Multivariate Time Series Forecasting & Automated Anomaly Detection

Isaac Burmingham

Data Science Term Capstone

1 Introduction

Our daily lives are surrounded by environments driven by time. From our house's power consumption, to satellites orbiting the Earth, time series data is vitally important to understanding our environment. From understanding time series data, we can understand how to predict values into the future, providing insights to the behaviors of the system. With these predictions, we can determine prediction errors, which in itself can lead to key outliers in the data. These outliers can be identified as anomalies, if they are uncharacteristic of the dataset, or uncharacteristic of the context of the dataset. Identifying anomalies within time series data, can be a crucial element to understanding the behavior of the system. For example, an identified anomaly from a spacecraft sensor, could indicate sensor failure, or drastic environment change. Another example, would be household power consumption greatly increases due to COVID-19 lockdowns, as a result of being in the house more. Anomalies are a key to understanding behaviors of the system and can lead to valuable information that can further influence decisions.

The analytical technique that will be used to predict time series ouputs and automatically detect anomalies are LSTM (Long Short-Term Memory) neural networks and Nonparametric Dynamic Thresholding. Nonparametric Dynamic Thresholding is a method proposed by NASA JPL in their GitHub project called *Telemanom* as well as their accompanying paper, *Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding* [1]. This project aims to analyze four different datasets across different industries, and use the methods described above to forecast the time series data and detect anomalies.

2 Data Description

Four multivariate time series datasets spanning multiple industries were used in this project. It is important to recognize the fact that they are multivariate, or contains more than one variable, and consider the feature interactions of the system. The datasets are as follows: satellite sensor data from LASP [2] (4 inputs, 4 outputs), simulated rocket thruster data [5] (2 inputs, 2 outputs), household appliance energy consumption [4] (27 inputs, 1 output), and household power consumption with weather features [3] (13 inputs, 1 output). All four of these datasets span different time frames, from 0.01 seconds to years and came in a variety of different formats that required different levels of data wrangling to produce the desired dataset output for model training. See visualizations of target variables in the Appendix. (See Figures 1-4).

3 Data Exploration

3.1 LASP Satellite Data

The LASP Satellite data originally came as four univariate files with the timestep and variable output. It was confirmed in a separate academic paper [2] that the data all came from the same satellite orbiting the Earth, and were sensors on this satellite. Therefore, it was logical to piece this information together in order to try to better understand the whole picture of the satellite. Although it can be inferred that not all the sensors were taken into consideration (as satellites often have 1000s of sensors) it at least attempts to address the feature interaction that could be present on such a spacecraft.

The original data had a time cadence of 5 minutes. This was too granular for our approach, so it was resampled to a 3 hour cadence. Any subsequent missing values from mean resampling were linearly interpolated to avoid any holes in the dataset. Once the data was correctly formatted, the feature interactions were analyzed with a heatmap to ensure there were no spurious correlations (See Figure 5). The correlations didn't show anything alarming so no action was needed. The features were also tested for any correlation with the Time variable, which would cause negative effects on the prediction model, but no correlations were found. Then, the dataset was scaled using a min-max scaler, and then split the data into 'features' and 'label' values where the label was how far into the future we wanted to predict. For this dataset, we chose to use the previous year (features) to predict all four outputs 1 day into the future (label). This was then used to be passed through a time series generator. The time series generator works by creating lagged datasets of your dataset, in order to turn the multivariate dataset into a supervised learning problem. Without it, we wouldn't have a target variable to predict. With the generator, we create a dataset, where our target is the next 'label' in the dataset and so on and so

forth. (See Figure 6). Once the time series generators were created for both the train and test set, the the model could be created and the data could be processed.

3.2 Household Weather Power Consumption Data

This dataset came in the format of two files, one corresponding to the target value of power consumption and its corresponding date, day of the week and notes about that day (COVID lockdown, weekend, etc.) where the time cadence was 1 minute. The other file contained weather data for the location of the house, where the time cadence was daily. In order to use this information together, one of the datasets had to be resampled so that both time series datasets were on the same scale. Thus, the power was mean resampled to the daily cadence, so that the weather data could be merged. The weather dataset had a lot of repetitive information, such as temperature max, min and average for the same day. We opted to only keep the max variables, as this is how weather is usually discussed, and dropped the other features to mitigate overfitting. With this correctly merged, we analyzed the correlation matrix (See Figure 9). and found no significant correlations. Additionally, in order to consider the 'notes' column that contained information about that specific day, we need to one-hot encode the variables. One-hot encoding of categorical variables turns the categories into their own binary columns. For example, our column of "notes" had values of "COVID lockdown", "weekday", "weekend", "vacation". These variables will be assigned to their own column, where it is a binary numerical value. This is useful because it lets us take the categorical information from the dataset and still be able to use it in our model. Lastly, we pass the numpy arrays through a time series generator as described above, choosing to

lag our datasets by 7 days, in order to predict 1 day (144 values) into the future.

3.3 Simulated Thruster Data

The thruster dataset is a simulation generated dataset of what the thrust and mass flow rate of a rocket engine doing multiple tests at different inlet pressures would be. The detail that it is a simulated dataset means that it is likely to not include anomalies, as these are spurious events that happen that likely a computer cannot simluate. Regardless, it is a useful dataset to use, to test how our model and process does at trying to find anomalies within the data. The dataset originally came in the format of 45 individual files, where each file was a test run of the rocket. Originally, and described in the data exploration notebook, it was attempted to try and merge these data files together, but ultimately was unsuccessful due to the large gaps of time between the runs. Therefore, it was only plausible to run on one dataset. The dataset is on a time cadence of 0.01 seconds, and has over 1B rows. Due to the sheer size of the dataset, mean resampling was chosen to better visualize the data, and also for better model learning. The data was reduced to 1 second intervals. Feature interaction was considered, and found that the features (thrust and mass flow rate) were very highly correlated (0.99, See Figure 8). This was a concern, but since they were the only two features in the dataset, the correlations negate themselves and be able to predict the other variable. Finally, the arrays were passed through a time series generator, choosing to lag the dataset by 1 minute, in order to predict 1 second into the future.

