Detecting Anomalies with Python and Influx DB using Machine Learning Algorithms

Mohammed Talat Abdel maqsoud, Sunitha Radhakrishnan, Reeshika Sundram, Elaheh Najafiani

Masters in Automation and IT

Technical University of Cologne, Germany

mtalatcise@gmail.com, sunitha0594@gmail.com, rssinha19@gmail.com, elahe.najafi-ani@gmail.com

Abstract: Anomaly detection is a process in data mining that identifies the data points or events that deviate from standard behavior. In reality we use technical systems for monitoring this behavior. Data for such monitoring systems will be changing over time to time and then stored in databases called Time Series Databases. This paper concerns about, InfluxDB to set up the multivariate time series database including the automatic process Kapacitor-Grafana interface to upload the incoming data. Anomaly detection software is developed using Python and Kapacitor which is a data processing engine coupled to Influx DB. Then the overall system is tuned and compared with other machine learning algorithms for accuracy with respect to detection performance. In this case study, Anomaly detector toolkit (ADTK) library in python is used from which PCA and Generalized ESD test AD algorithms along with OR aggregator is used for this application of multivariate anomaly detection. Also, real time implementation is done using Kapacitor-Influx DB-Grafana interface by inputting the algorithm and detecting anomalies in online fashion and the metrics were visualized. It was found that only the combination of the two algorithms (ESD+PCA+OR) gave promising results with a better F1 score as expected than the individual algorithms on multiple variables.

***Index Terms*—Machine Learning, Multivariate Time Series, Principle Component Analysis (PCA), Generalized ESD, Kapacitor, Anomaly Detection Software.**

1 Introduction

In data mining, anomaly detection is a technique of identifying the events, observation of rare items that deviates from the actual behavior of majority of the data set. In real world, with the use of anomaly detection system we can detect any kind of fraud, error signal generated by the instrument used in various kind of Industries. Anomaly detection can be noise, outliers, deviation and exceptions in the data set. Our goal of this case study is to develop a change(event) detection system to accurately predict any kind of

huge changes in time series or in other words to detect any kind of anomaly in time series data of drinking water quality.

There are three broad categories for anomalies namely point anomalies, contextual anomalies and collective anomalies. (i) Point anomalies: A single instance of data is anomalous if it is too far off from the majority of the data. Example: Detecting fraud in credit card usage on the basis of amount spent. (ii) Contextual anomalies: If a data instance is anomalous in a specific context/Condition but not otherwise then it is referred as contextual anomalies. Example: Spending money more on food during holidays as compared to normal days. (iii) Collective anomalies: If a collection of related data instances is anomalous with respect to entire dataset, it is termed as collective anomalies. Example: A reading of Electrocardiograph that break the rhythm of its reading can be termed as Collective anomalies.

For a healthy living, Purity of water is the most important factor. The provision of clean and safe drinking water is essential for water supply companies all over the world. The purity of water is the biggest challenge we are encountering with, in the 21st century. It directly impacts the whole ecosystem. In this Case Study, we perform anomaly detection for multivariate model using multiple variables that affect the quality of the drinking water. One of the large datasets known as the Gecco Challenge 2017 Thueringer Fernwasserversorgung dataset is used in which machine learning algorithms are built and also visualized in a real time scenario.

2 Data Analysis, Preprocessing and Data Visualization

The following dataset initially investigate the possibility to rely on *machine learning* (ML) models to detect anomaly. To take the first steps into this direction, the data was analyzed and preprocessed using python. Machine learning Algorithm was then implemented on the data. To implement real time simulation, the data was uploaded from a csv file to InfluxDB in real time, and the machine learning algorithm was implemented with Kapacitor to detect anomalies. The InfluxDB-Kapacitor interface was then integrated with Grafana, a power visualization tool to visualize the data along with anomalies in real time.

2.1 Data Analysis

The data contains time series denoting water quality data and operative data on a minutely basis. The Gecco challenge dataset consists of 122334 records and 11 columns. Before dealing with the algorithms in the data, we need to understand the type and characteristics of the data. Given is the amount of chlorine dioxide in the water, its pH value, the redox potential, its electric conductivity and the turbidity of the water. These values are the water quality indicators, any changes here are considered as events. The flow rate and the temperature of the water is considered as operational data, changes in these values may indicate changes in the related quality values but are not considered

as events. Variable importance for independent variables in our dataset w.r.t the EVENT is found out using Feature Selection from Random Forest classifier library and it is found out be that Redox and Cl_2 is the most important variable w.r.t the event and plays a major role in determining water quality. Also Covariance matrix is plotted to find out the highest co-related variable and it was found out to be Redox-Ph has the high covariance of 0.62.

