Group33Lab2report

May 26, 2020

0.1 CS4035 - Cyber Data Analytics - Lab 2 (Anomaly Detection)

0.1.1 Group Number : 33

0.1.2 Student 1

Name : Shipra Sharma

ID : 5093406

0.1.3 Student 2

Name : Sudharshan Swaminathan

ID : 5148340

0.1.4 Readme (setup instructions)

We have attached the `requirements.txt` file. The required libraries, apart from the ones that are already not being used by sklearn, can be installed from there. In case of any problems, they can be installed manually too: `tslearn`, `h5py` and `statsmodels`.

0.1.5 1. Familiarization task

Types of Signals: We have two training datasets, each of which has multiple types of sensor signals (43) present. These signals depict multiple properties like level in tank (L_T4, L_T5), pressure (P_J280, P_J300) and flow levels of pumps (F_PU1) and valves(F_V2). There are also values which depicts whether a sensor is active or not. The plots of signals L_T4 (training data) and F_PU1 (evaluation data) shows that each signal is accompanied by two values i.e. its labels and timestamps.

Cyclic behaviour: The heat map depcits clearly that lighter the color, the more correlated attributes are. We can observe that L_T6 highly correlated which gives a corelation among water level in tanks. Another example is a highly correlated pair i.e. F_PU1 and P_J280 which shows that flow level and pressure are correlated. We can observe the cyclic behaviour between this pair through one of our output plots.

Prediction We consider AutoRegression model to do the signal prediction. Autoregression is a time series model that uses observations from previous time steps as input to a regression equation to predict the value at the next time step. We use it as it is a very simple model that can result

in accurate forecasts on a range of time series problems. For this particular task, we use F_PU1 signal where the prediction seems to be matching the original signal which shows that prediction easy to spot. The prediction is done using the evaluation training set.

0.1.6 2. LOF - Sudharshan Swaminathan

As can be seen from the plotted LOF scores, all the anomalies are visible in the negative y axis (scores less than -1). More negative the score, more anomolous the data point. LOF scores are being calculated in the following way:

1. Extract the test labels from the training set 2. Change the value of the test labels from (-999, 1) to (1, -1) so that it can be matched with the results from LOF.
2. Fit the training data set 1 using the LOF function.
3. Use the computed factors to calculate the possible outliers using the predict function. Here, the novelty methods have been used.
4. Compare the results on the training evaluation set (training set 2) with the test labels from step 1 to compute False Positives and True Positives.

On trying various k neighbours from a range of 2 to 50, it was realised that k with values 6 and 17 gave the best results (the trials have been commented in the code, and only these 2 values have been taken for clarity in the report). These 2 values were the chosen according to the maximum AUC under the ROC curve. Also, these 2 values gave a balance between false positives and true positives, therefore the most suitable.

The abnormalities that are seen in the training samples are not very far away from the cluster as the scores given by *negative_outlier_factor_* are not very far away from -1 to be a probable attack, but far enough to be an outlier and therefore an abnormality. So these might just be malfunctioning of signals, thereby considered as an outlier and can be removed from the training set.

The anomalies that are detected using LOF are based on outliers, more negative the LOF score (using *negative_outlier_factor_*), more further away is the datapoint, more probable it is to be an outlier and therefore, an anomaly.

0.1.7 3. PCA - Shipra Sharma

The task is performed as following: 1. Normalize the datasets as PCA takes normalised data 2. Find the importance of each component of the data set by finding the explaied variance ratio 3. Pick the number of important components and find projections 4. Find projections for the chosen number of important components on both the data sets 5. Calculate residuals of each point in the signal and combine the same

Principal component analysis, or PCA, is a statistical procedure that allows one to summarize the information content in large dataset (like ours with 43 signals) by means of a smaller set of important principal components that can be more easily visualized and analyzed. For the given task we were supposed to find the PCA residual for each point in the signal. We did so for both the training data. To find the best number of components, we varied the value of n from 6-15 depending on the explained variance ratio and found that the best result comes for the value as n= 15. In the above plots it can be seen that the evaluation training dataset has more spikes than the training dataset. These high spikes refer to abnormalities as the value is largely deviating from the average value of the signals, this can be seen clearly in both the training sets. Since, PCA works on variance hence, it can be used to find anomalies where in any given set, there is observable changes

in variance. To analyse the detection of attacks via PCA we set the threshold with the maximum and minimum residuals of the first training data and calculate anomalies on evaluation training data. The results of PCA would be explained in the comparison task.