3.4 Household Appliances Energy Consumption Data

This dataset has a similiar context to the household power consumption dataset, where both are measuring the energy consumption of a household. Whereas the other dataset considered timely effects such as weekends and vacations, this dataset is focused solely on the weather, as well as temperature, humidity and other readings from within the house. In a similar fashion, the accompanying features will be used to predict the energy output of the household. This dataset was the only one that did not require extensive data wrangling, and was easily able to import the dataset. The feature interactions were visualized and analyzed, and found that there were a few features that had high correlations (See Figure 7). Two features, rv1 and rv2 had a 1.0 correlation, and from the given metadata said were random variables, so were subsequently dropped. Other variables like temperature and dew point were highly correlated, but since none were highly correlated with our target variable, we left the features in the dataset. The dataset was then passed through a time series generator, creating lagged datasets of 14 days, to predict 1 day ahead.

4 Model Theory & Application

Now that the data has been processed and has been passed through the time series generator, we can now build the nerual network to predict our target variables. We can do this through a type of Recurrent Nerual Network called LSTMs (Long Short Term Memory). LSTMs are distinguished against traditional Deep Neural Networks by their ability to maintain 'memory'. They continuously take information from prior inputs to influence the current input and output in the hidden layer. LSTMs are an improved version of RNNs (Recurrent Neural Networks), and are able to forget past information in addition to accumulating new

information through having gates regulate this in the LSTM cell. LSTMs are a great fit for anomaly detection because they can handle multivariate time series data without the need for dimensionality reduction, knowledge of the application and reduces the issue of vanishing and exploding gradients typically found in DNNs. In our approach, the model has one hidden LSTM layer, with variable number of units in the layer. In order to help mitigate overfitting, a dropout regularization was implemented, as well has a early callback. The model can be visualized here (See Figure 10). The number of units in the model depended on the complexity of the data, and how quickly it could learn the features. A full table of model parameters for each dataset can be found here (See Figure 11).

Once the LSTM generates predictions, the errors can then be calculated and smoothed, and Nonparametric Dynamic Thresholding as proposed by NASA JPL's paper [1] can be applied. In summary as described in their paper, a threshold is determined by evaluating if all the values above the threshold are removed, it would cause the greatest percent decrease in the mean and the standard deviation of the smoothed errors. As you continually step through the entire time series, your threshold dynamically adjusts, to account for the differences in errors and catch any anomalies.

With the anomalies classified, false positives can be mitigated through a pruning process. Since the threshold is only indicative of the mean and standard deviation of the smoothed errors, the false positive rate can tend to be high, which in a deployment setting could cause distrust in the algorithm. The algorithm can account for this via a pruning method. Pruning is done by calculating the percent decrease between two identified anomalies and comparing them to the set parameter, p, the minimum percent decrease. If the p calculated is less than the set p than the anomaly is reclassified as nominal.

5 Model Evaluation & Results

In order to evaluate the neural network models, loss curves were used to diagnose potential overfitting/underfitting concerns with the model as well as MSE (Mean Squared Error) and MAE (Mean Absolute Error) metrics were calculated to address the models accuracy. The below subsections will go through each dataset and discuss accuracy of given model and further considerations to improve predictions. A full table of the each models MSE scores and anomalies detected can be found in the Appendix (See Figure 12).

5.1 LASP Satellite LSTM

This dataset had 4 inputs and we aimed to predicted 4 outputs at once. We used a timestep of a year in our time series generator in order to predict a day in advance. This LSTM model had 250 units. The loss curve (See Figure 13, All loss curves can be seen here) showed a very strong fit and that the model was learning data well and quickly. We did not suspect overfitting as the train and validation loss curves were very close together and no sporadic movements. The MSE value was very close to 0, indicating that the model was accurate. The predictions were then reshaped and calculated, and plotted over the original data. (See Figure 14). With this, the model looked like it had trained accurately, so anomaly detection was the next step. The anomaly detection process looks at one sensor at a time to determine the anomalies within that sensor. Overall, for the first three features the algorithm seemed to fit the data well and detect points that did look anomalous although seems to have a high false positive rate. In some instances, it tries to identify anomalies that look to be trends in the data (See Figure 15). The last feature though, looked to have modeled the wrong feature. Recalling from the data exploratory section, the Current

feature and the Wheel temperature feature were highly correlated (0.8) and it looks as if the LSTM model is attempting to follow the trend of the wheel temperature values indicating it struggled to learn this feature although this was not evident in our diagnostics. Further evaluation and tuning of the model should be performed in order to mitigate the effect of this feature interaction.

5.2 Household Appliances Energy LSTM

The Appliances dataset has 27 features and 1 target output. We used the previous 7 days to predict 1 day ahead and this was passed through the time series generator. The LSTM model struggled at first with overfitting, but after increasing the dropout closer to 0.4, it seems that the model is better performing. The loss curves both exhibit a smooth elbow that converge closely together. Additionally, the MSE converges close to 0. Although the diagnostic evaluation looked healthy (See Figure 13), the predictions overlayed onto the orginal data did not look accurate. Although the predictions look to be underestimating, what matters here for anomaly detection is that the errors calculated do not deviate from the mean and standard deviation of the entirety of the errors. This deviation is what finds the anomalies, so even though it looks like our model didn't perform well, the diagnostic plots indicate a good model fit, and the anomalies detected do not seem far off, although being that it is unsupervised anomaly detection, it is harder to gauge our accuracy. (See Figure 16).