Column	Description
Time	Time of measurement in following format: yyyy-mm-dd HH:MM:SS
Tp	The temperature of the water, in °C
Cl	Amount of chlorine dioxide in the water, given in mg/L (MS1)
pH	PH value of the water
Redox	Redox potential, given in mV
Leit	Electrical Conductivity of the water, given in µS/cm
Trueb	Turbidity of the water, given in NTU
Cl_2	Amount of chlorine dioxide in the water, given in mg/L (MS2)
Fm	Flow rate at water line 1, given in m3/h
Fm_2	Flow rate at water line 2, given in m3/h
EVENT	Marker if this entry should be considered as a remarkable change with Boolean data type

Table1: Description of the Thuringian water supply dataset

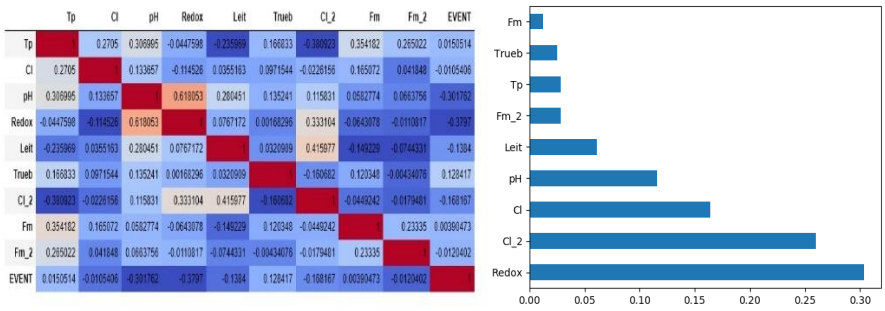


Fig 1: The corelation matrix and the variable importance plot

2.2 Data Preprocessing

Data Preprocessing is an important step in data mining process. Data Pre-processing include Cleaning,normalization, feature selection,etc.Though there are various methods to fill NA based on the dataset and to get more accurate result.Here in our dataset, we

have checked for the NA values and filled it with the mean values. Data Normalization was done by scaling all the independent variables in the range of 0 to 1.

2.3 Data Visualization

For data Visualization, we used matplotlib and adtk plot in python to analyze the trend of the various variable in the data in offline mode. For real time analysis we used Grafana tool. Grafana is an open-source solution for data visualization and analytics. It can be integrated with various databases including InfluxDB. The following figure shows the main layout of the implemented Grafana Dashboard:

Fig 2: Data Visualization in Grafana and its dashboard layout

The Dashboard shows real-time plots of the three main variables (Redox, pH and Cl_2). For each variable, the following points are shown: (i) Real-time points uploaded to InfluxDB (green), (ii) Real-time anomalies (red), (iii) last window provided by Kapacitor HTTP endpoint (blue). In addition, performance measures are also displayed, including confusion matrix, accuracy, balanced accuracy and F1_score.

3 Anomaly Detection

3.1 Algorithms used for detecting Anomalies:

In this case study, anomaly detector toolkit (ADTK Library) was used for detection of anomalies. ADTK provides 3 components such as detector, transformer and aggregator to be combined into a model. But for our model, only detector and aggregator was used. A detector is a component that scans time-series and returns anomalous time points. They are all included in a module 'adtk.detector'. An aggregator is a component that combines different detections results (anomaly list). The significance of adtk library is it gives the result of the determination anomalous point as in the Boolean form as 'True' and 'False'. This makes the anomaly detection process easier to compare with the' EVENT' column which is a Boolean data type form.

In ADTK, 3 algorithms were used to detect anomalies in different variables. They are Principal Component Analysis Anomaly Detector (PcaAD), Generalized ESD Test AD. And then combining the anomaly results with OR aggregator.

