CS4035: Cyber Data Analytics - Assignment 2

Group 18: Shashank Rao (4624017), Apourva Parthasarathy (4597583)

June 6, 2017

1 Visualizations

1.1 Signal Types

The dataset contains 51 signals from sensors and actuators. The sensor measurements include water tank level, flow meters, water properties and pressure. The actuator measurements include water pump, motorized valves and ultraviolet de-chlorinator which have integer values indicating on or off. Figure 3 shows the correlation matrix for all the signals. Signals' heat map in red indicates that they are highly correlated. Hence, the sensors are correlated and are less or uncorrelated with actuators.

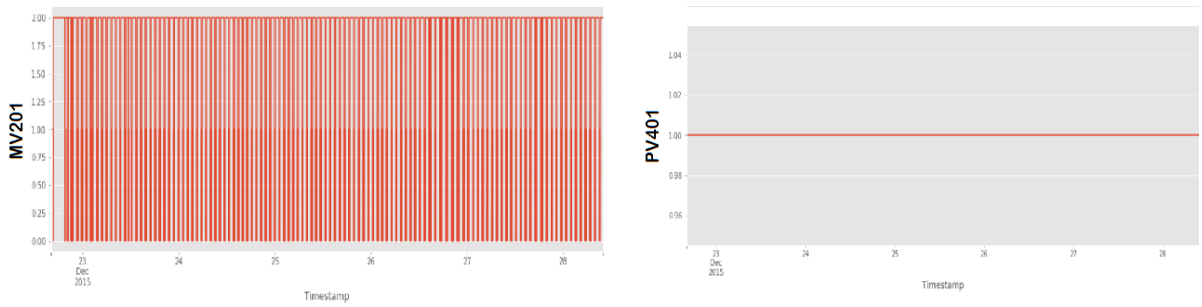

Figure 1: Actuator signals MV201 and PV401

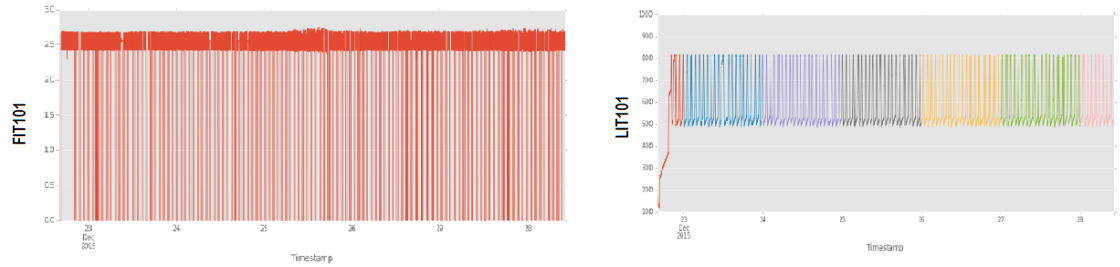

Figure 2: Sensor signals FIT101 and LIT101

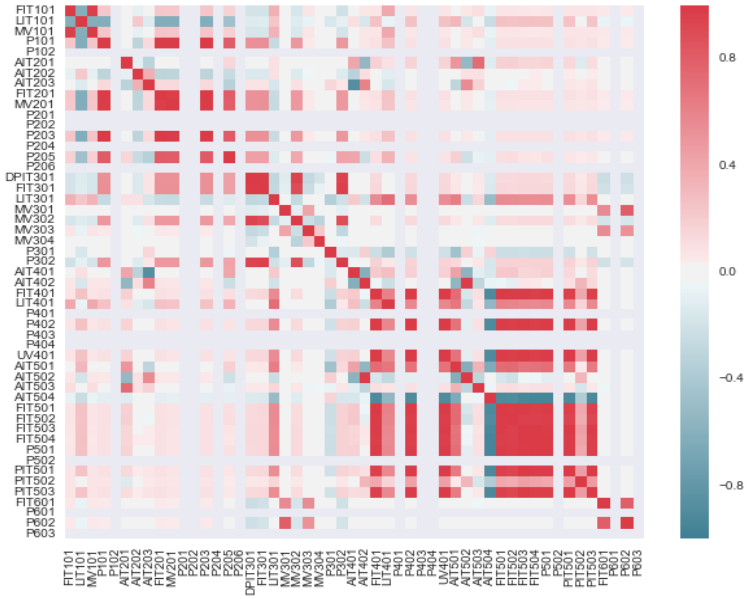

Figure 3: Correlation matrix heat map

2 Principal Component Analysis (PCA)

PCA is a statistical technique that maps data points to linearly uncorrelated principal components (PCs) such that the first PC explains maximum variance in the data, the second PC has second highest variance while being orthogonal to the first and so on. Hence, fewer dimensions are required to represent the data since the top few components explain large variance in the data. The PCs are then used to model to normal and anomalous subspaces which can be used to detect anomalies in the dataset.

2.1 Pre-Processing

- Normalize the data to contain zero mean and unit variance.
- The data collected on 22/12/2015 (Day 1) represents data when the system has not stabilized yet. Figure 2 shows the abnormalities in the data from Day 1. It is important to remove this data before training the model because we wish to model the normal behavior.
