Cyber Data Analytics*

Assignment 2

Tugce Arican
University of Twente
s1862863

Aishwarya Shastry†
Delft Institute of Technology
4743016

ACM Reference Format:
Tugce Arican and Aishwarya Shastry. 1997. Cyber Data Analytics: Assignment 2. In *Proceedings of ACM Woodstock conference (WOODSTOCK'97)*, Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 4 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

This Assignment is for the Course Cyber Data Analytics.

In this assignment, we apply various techniques for anomaly detection in SCADA systems. We use the following 3 models for our assignment: PCA, ARMA and Discretization Model. We use Training sets 3 and 4 for our task.Our code could be found in https://github.com/Aishwarya-96/CDA_Assignment-2

2 FAMILARIZATION TASK

2.1 What kinds of signals are there?

The dataset contains 43 signals. There are two types of Signals to be specific: Sensors and Acutators. Sensor Signals being:L_T1-L_T7, F_PU1-F_PU11,F_V2,P_J280-P_J422 and Acutators being S_PU1-S_PU11 and S_V2. .Few of them have been visualized below in fig1 and fig2

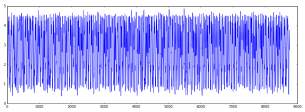

Figure 1: L_T1 Signal

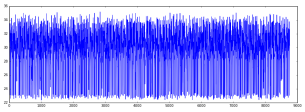

Figure 2: P_J422 Signal

*Produces the permission block, and copyright information
†The secretary disavows any knowledge of this author's actions.

WOODSTOCK'97, July 1997, El Paso, Texas USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

2.2 Are the signals correlated? Do they show cyclic behavior?

We performed some analysis on the training datasets and found the correlation among the signals as in fig3. We can see that there is a high correlation among same signals and other signals which are lighter in color in the Heatmap.

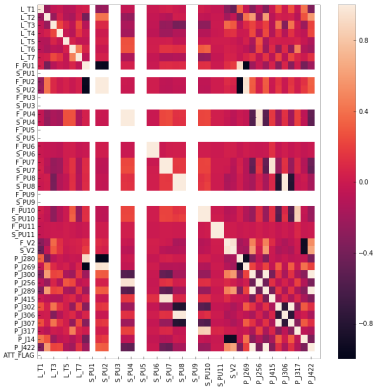

Figure 3: Correlation of Signals

We can see from the fig 4 that the signals here L_T2 and L_T6 show cyclic behavior.

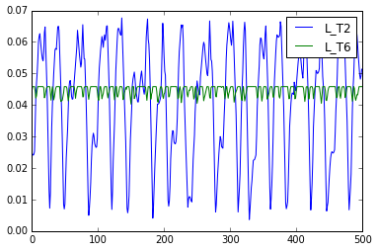

Figure 4: Behavior of Signals

2.3 Is predicting the next value in a series easy or hard? Use any method from class.

For this task, we decided to use the Persistent task from the tutorial[3]. We did it for first signal L_T1 and checked it for first 50 entries to get a clear picture. We get an MSE of 0.042 which is quite low and it should be not hard to predict the next value for this signal. But it depends on the data and the spikes.

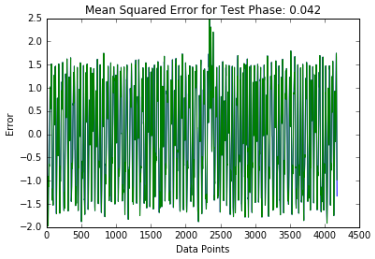

Figure 5: MSE for signal L_T1

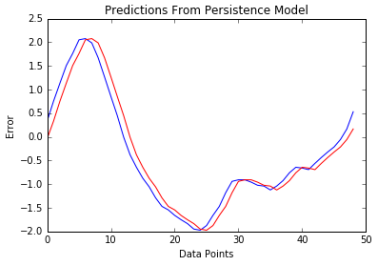

Figure 6: Prediction from Persistent Model

3 ARMA TASK

Auto Regressive Moving Average(ARMA) is a combination of Auto Regressive and Moving Average statistical models which are using history and error terms in order to forecast a data point. In this task, it is asked for investigating ARMA models on sensor data given by Batadal. Each tank sensor,L_T1L_T7 has been tried to be modeled by an ARMA model, individually.