5.3 Simulated Thruster LSTM

The thruster dataset had 2 inputs, and 2 outputs. These were thrust, and mass flow rate. The LSTM model had 250 units, and learned the dataset quickly, indicating a sharp inital dropoff after the first epoch and loss was very close to 0 indicating a well fit model. MSE also looked strong, converging around 0.007.

With the predictions of the train and test overlayed on the original thruster data (See Figure 17). we see that the predictions match well according to the original data. After running the anomaly detection algorithm, there are no detected anomalies in the dataset (See Figure 17). This is unsurprisingly, as the data is simulated, so it would make sense that there are no anomalies that occur, because the simulation probably did not account for random anomalies when the data was created.

5.4 Household Weather Power Consumption LSTM

This dataset has 12 features and 1 target ouput. We are predicting the power consumed by the household based on the sensor readings inside the house, as well as the weather outside. This dataset after having the resample was by far the smallest dataset, only having 1200 rows. The loss curves look a bit more bumpy, but overall looks strong as the train and validation converge to a similar point. The prediction overlay for both the train and test sets look very strong and accurate (See Figure 18). With accurate predictions, the anomalies were then detected (See Figure 18). The chart looks to be detecting anomalies near the end of the time frame for the training set, which seems a bit peculiar. The errors must have increased as the prediction went on and gives cause to assess the accuracy of our thresholding method in further analysis.

6 Discussion

In summary, LSTMs seemed to be an adequate fit for predicting the values of these various datasets that broadly covered multiple industries. In addition, nonparametric dynamic thresholding seemed to be a strong technique for detecting anomalies in the data, although tough to understand how accurate the

thresholding is. It should be mentioned that the purpose of using this anomaly detection method was not to be exhaustive, but rather explore and understand the method as proposed by NASA JPL. For further consideration, a supervised learning dataset should be considered, to help address the accuracy of the anomaly detections.

The analysis provided has been able to provide key insights into the robustness of LSTM nerual networks, and the nonparametric dynamic thresholding method for detecting anomalies. Not only were we able to accurately predict multivariate time series data into the future, but were able to do so on various datasets of different genres and data features. This shows that there is a significant use case for building these models, and being able to understand them to not only predict, but also detect anomalies. For further investigation, hyperparameter tuning of both the LSTM model and nonparametric dynamic thresholding methods should be investigated more in order to fine tune the results and mitigate any errors that occured. Lastly, further mitigation of false positives should be considered in order to build operational trust for deployment into industry.

References

[1] Hundman, Kyle and Constantinou, Valentino and Laporte, Christopher and Colwell, Ian and Soderstrom, Tom *Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding* 2018 https://github.com/khundman/telemanom

[2] Polson, Shawn *Unsupervised Machine Learning for Spacecraft Anomaly Detection in WebTCAD* June 7 2019 https://github.com/sapols/Satellite-Telemetry-Anomaly-Detection

[3] https://www.kaggle.com/srinuti/power-consumption-time-series-analysis/data

[4] https://archive.ics.uci.edu/ml/machine-learning-databases/00374/

[5] https://www.kaggle.com/sylar68/spacecraft-thruster-firing-test-dataset

[6] Jain, Vedant https://databricks.com/blog/2019/09/10/doing-multivariate-time-series-forecasting-with-recurrent-neural-networks.html

[7] Brownlee, Jason https://machinelearningmastery.com/diagnose-overfitting-underfitting-lstm-models/

[8] Narayanappa, Sada https://github.com/sada-narayanappa/NNBook

7 Appendix

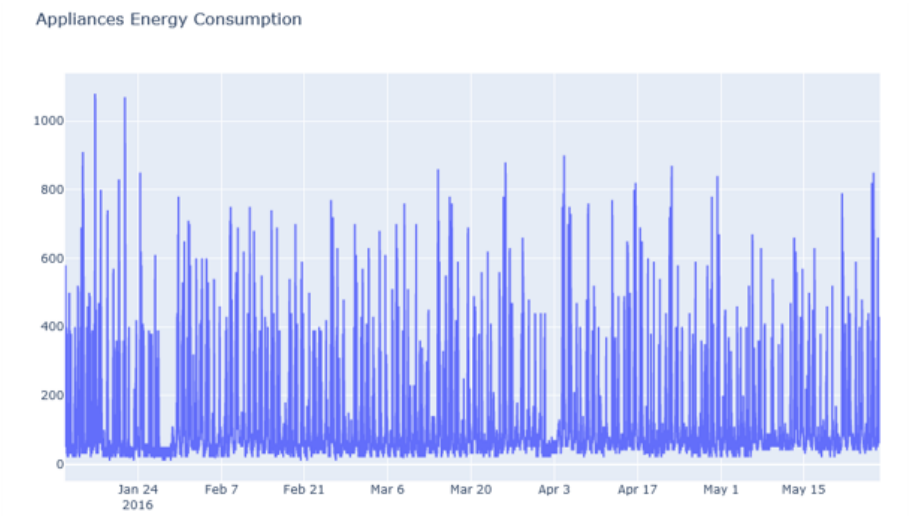

Figure 1: The Appliances Target Variable plotted over time

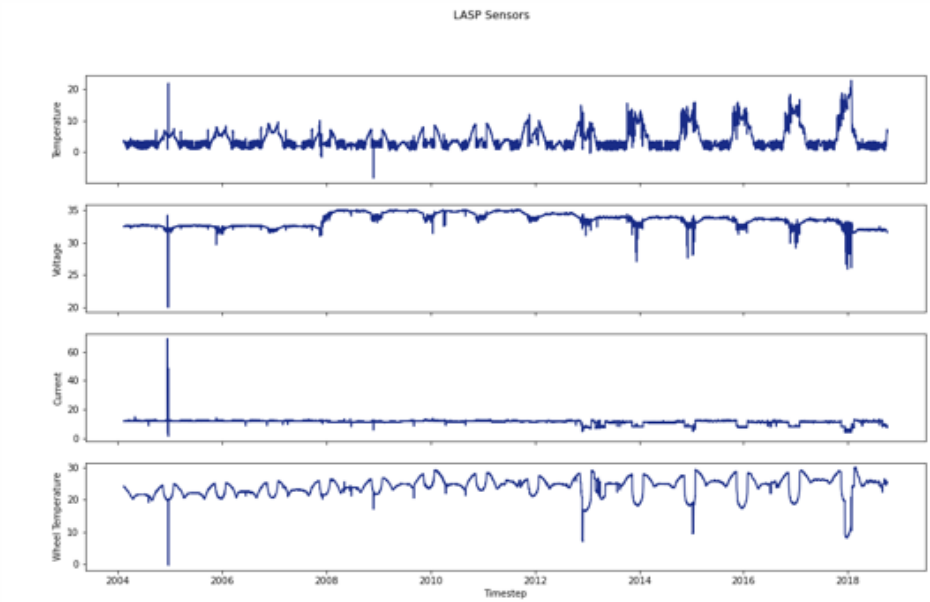

Figure 2: The four sensors on the satellite from LASP. The sensors are Battery Temperature, Voltage, Current and Wheel Temperature

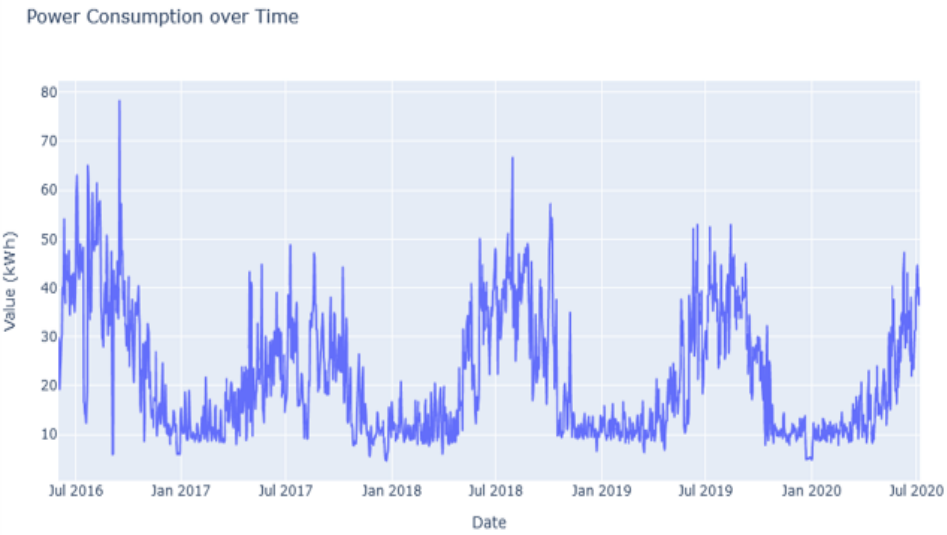

Figure 3: The Power consumption output of a household

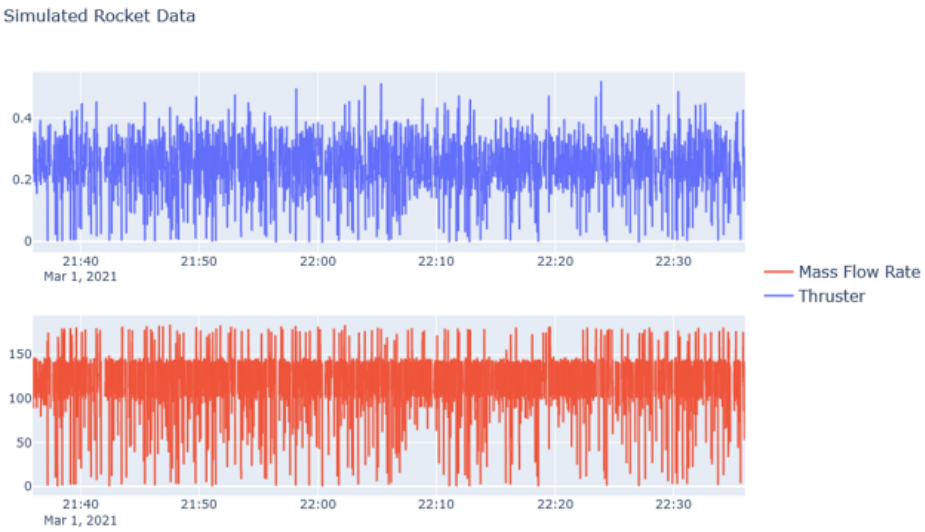

Figure 4: The two targets from the rocket thruster dataset, thrust and mass flow rate

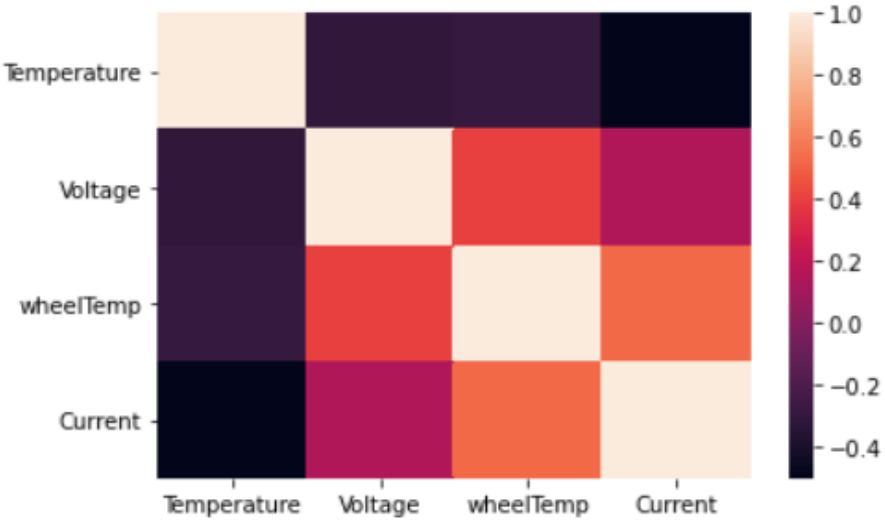

Figure 5: Heatmap of the feature correlations in LASP dataset. Only significant correlation is between Current and Wheel Temperature

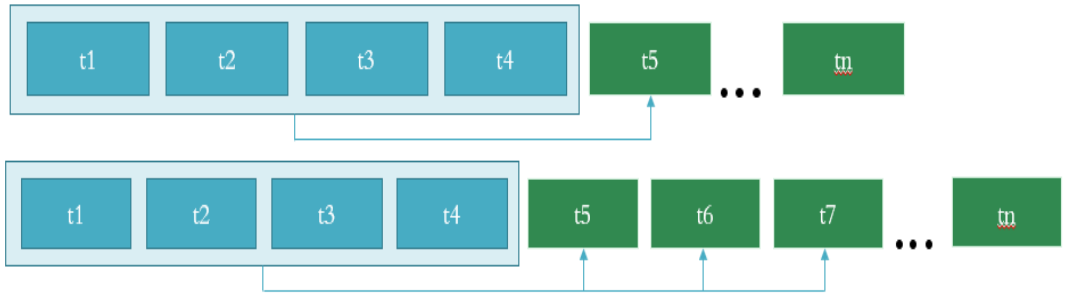

Figure 6: A basic Time Series Generator chart. The inputs are used as a subset of the data to predict the next timestep. The window then slides to predict the next timestep and so on. Alternatively, they can be used to predict a sequence of timesteps into the future.

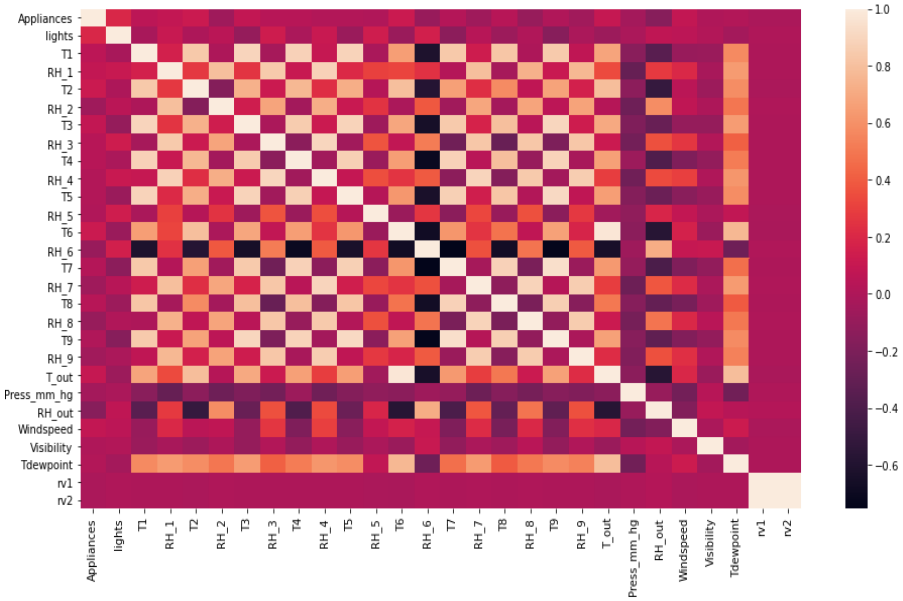

Figure 7: Heatmap of the feature correlations in Appliances dataset. The rv1 and rv2 variables in the bottom right were dropped. There are some other high correlations, but none with target.

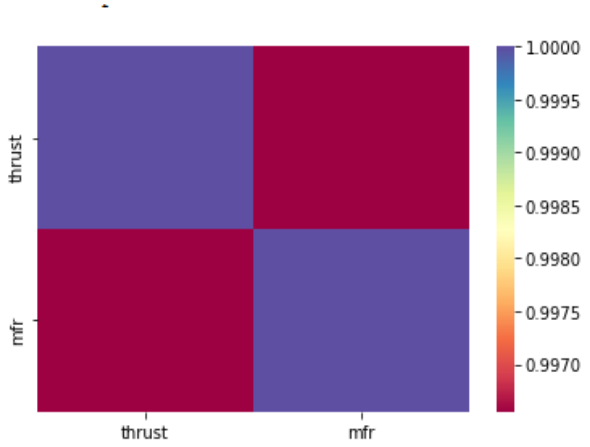

Figure 8: Heatmap of the feature correlations in Thruster dataset. The two features are highly correlated with each other

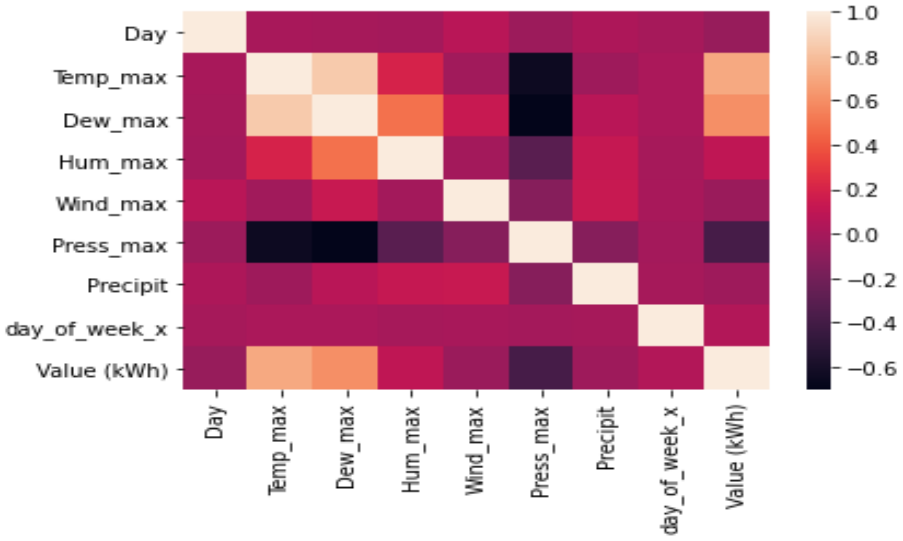

Figure 9: Heatmap of the feature correlations in Household power consumption weather dataset. A few notable high correlations, but nothing that is too high with the target

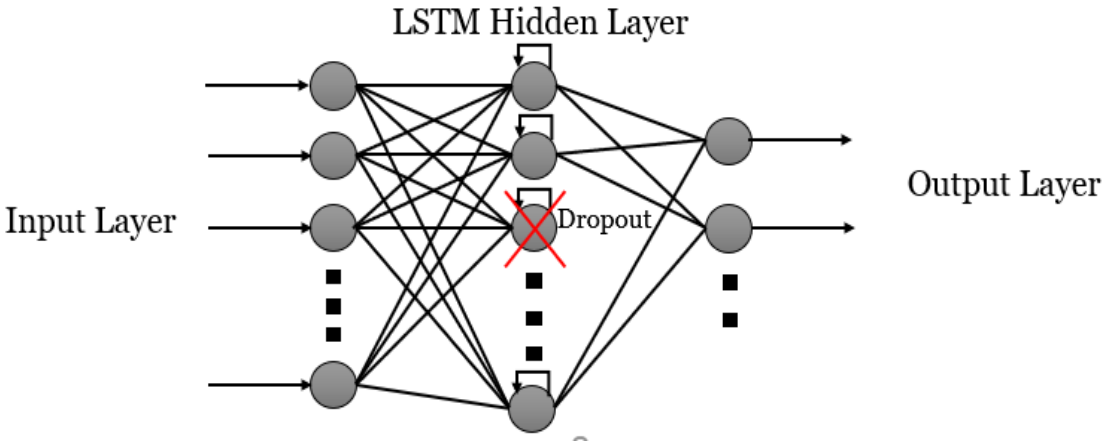

Figure 10: Neural network diagram for LSTM model

LSTM Parameters	Units	epochs	learning rate	dropout
Appliances	8	100	0.001	0.4
LASP	250	50	0.0015	0.3
Thrusters	250	100	0.001	0.1
Household Power Weather	250	100	0.001	0.1

Figure 11: LSTM Model Parameters for each dataset

	MSE
Appliances	0.019523
Appliances Validation	0.010384
LASP	0.006944
LASP Validation	0.005777
Thruster	0.007884
Thruster Validation	0.007777
Household Power	0.010285
Household Power Validation	0.008318

	Anomalies Detected
Appliances	1383
LASP	[2846, 630, 1774, 0]
Thruster	0
Household Power	209

Figure 12: MSE and anomalies detected results

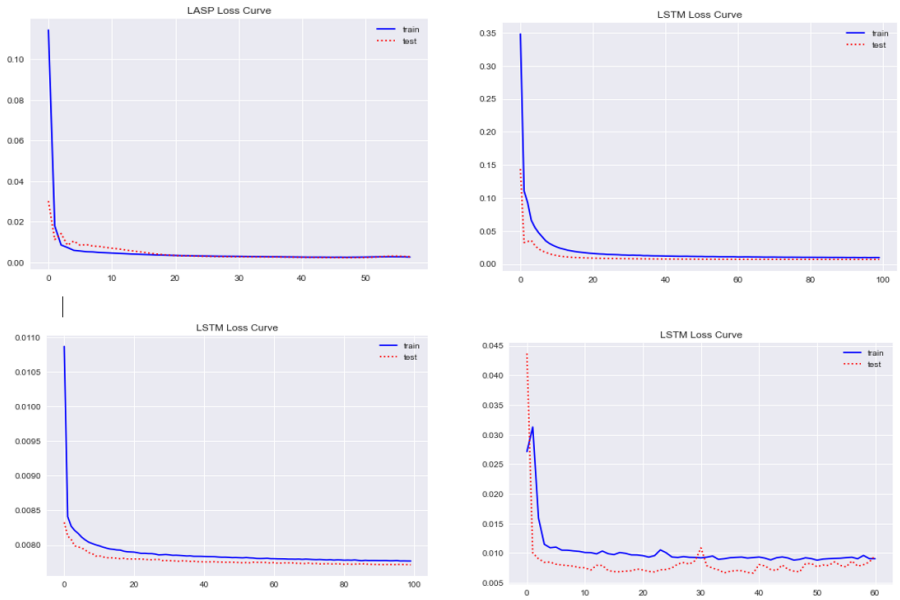

Figure 13: Loss Curves for all 4 models. From top left to bottom right: LASP, Appliances, Thruster, Household Weather Power

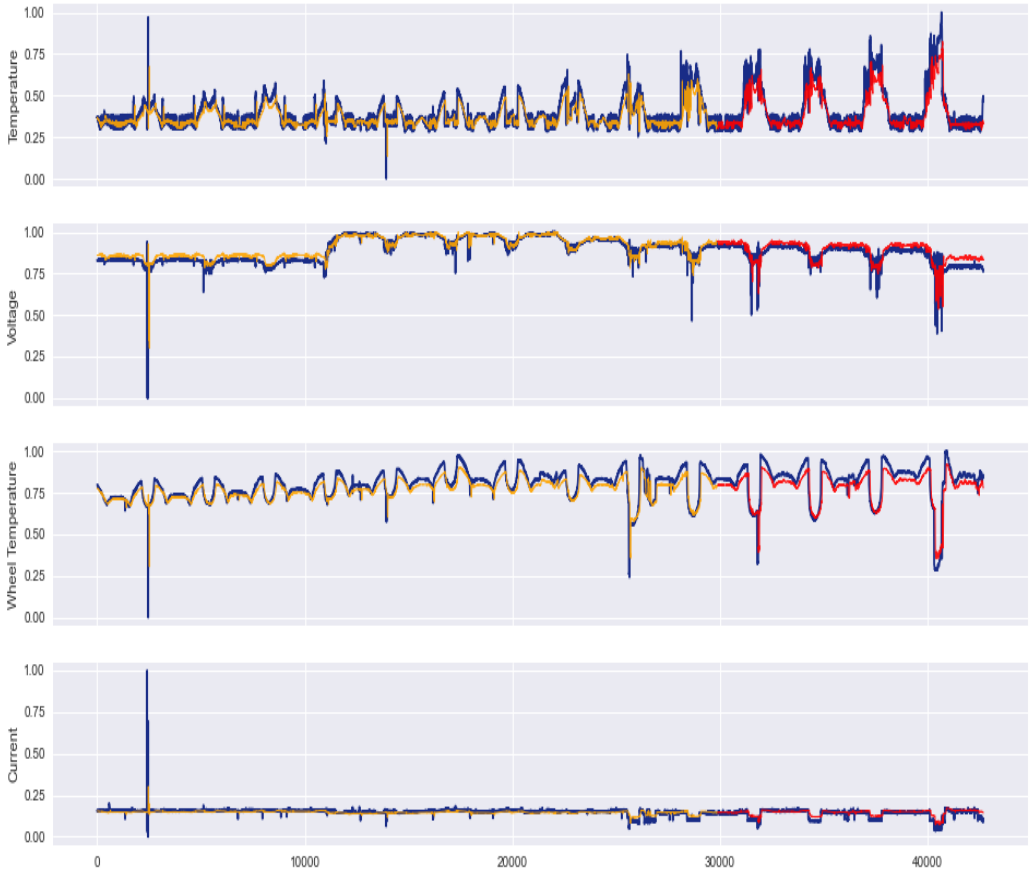

Figure 14: LSTM Predictions overlayed original data

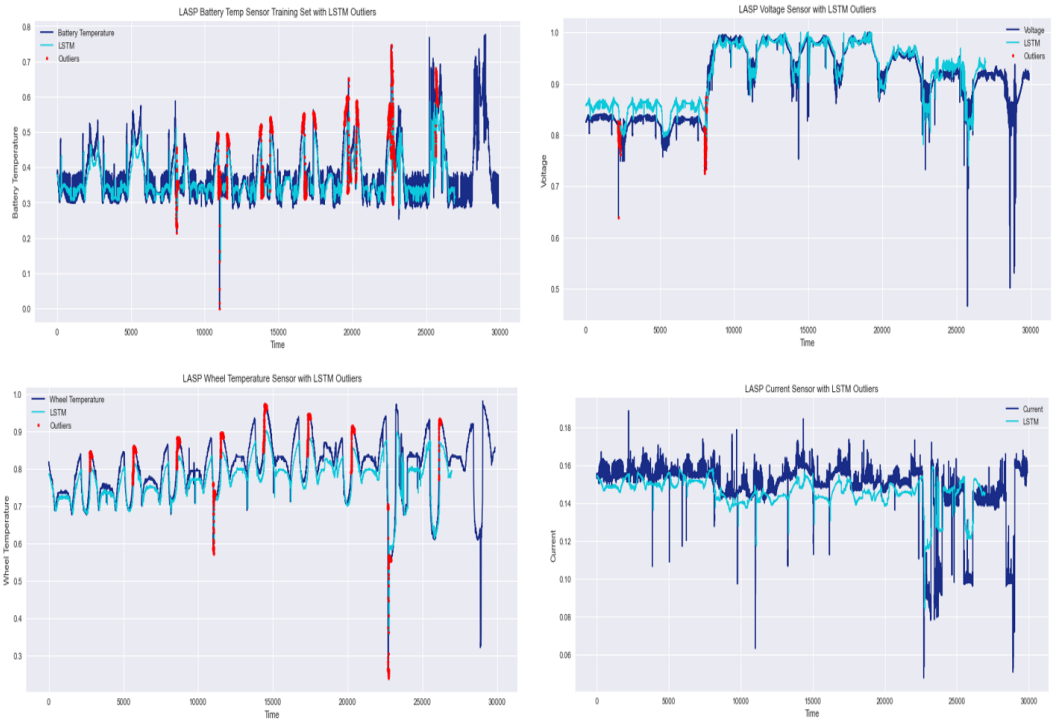

Figure 15: Anomalies detected in LASP datset

Figure 16: Green is train predictions, blue is test predictions, where red is the original data. Bottom chart shows train predictions, along with red anomalies detected

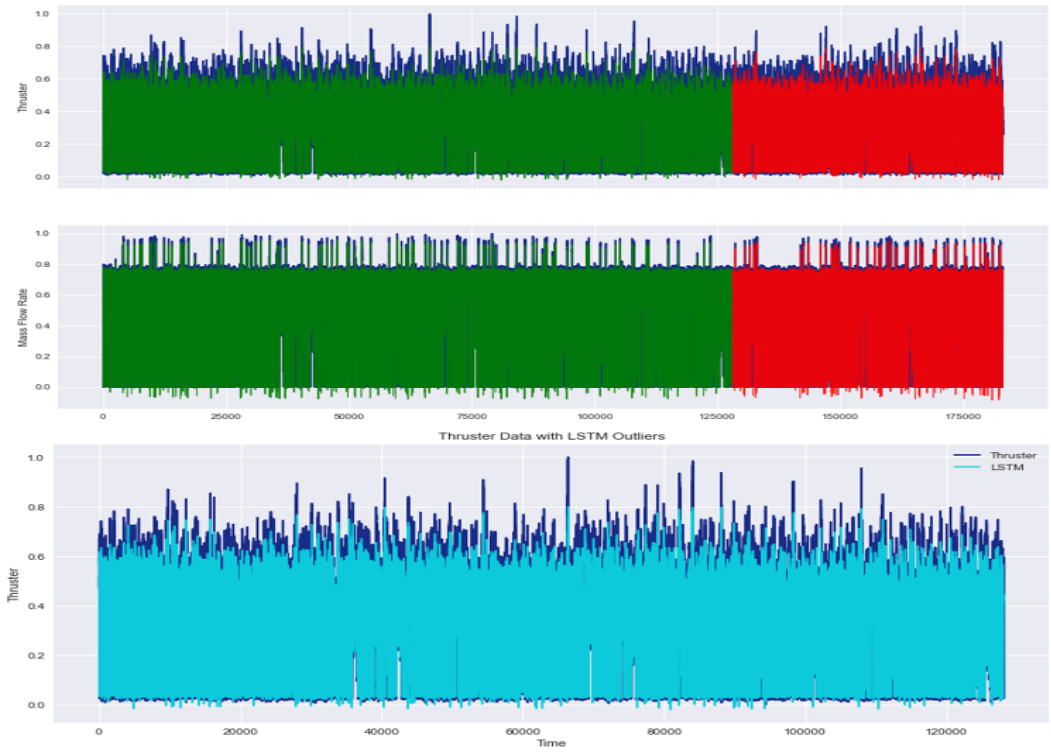

Figure 17: Green is train predictions, blue is test predictions, where red is the original data. Bottom chart shows train predictions, no anomalies were detected

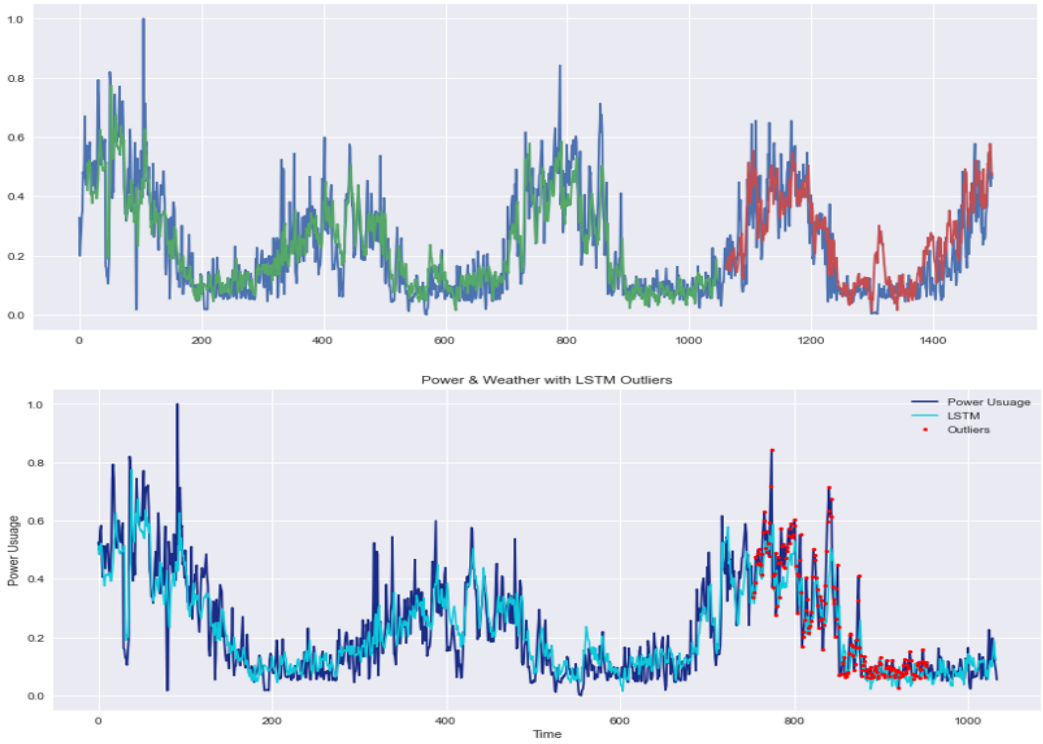

Figure 18: Green is train predictions, blue is test predictions, where red is the original data. Bottom chart shows train predictions, red shows anomalies were detected