- The data from 23/12/2015 till the date of first attack is used as training data. The remaining data is used for testing.

2.2 Modeling Normal and Anomalous Subspaces

Figure 4 shows that the first 11 principal components are able to explain 90% of the variance in the data and are chosen to model the Normal Subspace. By plotting the projection of the signals onto the next 9 components, we see that the signals show a number of spikes which can be used to model abnormalities in the data. Hence, principal components 12 to 20 are chosen to model the Anomalous subspace.

2.3 Detecting Anomalies From Residual Vector

After modeling the normal and anomalous subspaces, we project the test data onto the these subspaces. The abnormal changes in the residual traffic indicates the presence of anomalies. This can be detected by computing Squared Prediction Error (SPE) [1] for the residual vector and classifying data above a

threshold as anomaly. Figure 5 shows the residual vector obtained by projecting data onto anomalous subspace. The threshold is tuned to find an optimal trade-off between detection rate and number of false positives.

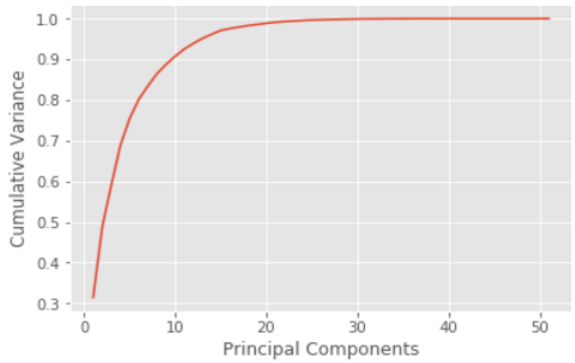

Figure 4: First 11 components explain 90% variance in the data.

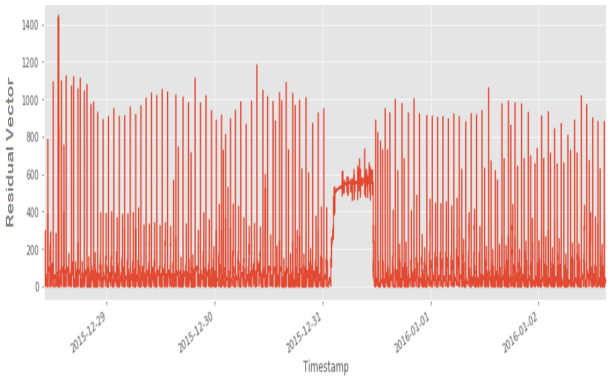

Figure 5: Residual Vector obtained by projecting data onto anomalous subspace.

The confusion matrix in Table 2 shows the results obtained for threshold of 1000 on residual vector.

Table 1: Confusion Matrix: Columns represent predicted labels and rows represent actual labels.

