**Journal Report 26**

**03/15/24**

I spent the class time cleaning up my pipeline, which includes receiving data input, forecasting data for a given time segment, and utilizing the filtering model to pinpoint anomalies. For the first 40 minutes of class, I spent my time reorganizing my data into a new Jupyter Notebook and overall ensuring that the process for training and testing was streamlined. Afterward, I spent the last section of the class researching the different hyperparameters of the model, while looking at the source code of the FilteringAnomalyModel to enhance performance and understand the inner workings of the model.

These are the main points that I gathered from my research:

* Window Size: This parameter determines the size of the moving window used in analyzing data. A larger window size may capture longer-term patterns but might miss short-term anomalies, while a smaller window size may be more sensitive to short-term changes but may also introduce more noise.
* Kalman Filter: The Kalman filter is a mathematical model used to estimate the state of a system from a series of noisy measurements. Integrating this into the anomaly detection model could enhance its ability to distinguish anomalies from normal variations by incorporating state estimation.
* Lower Threshold: Setting a lower threshold helps define the minimum level of deviation from the expected distribution that should be considered an anomaly. This parameter is crucial for fine-tuning the sensitivity of the model to detect anomalies.
* Upper Threshold: Conversely, the upper threshold establishes the maximum level of deviation deemed acceptable before flagging an anomaly. Adjusting this parameter allows for controlling false positives and false negatives, ensuring a balance between sensitivity and specificity in anomaly detection.

**03/18/24**

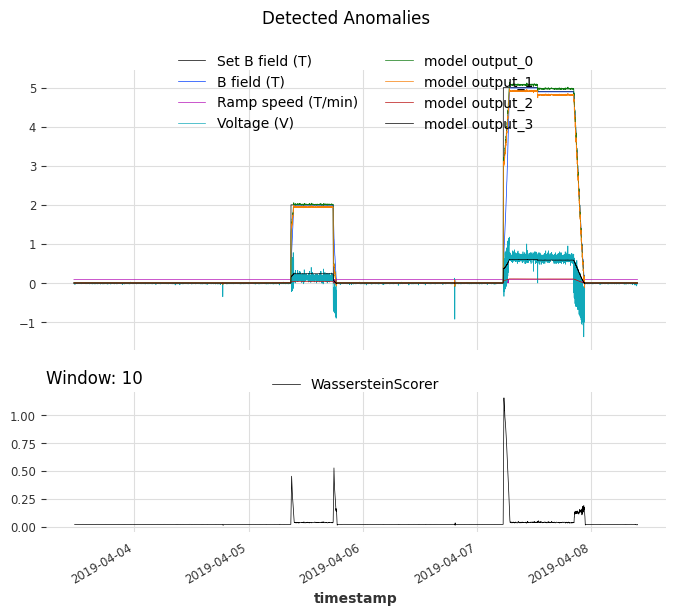
I spent the whole class period training my filtering model on different time series. After training the model on each time series, I graphed the results and corresponding time segments that were identified as anomalies. This was a very tedious process, but I was able to get an understanding of what kind of instances the model was predicting as anomalous. Specifically, the model did not understand that a change in the Set B field indicated a drastic change in other variables. This led the model to predict user changes as anomalies. I hope to fix this problem and make the anomaly segments smaller in my data.

**03/20/24**

I spent the majority of my class time hyperparameter tuning the thresholds in the ThresholdDetector and window size in the KMeansScorer. The training itself took 5-10 minutes for every hyperparameter modification. Throughout the waiting periods, I researched more anomaly scorers to enhance my FilteringAnomalyModel as I noticed that the KMeansScorer is sensitive to the noise present in the variables, especially the voltage signal. I came across a scorer called WasserteinScorer, which looks promising.

**03/22/24**

I spent this class time further implementing another scorer called the WasserteinScorer. This scorer first creates a training distribution by applying a moving window to the input series and stores the resulting vectors. Then, it computes anomaly scores by comparing the Wasserstein distance between the training distribution and each vector obtained from the input series. Following the implementation, I wanted to combine both the KMeansScorer and Wasserstein Scorer, but am running into some problems in regards to feeding multiple scorers and getting one prediction from feeding these two inputs into the model.



**Spring Break + 4/3/24**

Over the spring break and during class time on Wednesday, I continued to optimize my hyperparameters. This was a very manual process as I had to continuously train and compare results among different hyperparameters. I couldn’t find an optimal metric as I lack labeled anomalous data for long-term trends. I started working on creating some more labeled data for point anomalies and am thinking about how to label and simulate long-term anomalies for my testing dataset. Furthermore, I spent time getting the WasserteinScorer and KMeansScorer to work in tandem when detecting anomalies. I am running into the error shown below and I hope to fix this by Friday.