3.1 Parameter Selection

ARMA is a parametric model. These parameters indicate how many historical data points and error information will be used to predict a new data point. ARMA parameters can be chosen in several ways. One of them is AIC value. The lower AIC value shows the better fit between the original signal and the modeled one. This information can be used for a grid search. Although a small parameter search has been implemented in the assignment, it was not used in the final decision of AR and MA parameters, since it mostly found higher orders. Moreover, models found did not give a significant improvement, and in some cases, the searched parameters generated worse models. Autocorrelation and partial autocorrelation plots have been chosen for parameter estimation, instead. Although it requires manual inspection, it still can provide good models. Parameters selections have been done based on the rules mentioned in[2]. Prediction threshold has been calculated based on standard deviation and mean squared error. Any prediction differing from mean squared error by a factor of standard deviation has been marked as an anomaly. This factor has been chosen after some experiments.

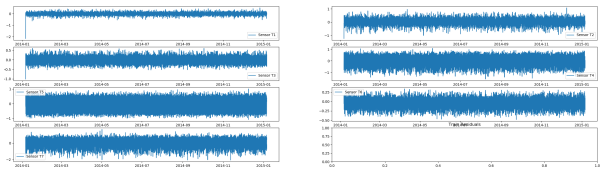

Figure 7: Train model residuals

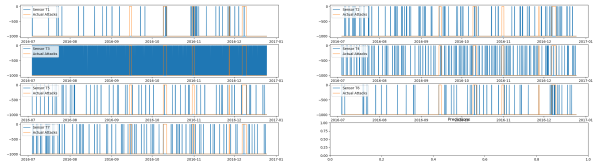

Figure 8: ARMA predictions

3.2 Residuals and Predictions

An individual ARMA model has been trained for each sensor. The following figure shows the residuals of training model7. Train residuals show that models were able to capture patterns in the train data. Some spikes in sensor T1 and T2 indicates that small amount of correlations have been left. These models can be improved, yet, that might not make much improvement. Autocorrelation plots of trained model residuals were removed from this paper due to page constraints. With test residuals, they can be found in the code. Detection results can be seen in the figure below**??**. None of the learned models was able to detect all attacks in the second dataset. However, some models and some sensors slightly better than others. Most of the Attack 4 could be detected in sensor T1. Some hours of Attack 3 and Attack 5 were also be detected in the same sensor. Sensor T6 could mostly detect Attack 6, while it missed almost all other attacks. Sensor T3 was the most difficult sensor, which has the highest false positive rates.

4 DISCRETIZATION TASK

Continuous data can be converted to discrete sequence. Besides reducing the size of continuous data, discretization also allows applying some other techniques for time series analysis. In this task, SAX discretization has been applied. SAX is based on PAA transformation. In PAA, first, time series is divided into smaller equal sized chunks. Data that falls into a chunk is represented by the chunks mean value. Later, based on some cut points, this data is labeled with a letter from an alphabet.

4.1 Parameter Selection

The number of chunks, required by PAA transform, and alphabet sizes has been chosen in ranges 500-800, and 5-7, respectively, via manual inspection of graphs9. Cut points required for SAX transformation has been taken from Qin Lin's SAX implementation. Detection threshold has been set after brute force search based on tp/fp rate.

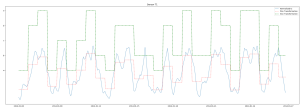

Figure 9: SAX transformation of sensor T1

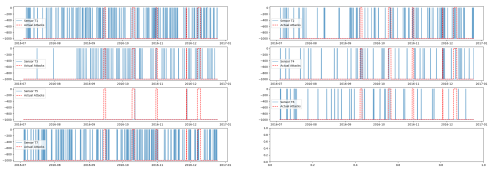

Figure 10: Predicted attacks

4.2 Anomaly Detection

Firstly, both train and test data have been discretized by using SAX transform. Later, 3-grams were extracted from train data. Test data has been read with a sliding window of size 3. This test sequence, later, has been compared with 3-grams generated from train data. The comparison is based on a distance metric defined by a look-up table. This look-up table includes the distances between letter pairs within the given alphabet. Test sequences that were decided to distant from a trained n-gram by a threshold have been marked as an anomaly. This threshold has been set via brute force for each sensor. Following images show anomalies detected by this implementation for each sensor10. Sensor T1 and T7 had several false positives. Sensor T5 was able to detect attack number 5. This sensor detected only 4 anomalies, yet it did not raise any false alarm. Sensor T3 was able to address all attacks but attack number 5. However, sensor T3 has a high false positive rate. On the other hand, sensor T6, missed all attacks but attack number 3.