Table 2: Threshold = 1000

	Anomaly	Normal
Anomaly	114	395551
Normal	468	53749

Table 3: Threshold = 500

	Anomaly	Normal
Anomaly	30688	392914
Normal	3105	23175

A threshold of 900 detects 156 anomalies but produces over 1369 false positives. From table 3, we see that a lower threshold of 500 is able to detect 30688 anomalies and produces 3105 false positives.

3 ARMA

The sensors used for the ARMA model are *FIT101*, *LIT101* and *AIT201*. *MV101* was not used to fit ARMA model as it being an actuator , its values were either 2(open) or 1(close). This kind of values would not be helpful to ARMA since it depends on autocorrelation of instances.

The preprocesing for ARMA involved collecting the data points till the first instance of *Attack* and creating a training set out of it. The remaining data points were used as test set. The data points were not normalized as Autoregressive models are robust to raw values.

We used *Dickey-Fuller Test* to check if the signal was stationary or not. For all the above three sensors we found the *Test Statistic* to be greater than *Critical Values* which indicates the non-stationarity of the signal [2]. To make the signal stationary, we took the first difference of each signal by using *ARIMA* model with $d = 1$.

We used *AIC* statistic to determine the model that has best captured the training data. The model that has the least *AIC* value for a given set of parameters, p & q, is chosen eventually [2]. Tuning of parameters is a computationally heavy task for the size of our dataset. Hence, we resampled the training data in term of *Minutes*. With the current size of around 8900 samples, tuning became quicker. Test data was not resampled.

Instead of predicting or forecasting each sample in test set, we used the parameters that the model had learned while fitting on train set and fit a new model on the test set with these parameters. This way, the model did not treat anomalous data as normal and we saved lot of computation time. The threshold was heuristically set to $\boldsymbol{2} * \sigma$ where σ denotes standard deviation of the residual signal.

To combine the models into a single anomaly detection method, we simply checked the value of coefficients that were either close to each other in each model or the values that lied under the 95% confidence mark. This gave us good results as the residual values for *FIT101* and *AIT201* from the combined model were way lower than the values of *LIT101* which makes sense since no anomalies have been reported for *FIT101* and *AIT201*. Figure 6 gives the plots for the residual vector derived from the *LIT101* sensor.

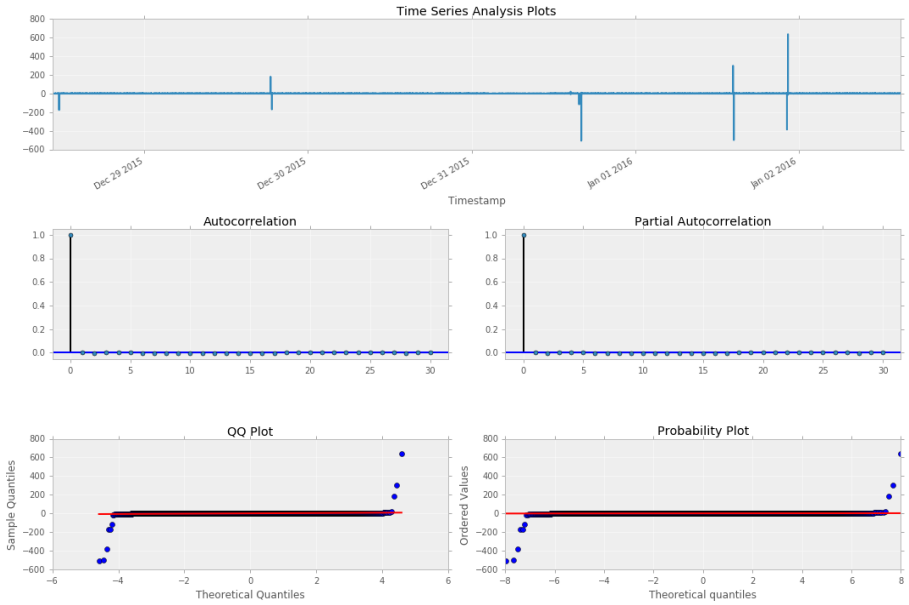

Figure 6: Residual, autocorrelation and correlation plot for *LIT101* with $p = 3$ & $q = 2$

The residual vector should be normally distributed, be random and constant, which can be seen from the probability plot. The residual plot shows some high spikes that could correspond to reported anomalies.