0.1.8 4. ARMA - Shipra Sharma

Following steps are performed while implementing ARMA: 1. Analyse whether the given signal is staionary and plot autocorrelation and partial correlation for L_T1 signal 2. Using AIC to find best order of p and q 3. Find the training and evaluation residuals and plot them respectively 4. Find the anomalous region based on threshold and residual values 5. Spot potential_attacks and evaluate ARMA by finding the TP and FP for the chosen signal data

ARMA (Autoregressive moving average model) uses the p (previous values) and q (residual values) of a given signal/series (here we use L_T1) to understanding the past and predicting the future values. While performing procedures on a signal, the first step might always be to check whether its stationary or not. In our case it turned out to be a stationary one. Next was to find the range of p and q from the autocorrelation plots. From the ACF plot, we can deduce that a positive correlation exists between 0-10 hence the MA order could be between 1 and 2. The PACF plot gives the exact relation for the AR constants. We then use AIC to compute the best parameters for the ARMA model (Here we tried with different range of p and q and used the best one). These parameters are then used to get the residuals for both the training set and evaluation set. We aim to find the anomalous region by tuning the threshold accordingly. We tried different multipliers for evaluating the maximum and standard threshold to limit the anomalous region for the training data. We can clearly spot the difference between the residuals of the training and evaluation data by looking at the two plots above. The sudden spikes in the first residual plot symbolises the abnormalities and such instance must be treated in order to suspect attacks. The results are mentioned in the comparison task.

In our analysis, ARMA works better for detecting anomalies that arise in a sudden timestamp and not frequent/repeating manner, because it uses previous values and residuals to predict the next values. This is because, if there are too much anomalies in the previous data and a high spike might be missed.

0.1.9 5. N-gram - Sudharshan Swaminathan

Visualise discretization - The dicretized data has been plotted below. As can be seen the signal is shown by blue, which has then been coverted to particular discrete data using SAX transform shown in green. Here, only 2 discrete values have been used which depends on the number of PAA segments and the alphabet mapping size used.

4-grams was chosen to convert the data into N-grams. Once all the data points were converted to n-grams, where 4 data points are taken as one window, we then split the entire data set into windows on the basis of time. For example, with window size 1000, the first 1000 n-grams for that signal is taken. Then the second window is taken in the range 900-1900, then the 3rd window segement is 1800-2800, and so on. In this way, it will be possible to localize an anomaly in a span of short window. That is if we compare the 2nd window of the training data set with the 2nd window of the evaluation data set, then if there are ngrams that do not exist in the evaluation data set for this window, it can categorized as an anomaly for this window, and we can then say that 2nd division of time consists of an anomaly. This is further inspected by extracting the data points. But because the data is discretized and converted to ngrams, the exact data points where

the anomaly can be found is determined with less probability. This might be the reason why, as seen in the above plots and the precision data, we get less precision using this method. The tables above show the sliding windows over the time period for the selected ngrams for the signal L_T1.

As stated above, we consider it an anomaly if an ngrams is present in the testing set but not in the training set. Therefore, the anomalies detected here are based on signatures. If the signature of the testing set varies from that of the training set, an alarm is raised.

We can also see that the precision is the highest for L_T2, P_J300, P_J289, and P_J422. We also see a significant number of True positives for F_PU11 and S_PU11. These signals can be used for detecting anomalies in a relatively better manner. Also signals F_PU2, F_PU10, and F_V2 give all the true positives, 219, but with a high numbrr of false positives which can concur a lot of cost if the process of validation is expensive.

0.1.10 6.Comparision

From the given instructions we try to evaluate the performance of the above implemented four tasks (LOF, PCA, ARMA and N-gram model) on the basis of true positives and false positives. The comparison of the given four methods is not easy because all of these methods have different implementation to find anomalies. Below are the precisions for all the four methods (precision = tp / (tp + fp)), based on which we make our comparisons.

LOF : 0.287 (with k = 17)

PCA : 21/25 = 0.84

ARMA: 4/8 = 0.50

N-Gram: 0.18750 (with multiple signals as stated in its respective answer)

LOF uses the concept of outliers to find anomalies. If a datapoint lies far away from its cluster, then it is detected as an anomaly. As there are always abnormalities in the signals, especially in an ICS system, detecting anomalies through outliers is bound to give significant number of false positives. However, we also note here that in spite of giving false positives, there are also significant number of true positives given by this method.

In PCA point wise detection of true positives and false positives, made the task a bit easier. TP is nothing but a point detected in the anomolous region based of residuals and thresholds (potential_attacks). The choice of 15 most important components and computed thresholds of the anomalies led to the detection of 21 true positives and 4 false positives from the computed list of 25 potential attacks.

ARMA is not a point-based implementaion and works on signals. Thus, we tried evaluating multiple signals like L_T6, L_T1 and F_PU1 and found out that L_T1 analysis and results seemed to be the best hence, we chose that. However, our approach creates a biasness in the system because if an attack is made on one of the signal, the others might be unaware of the same, due to the single signal processing that we did in our model.

N-Grams depends on signature based detection using a series of ngrams. However, this signature is computed over different periods in time, and therefore, proper alignment of the time-series data needs to be done. This alignment proves as a hinderence for attaining efficiency in this case, and results in low precision. In some case, all the anomalies are detected but giving a lot of false positives at the same time.

To conclude based on the precision scores (which is an important metric in terms of evaluating a model), we could clearly spot that PCA outperforms every other model and it might be because:

- it enables the user to choose the most significant signals thus reducing the features of the given large data
- because it considers point wise implementation which is easier to analyse