5 PCA TASK

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.[1] The First principal component explains maximum variance, the second principal component second highest variance and is orthogonal to the first component and so on. As the top components represent large variance, we only need few components to represent our data. These principal components are used to model normal and anomalous subspaces which can be used to detect anomalies in the dataset. For this task we have assumed the ambiguous values 999 to be zero.

5.1 Preprocessing

For this task,we have normalized the data to contain zero mean and unit variance.

5.2 Choosing Principal Components by Modeling Normal and Anomalous Subspace

From fig 11, we can see that the first 12 components explain 95% of the variance which represent the normal subspace. The projection of signals is plotted on next 5 principal components and we observe that because of the spikes they can be used to model abnormalities in our data. So, we select 13 to 15 PC for our model.

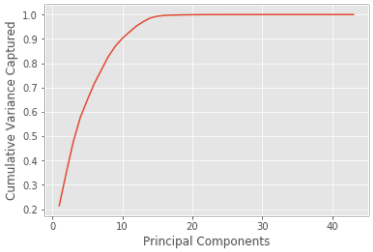

Figure 11: PCA components showing variance

5.3 Detecting Anomalies

We train our model with Training set 1 and test it on Training set 2. For this task, after modeling the normal and anomalous subspace, we project our training dataset 2 to these subspaces and observe the abnormalities12. These abnormalities indicate the presence of anomalies. It is done by computing Squared Prediction Error(SPE) for the residual vector and classifying data above a threshold as an anomaly. We calculated the threshold as mentioned in the paper[4] and found it to be 147. We obtained TP-57, FP-66,FN-3892,TN-162. The spikes in Fig 12 are similar to the attacks provided by BATADAL.

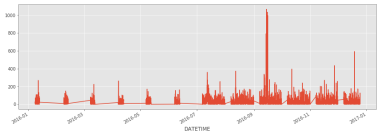

Figure 12: Abnormalities in Training Set 2

6 COMPARISON TASK

For this task, We chose precision and recall to compare our three anomaly detection methods. Precision tells us the probability of a true attack when there is an alert and Recall here refers to the ability to identify all anomaly. We want a high precision as we want we want low false alerts and a high recall as we donâĂŹt want to miss many attacks. For our ARMA and Discrete Model tasks, we sum up the metrics of confusion matrix for the 7 signals and then calculate Precision and Recall. We decided to do this because we can't get an accurate estimate of the metric if a single signal is chosen.

Table 1: Comparison of metrics for our models

Metric	PCA	ARMA	Discrete
TP	57	175	197
TN	162	1158	1336
FP	66	3010	2107
FN	3892	24682	25571
Precision	0.463	0.0549	0.0855
Recall	0.0144	0.007	0.008

An anomaly detection method needs a good tradeoff between Precision and Recall. From the above table 1, we can see that PCA performs better in terms of that. We would like to comment that all these models have their benefits. PCA is faster and it is easier to detect anomalies through it but ARMA tells us which sensor is behaving abnormally.Time series discretization helps to reduce data size, and also allows to apply sequential analysis techniques. Although it is easy to implement discretization and sequential methods, setting up all required parameters needs some field knowledge and exhaustive experiments.SAX was faster than ARMA and could detect anomalies which could not be detected by PCA.Thus, it is better to employ all these methods to build a hybrid model for anomaly detection.

REFERENCES

[1] Principal Component Analysis. (????). https://en.wikipedia.org/wiki/Principal_component_analysis

[2] 2015. Time Series Analysis using iPython. (2015). https://bicorner.com/2015/11/16/time-series-analysis-using-ipython/

[3] 2017. Autoregression Models for Time Series Forecasting With Python. (2017). https://machinelearningmastery.com/autoregression-models-time-series-forecasting-python/

[4] Anukool Lakhina, Mark Crovella, and Christophe Diot. 2004. Diagnosing Network-wide Traffic Anomalies. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '04)*. ACM, New York, NY, USA, 219–230. DOI: http://dx.doi.org/10.1145/1015467.1015492