4 Comparison

For the PCA model we have used point-wise detection of true and false positives. As explained in Section 2.3, a point is counted as a TP if the detected residual value lies in the anomalous region. This way we could calculate point-wise precision and recall. If the overall aim is to have lower data accounted as FP then this metric is a good way to validate. However, this metric does not give the interval at which an anomaly has been detected or by which sensor.

For ARMA model, we count a true positive if it detects atleast one anomaly in an anomalous region. The reason being that an ARMA model may or may not capture the anomalous behavior for the whole time segment since some of the anomalies depend on the behavior of other correlated signals. A univariate autoregressive model still has limited information about the anomalies and hence, it is enough to detect the point where there is a sudden spike of values.

The other metrics that we considered to be important as well was *Segment-based metrics* and *Event-based metrics* [3]. *Segment-based metric* would say how many instances of anomalies were found in a given segment whereas, *Event-based metrics* would provide the number of times a sample as been detected as anomalous. Segment-based metric would be a good metric if an analyst would want to see how good his anomaly detection method is. This could help in deciding the sensitivity of the model. Event-based metric could be a good metric to count the false positives and true positives for a general system.

Figure 7 shows the performance of our ARMA model on the *LIT101* sensor. The sensitivity of our model could be deemed as low if we use Segment-based metric as less than 3% of instances have been detected as anomaly in each segment. However, going by the Event-based metric, our model detects 40 values as anomalous out of the total 70 which is quite good. On a general overview, our model is able to detect all the anomalies that have been reported for *LIT101*.

In conclusion, we believe that both, PCA and ARMA models have their own advantages. It is easier and quicker to detect anomalies with PCA when the dataset has high dimensions (multivariate). However, ARMA models can tell us exactly which sensor is giving anomalous behavior and when. Hence, a combination of PCA and ARMA could be employed to build a better anomaly detector where the PCA could be used to reduce the dimensionality and make the data suitable for ARMA [4]. Alternatively, Robust-PCA [5] could be used as it considers the transformed data as the sum of normal data and sparse data (which, could be tested for anomaly).

```
Start Time : 11:22:00
Anomaly instance : 1 out of 382 instances
End Time : 11:28:22
Start Time : 18:29:59
Anomaly instance : 3 out of 721 instances
End Time : 18:42:00
Start Time : 15:47:40
Anomaly instance : 30 out of 1170 instances
End Time : 16:07:10
Start Time : 14:21:12
Anomaly instance : 1 out of 443 instances
End Time : 14:28:35
Start Time : 22:16:01
Anomaly instance : 5 out of 539 instances
End Time : 22:25:00
 Total Anomaly instances found : 40 out of 70 values
 Total Anomalies found : 5 out of 5 values
```

Figure 7: Performance of ARMA for anomaly detection on the *LIT101* sensor.

References

[1] Anukool Lakhina, Mark Crovella, and Christophe Diot. "Diagnosing network-wide traffic anomalies". In: *ACM SIGCOMM Computer Communication Review*. Vol. 34. 4. ACM. 2004, pp. 219–230.

[2] *Time Series Analysis (TSA) in Python - Linear Models to GARCH*.

[3] Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. "Metrics for polyphonic sound event detection". In: *Applied Sciences* 6.6 (2016), p. 162.

[4] Augustin Soule, Kavé Salamatian, and Nina Taft. "Combining filtering and statistical methods for anomaly detection". In: *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. USENIX Association. 2005, pp. 31–31.

[5] Mei-Ling Shyu et al. *A novel anomaly detection scheme based on principal component classifier*. Tech. rep. DTIC Document, 2003.