.6\*len(time\_series) timestamps of each time series was the training set and the rest were part of the testing set.

I created a timeline for the steps of my project:

1. Test that forecasting is working on just one time series for forecasting
   1. Is it possible to predict based on one variable for the test set? This would allow the model to predict solely on the set B field to forecast the other variables.
   2. See how well it works with combination of variables too (how do I make the model look at set point when predicting)
2. See which variable combination works the best for forecasting B-field and voltage
3. Learn how to train the model on multiple time series and how to forecast more than one variable

Afterwards, I created a new notebook for a model that I will train only on one time series. I want to go slowly because my goals have been too ambitious currently. This is what the dataset I chose looks like:

