Cyber Data Analytics - Assignment2

Prerak Mody - 4777042, Sujit Shankar Jaishankar - 4779657

May 2019

1 Task1 - Familiarization Task

The BATADAL Dataset consists of 43 columns of raw data (X) and one column with the attack flag (Y). The different types of signals (few examples in fig. 1) that exist are :-

- Continuous signals - Eg : tank sensors for water level (L_T1 to L_T7). A trend/seasonality (or lack of it) in such signals can be helpful in time series modeling methods such as ARMA and SAX.
- Binary Signals - Eg : Pump Actuator Status (S_PU1 to S_PU11) or ATT_FLAG (1's and 0's). Some sparse binary signals may be useful to detect anomalies via spectral methods such as PCA.

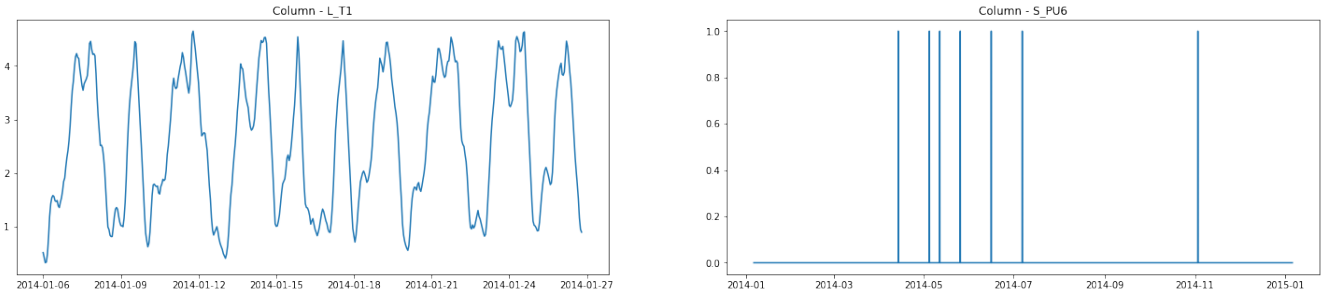

Figure 1: (Q1) Types of signals as seen in *BATADAL_dataset03.csv*

We can notice in fig. 3 that there exists the presence of both highly positive and negative correlations between signals such as :-

- Strong Positive Correlation - F_PU2 and S_PU2 with P_J269 (i.e flow and status of pump2 is positively correlated with the pressure)
- Strong Negative Correlation - P_J269 with F_PU1 (i.e pressure is negatively correlated with flow in pump1)

Predicting the next value via a time series prediction (using Linear Regression) outputs varying results (RMSE) on different signals due to their seasonality and amplitudes. We show a relatively easy-to-model signal in fig. 2

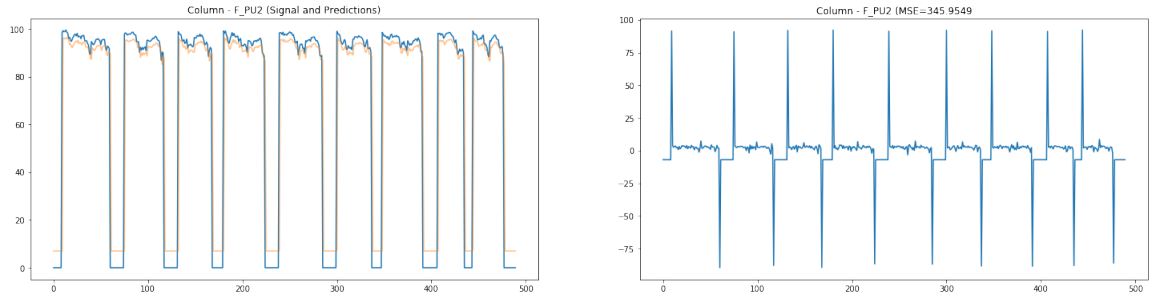

Figure 2: (Q1) Predicting time series values via Linear Regression

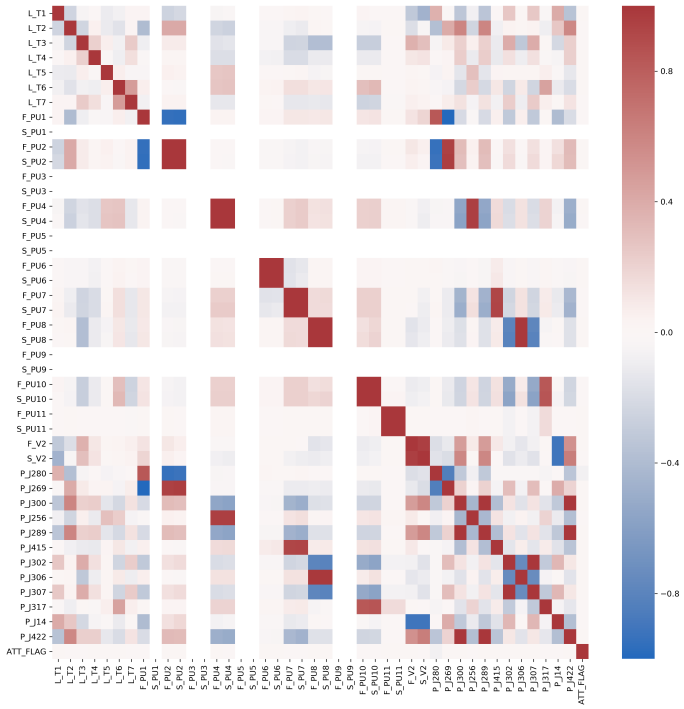

Figure 3: (Q1) Correlation between signals

2 Task2 - ARMA Task

In this task, we use the ARMA model to detect anomalies in the dataset. The datasets loaded are the first and second training sets. We then visualized most of the sensors and the variation in their value over time was looked at. For training we use the first dataset with normal data. For testing we use the second dataset, which contains normal and attack data.

The initial step if to check if the signal being tested is stationary. For this, the Augmented Dickey-Fuller Test is used. Using the ARMA model requires to test for stationarity. In order to determine the 'd' parameter (differencing) it is imperative to know if the signal is stationary. The autocorrelation and partial-autocorrelation plots and AIC statistics are then used for determining the optimal order (p,q) of the ARMA model. The ARMA model is then created using the first dataset. Based on the parameters of the model created, the ARMA model for the second dataset (test set) is created.

The residuals for the prediction model are then obtained. In order to categorize sample as anomalies or not requires the setting up of the threshold value. The threshold in this case is set as five times the standard deviation. The samples with residuals higher than the threshold are marked as anomalies. The final model uses 31 signals to learn ARMA. Certain signals could not be effectively modeled with inf AIC or singular matrices during fitting (refer notebook). Here we present plots for the signal 'L_T1'.

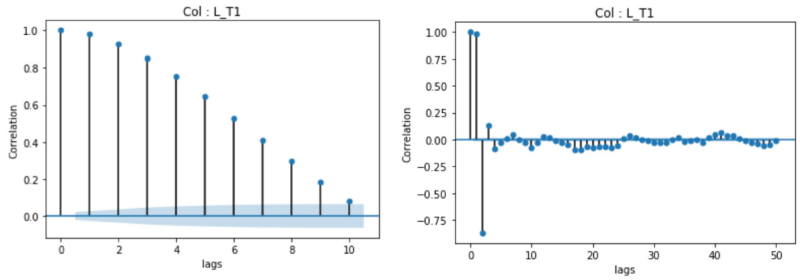

Figure 4: (Q2) - ACF and PACF Plots for Signal L_T1

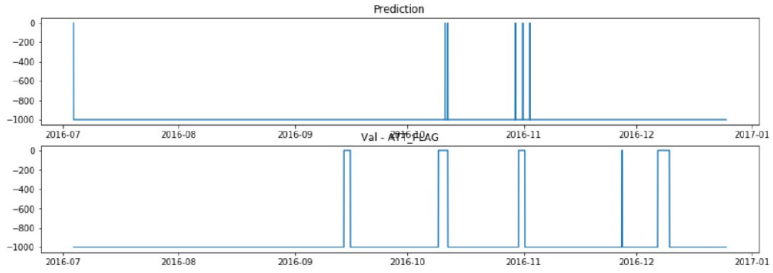

Figure 5: (Q2) - Prediction for Signal L_T1 with ATT Flag

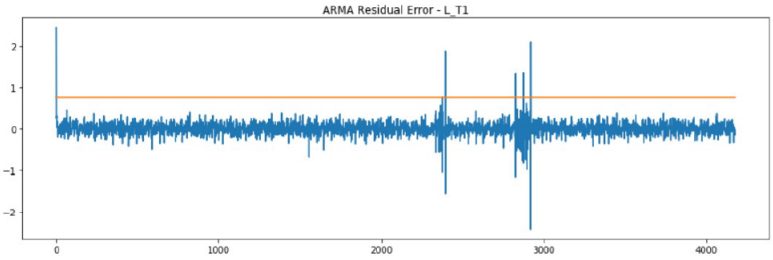

Figure 6: (Q2) - ARMA Residual Plot for signal L_T1 in Test set with threshold

The above plots were plotted for all signals for which ARMA could be applied. Figures for the sensors modeled can be found in the Jupyter notebook. The files ARMA_ACF(P)_Plots.pdf and ARMA_Output_Plots.pdf can be referred for associated plots of selected signals. For individual sensors, the modeling can be seen in the Jupyter notebook. For the complete model, the first dataset was used for training and tuning the parameters (p,q), and was tested then on the second dataset. Certain signals were modeled well as evident from their F_scores(refer Jupyter Notebook) - such as L_T1 and S_PU2, S_PU6, S_PU11.

It can be seen that sensors with better performance in terms of metrics, i.e, accuracy, precision, recall have been modeled more effectively. For some signals the predictions are not good. Moreover, modeling each signal for ARMA takes a significant amount of time.

Certain anomalies that can be detected by ARMA can only be done so if the anomalies occur with a short duration of time. If the anomalies linger, then due to the prediction based approach of ARMA, these will be considered as normal signals over a period of time. We can say that sudden anomalies are easier to detect for ARMA.

3 Task3 - Discrete Models Task

We chose the Symbolic Approximate Aggregation (SAX) method [1] which transforms a time series input to a high level representation (i.e strings). This method is quite useful since it reduces the dimensionality of the data and makes it easier to process. The original time series is first transformed into its Piecewise-Aggregate-Approximate (PAA) format (for dimensionality reduction using windows of size=10) and then utilizes a lookup table to convert the PAA data into a string using a predefined alphabet as can be seen in fig. 7

We apply the N-grams technique (N=3) to identify any anomalies. We consider an n-gram in validation dataset to by an anomaly if that n-gram was not seen even once in train dataset. Such an approach is able to detect volume

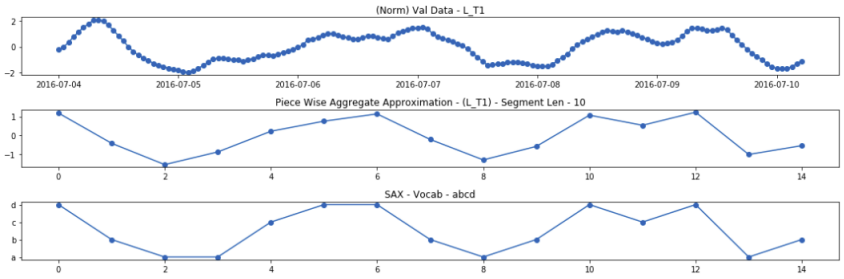

Figure 7: (Q3) Discretization using SAX

anomalies very well in signals such as ['F_PU6', S_PU6, F_PU7, S_PU7, 'F_PU10', 'S_PU10', 'F_PU11', 'S_PU11', 'P_J14', 'P_J302', 'P_J307']. Vizualizations for these signals can be found in *03_Discretization.ipynb*

4 Task4 - PCA Task

On applying PCA to the entire dataset, it is noticed that 14 components capture more than 99% of the variance in the data as can be seen in fig. 8. We also plot the residual errors on the train data when we project the data back to its original space. The residual results are shown in fig. 8 where we notice some abnormalities most probably caused due to outliers.

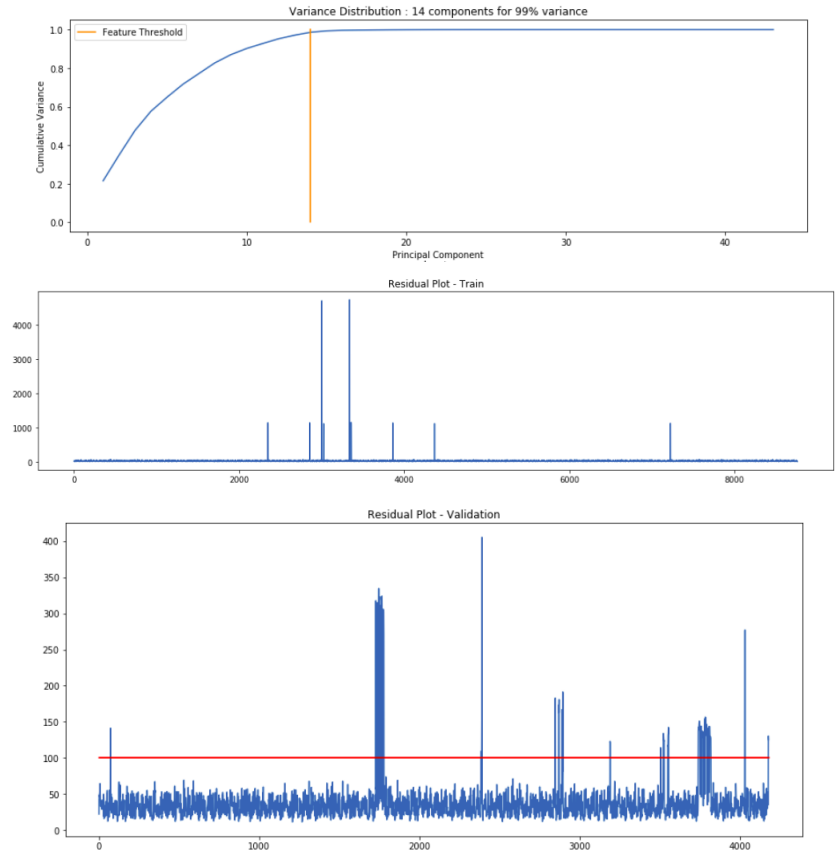

Figure 8: (Q4) The top two figure show our PCA analysis on train data and the bottom figure shows the residual errors along with threshold on validation data

We are able to detect point anomalies with PCA as can be seen in the jupyter notebook titled *04_PCA_Task.ipynb*

5 Task5 - Comparison Task

We decided to evaluate the systems based on accuracy, precision, recall and F_score.

With respect to ARMA, we find that it analyses majority of the signals but cannot for certain binary signals. It analyses multiple signals and combines the predictions made. There are certain downsides to it. It requires the signal to be stationary. Also, building separate models for each signal is time consuming especially in during training. An analyst perhaps refrain from using ARMA since it consumes a lot of time while at the same time not being able to model all signals.

The PCA model is straight forward. It reconstructs the data in principle components and calculates residual between original and restructured data. While most of the variance is captured by 14 components which seems to give us better results. The outliers will have variability in the later components and this way, anomalies can be detected. PCA is simpler to implement and faster than ARMA. It also works well here due to being a dimensionality reduction technique. The caveat however, is that the assumption that even in the reduced space the normal and abnormal instances and as distinguishable as in the original data.

For the discrete model, we use SAX. Again, each signal has to be analyzed separately but unlike ARMA, it was much faster. SAX may help overcome the reduced dimensionality disadvantage of PCA as mentioned earlier.

Appendix

References

[1] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing sax: a novel symbolic representation of time series," *Data Mining and knowledge discovery*, vol. 15, no. 2, pp. 107–144, 2007.