3.1.1 Generalized ESD (Extreme Studentized Deviate) Test AD:

Generalized ESD test AD is a statistical test for outliers. It is used on Univariate data which follows an approximately normal distribution, and can be used to detect one or more outliers. This is a robust technique which will often down-weight the effect of outlying points without deleting them Extreme studentised Deviation is also called Grubbs test: This test detects one outlier at a time. This outlier is expunged from the dataset and the test is iterated until no outliers are detected

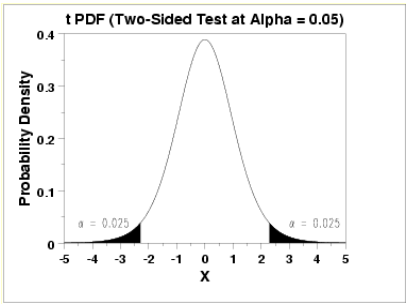
H_0: There are no outliers in the data set.

H_a: There is exactly one outlier in the data set,

For the two-sided test, the hypothesis of no outliers is rejected (this means outlier is available) at significance level α if

$$G > \frac{N-1}{\sqrt{N}} \sqrt{\frac{t^2_{\alpha/(2N),N-2}}{N-2+t^2_{\alpha/(2N),N-2}}}$$

with $t_{\alpha/(2N),N-2}$ denoting the upper critical value of the t-distribution with N-2 degree of freedom



For two-sided test, we calculate $\alpha/2$, the most commonly used value of $\alpha = 0.05$ .That gives the critical value 0.005/2=0.025, if a value is greater than 0.025 then the null Hypothesis is rejected and the point is denoted as an Outlier

Confidence Level:

The confidence level, 1 – α, has the following interpretation. If thousands of samples of n items are drawn from a population using simple random sampling and a confidence interval is calculated for each sample, the proportion of those intervals that will include the no outlier population standard deviation is 1 – α.

3.1.2 Principal Component Analysis Anomaly Detector (PcaAD):

PCA is used for multivariate time series and tracks reconstruction error of those error of those vectors.

Detector that detects outlier point with principal component analysis. This detector performs principal component analysis (PCA) to the multivariate time series (every time point is treated as a point in high- dimensional space), measures reconstruction error at every time point, and identifies a time point as anomalous when the reconstruction error is beyond a threshold based on historical interquartile range.

The detection of anomalous points in the reconstructed vector from the principle component is determined by two parameters 'k' and 'c'

k= number of principle component to use.

c= factor used to determine the bound of normal range based on historical interquartile range. We have used a default value of 5.0 in our case that indicates the value lying between 25% to 75% interquartile range.

3.1.3 OR Aggregator:

OR Aggregator identifies a time point as anomalous as long as it is included in one of the input anomaly list. It is applied to merge anomalies in the form of binary series.

3.2 Implementation of Anomaly Detection Algorithms:

3.2.1 Train-Test Split

The data set has a minutely data records for three months Feb-May. For most of the real-time case detection of anomaly on daily basis is most feasible. An algorithm that detect anomalies on daily basis can be of more interest for the real-world plant. Hence the train test approach is carried out on the daily Basis in this case study. The past four days data is trained with the algorithm and tested on the fifth day.

3.2.2 Implementation of PcaAD on Redox-Ph

Redox is the most important variable and according to the covariance matrix the relation between Redox-pH is 0.6. Hence Redox-pH was selected for PCA anomaly detection. The parameter selection is as follows:

K=2, as we have two principle components as Redox and pH

C=5.0, the default value is selected in our case that indicates the value lying between 25% to 75% of the historical interquartile range. This determine the threshold to consider the point as an anomalous point

3.2.3 Implementation of Generalized ESD for Redox and Cl_2:

Here only Generalized ESD test was performed on Redox and Cl_2 and then the results of the anomalous points were merged into one data frame

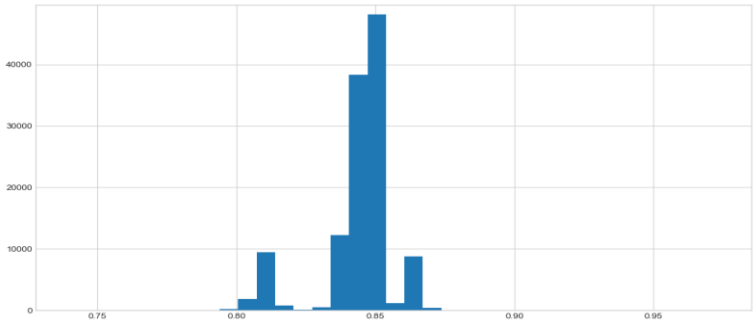


Fig 3: Histogram of Scaled Redox

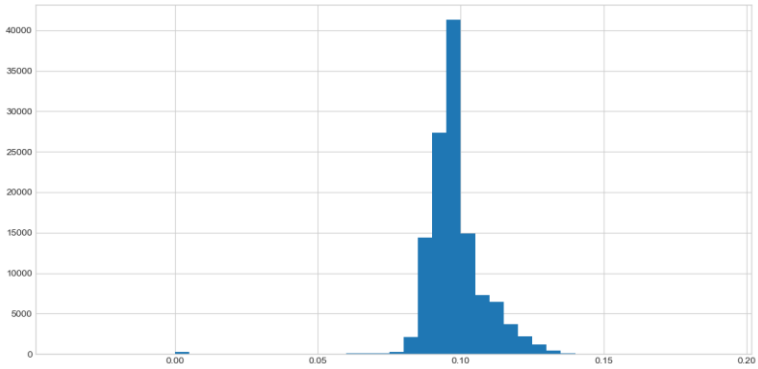


Fig 4: Histogram of Scaled Cl_2

It can be seen from the histogram plot of the scaled Redox and scaled Cl_2 it follows almost normal distribution pattern and is two tailed, most of the non-outlier value is concentrated within a narrow confidence interval of 1 standard deviation.

While applying the ESD algorithm we selected the value of alpha (Significance level) of 0.3 that leads to a confidence level of 70 percent. We tried the value of 0.05 that is the most common confidence level that is generally used for the data having Normal distribution pattern but since our data is a real-world data and after trying different confidence level the best suited result was achieved at 70 percent confidence level.ie; α=0.3

3.2.4 Implementation of OR Aggregator:

The main goal of this case study is the EVENT detection. Our Data is a multivariate data. The final event for indicating the water quality is the combination of anomalies generated by each variable. It is well explained before that redox, cl_2 and Ph are the significant variables for the final event for the water quality indication and hence the combination (ORing) of their results are important for detecting the event.

ESD algorithm is performed on Redox and Cl_2 to determine the anomalous points PcaAD algorithm was performed on reconstruction vector of Redox-Ph to determine the anomalous points. The OR aggregator aggregates the results of both and gives the final Boolean result of True event, if even one of the variable is True (anomalous).

3.3 Fetching Real Time Data using Kapacitor and InfluxDB

3.3.1 InfluxDB:

InfluxDB is one of the well-known time series databases that provides a simple interface to upload and analyze real-time data. In InfluxDB, data is stored in "databases" which consist of different measurements, each measurement contains a set of variables that represents a certain aspect of the dataset. In this implementation, we introduced "Water-Data Database" which contains the following measurements: (i) Water-Data-Points: which stores the real-time data uploaded from the csv file, (ii) Water-Data-Outliers: which contains the detected outliers, (iii) Water-Data-Performance, which includes the performance measures of the anomaly detection algorithms.

3.3.2 Kapacitor:

Kapacitor is an open source data processing framework that can be integrated with InfluxDB. It is designed to process streaming data in real-time, and it provides a simple plugin architecture, or interface, that allows it to integrate with any anomaly detection engine. To use Kapacitor for data processing, the required processing task should be implemented in a TICKscript code. This code contains the configuration of the services that will be running during the processing of real-time data.

One of the services that Kapacitor provides is the "HTTPOutNode", this service exports the recent data from a specific measurement to a HTTP endpoint. The parameters required for this service are window size (5 days) and update rate (1 day). Using this configuration, we can fetch the last 5 days data to the HTTP endpoint, which is automatically updated on daily basis.

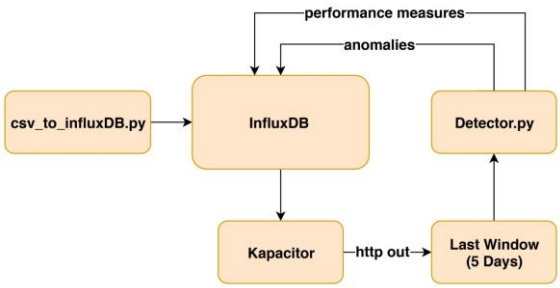


Fig 5: Real-Time Implementation with InfluxDB and Kapacitor –Overall Schema

3.3.3 Real-Time Anomaly Detection Code:

The detector monitors Kapacitor HTTP endpoint to process new data points as soon as they become available. The algorithm of this code is shown in the following figure:

Detector.py

```
if New Window Available do
  fetch data from http endpoint
  convert json to data frame
  split into training set (4 days) and test set (1 day)
  apply anomaly detection algorithm
  send points which are identified as anomalies to InfluxDB anomalies set
  calculate performance measures (confusion matrix, accuracy, balanced accuracy,
  f1 score)
  send performance measures to InfluxDB performance measurements set
else
  wait for new data
```

Fig 6: Anomaly detection Algorithm in online mode with the help of kapacitor

4 Results

ALGORITHM	True Positive	True Negative	False Positive	False Negative	Precision True Events	Recall True Events	Accuracy	F1_score of True Events
PCA-ESD	1108	114442	392	632	0.74	0.64	0.99	**0.68**
ESD	949	114442	392	791	0.71	0.55	0.99	0.62
PCA	265	114834	0	1475	0.1	0.15	0.99	0.26

Table:2 Results from confusion matrix for different cases along with Metrics

From the table above, it is seen that with the help of two algorithms on three different variable and combining (ORing) the results together there is a probability of achieving a high-F1 score. The F1_score achieved is 68% and accurately detected 1108 anomalies Out of 1740 anomalies. However, there is also 392 False positives that means they are actually not anomalies but was detected as anomalies by the algorithms

Fig 7: Comparison of results of final Anomalies with the EVENT column

The graph above were plotted using ADTK plots in python with help of matplotlib library for the whole dataset.

Fig 8: Visualizing the results of Anomalies in real time with Grafana.

From the above figure, it is shown that the algorithm was built in real time scenario using Kapacitor and anomalies were detected. The metrics were found out automati-cally and proved to get more or less the same f1 score of 69% similar to offline mode.

5 Conclusion and Future work:

From this case study, it was found that, for a multivariate data set it is important to select the variables for detection of the target variable(EVENT in our case).Variable importance(Redox-pH-Cl_2 in our case) plays a significant role in any dataset.ESD algorithm is used on a single variable following an approximate normal distribution pattern. Since Redox and CL_2 both follows an approximate normal distribution pattern the value of significance level selected for both of them is 0.3. This different Significance level value can be selected, since the normal distribution pattern for these two variables are different. However, PCA(Multivariate- analysis) gave us 0 false positives but a very less f1-score.Though different algorithms from supervised learning to unsupervised learning were built initially and for us the only the combination of these two (ESD+PCA) Anomaly detection algorithm on different variable gave us promising results with almost a good f1-score.

As it interests to us, the future work would be as stated: There is a co-relation between Leit and Cl_2, hence for PCA analysis the relation between these two variable in comaprison with other variables can also be taken in consideration for modelling.The default value of 5.0 was selected for the historical interquartile range in PCA Analysis. The different values can be selected as threshold to find out the best suited results which is also called as hyper parameter tuning. In addition, since false alarms can cause interruptions and additional costs, the detection algorithm should be improved to reduce the number of false positive points.

6 References

[1] Rameswara Anand. P, Tulasi Krishna Kumar. K: PCA Based Anomaly Detection. International Journal of Research in Advent Technology, Vol.2, No.2, February 2014 E-ISSN: 2321-9637

[2] Introduction on Anomalies. https://towardsdatascience.com/a-note-about-finding-anomalies-f9ce-dee38f0b

[3] Anomaly Detector Toolkit library in Python. https://arundo-adtk.readthedocs-hosted.com/en/stable/notebooks/demo.html

[4] Generalized ESD Test. https://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/esd.html

[5] Kapacitor and Influx DB Interface. https://www.influxdata.com/time-series-platform/kapacitor/

[6] Available dataset. https://www.spotseven.de/gecco/gecco-challenge/gecco-challenge-2017/ -

[7] Selection of alpha (GESD). https://www.itl.nist.gov/div898/handbook/eda/section3/eda3672.html