FH University of Applied Sciences TECHNIKUM WIEN

BACHELOR PAPER

Term paper submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Computer Science

Anomaly detection in data for data cleansing

By: David Zelenay

Student Number: 000000000000

Supervisor: Degree First Name Surname

Vienna, March 6, 2022

Declaration

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz /Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.“

Vienna, March 6, 2022 Signature

Kurzfassung

Change this

TODO Add Kurzfassung

Abstract

Change this

This paper analyzes methods for anomaly detection to cleanse data. An overview of some key features of data quality is provided in the beginning. The difference between data cleaning and data cleansing is elaborated. Additionally a few methods for anomaly detection (mainly outlier detection) are outlined. The hypothesis of this paper is: Which characteristics define data quality, with regard to IoT (Internet of Things) Sensors? Which methods are there to detect and clean or cleanse faulty data?

Contents

1 Planned Structure

This chapter describes a rough estimate of the planned structure of my bachelor thesis.

1.1 Overview

This section will provide an overview and introduction to the topic anomaly detection and data cleaning/cleansing.

1.1.1 Introduction

Partly already written in the paper. See chapter 2.

1.1.2 Data cleaning & data cleansing

chapter 3 (including data cleaning vs data cleansing)

1.2 Outlier detection

1.2.1 General Overview

chapter 4

1.2.2 Outlier detection methods

section 4.1

1.2.3 Threshold based outlier detection

section 4.2

1.2.4 Other variants of outlier detection methods

Provide a deeper insight in other outlier detection approaches. E.g. Clustering / predictive or distance based.

1.3 Outlier detection based on "real world data" (pegelalarm.at)

Create a connection between the theoretical descriptions of outlier detections to a real world use case. Data from https://pegelalarm.at/

1.4 What's the goal?

Describe the goal to archive:
“

1. We are looking for an algorithm that detects outliers using only historical values. This would allow us to assign a probability to the last measured water level of a station, which would indicate how likely it is to be an outlier. We would then not store outliers in our system at all or classify them as outliers from the beginning.
2. For us also an algorithm would be helpful, which assigns an outlier probability to each arbitrary measured value X of a time series. This algorithm would not only have access to the measured values before it, but also to those after it. This would allow us to detect outliers for all the time series data that we already have in the system and, for example, delete them.

Point 2 is probably easier to implement than point 1, so an algorithm 1 would be more helpful for us. Also important would be that the algorithm adjusts its (hyper)parameters accordingly based on the historical data. This means that a level at which there are often strong fluctuations, an outlier must already be quite outlier so that it is considered as an outlier. ”

1.4.1 How to retrieve the data (description of the API)

Short overview on how to use the API to retrieve data? Python project to retrieve data: https://github.com/SOBOS-GmbH/pegelalarm_public_pas_doc

1.4.2 Overview of the data

Provide an overview oth the data.

1.4.3 Explorative data analysis

Similar to overview of the data

1.4.4 Manually detect outliers for a subset of data

Show cases of outliers in the data and manually classify them. (Also define a way/data structure to classify outliers for time series data)

1.4.5 Define outlier detection performance metrics given on a subset of data

Define a way to compare different outlier detection models / define performance metrics. E.g. number correct outliers, average confidence for the correct outliers, number of missed outliers,....

1.4.6 Implement different outlier detection approaches

Develop different outlier detection methods in Python and calculate performance metrics for each

1.4.7 Compare different outlier detection approaches

Compare detection methods from the previous section.

1.5 Conclusion

Summary and conclusion

1.5.1 Advantages and disadvantages of used outlier detection methods

2 Introduction

With the growing popularity of Internet of Things (Internet of Things (IoT)) and digitizing business processes there is a growing amount of data available for analysis. In order to utilize the data from the IoT sensors it needs to be preprocessed. One step of preprocessing is data cleaning (also referred as data cleansing). The main goal of data cleansing is to increase the data quality and furthermore to detect and remove anomalies in the data. The quality requirements for the data can differ depending on the use case. Anomalies in sensor data are datapoints which do not picture the reality. For example an anomaly of a temperature sensor would be if the sensor reads 0 °C and the real temperature is 23 °C.

2.1 Research method

This paper is a literature research. To get an overview of the topic, papers related to: data cleaning, anomaly detection for IoT data / time series data and outlier detection methods were researched. After some base knowledge was established the major topics of the paper were defined. Subsequently more research was done on the major topics (Features of data quality, data cleaning & cleansing, outlier detection). To organize the references found while researching Zotero was used, with the Add-on Better BibTeX.

2.2 Features of data quality

This section will provide a few example key features of data quality.

Completeness

Data completeness describes the wholeness of data. If there are certain aspects of data missing the data is not complete. For example if each datapoint of a sensor includes the date, time and production speed, the data is not complete, if one of those features is missing or not entire, this datapoint is not complete. [1, 2]

Accuracy

The accuracy of data describes the exactness. Example for possible data which decrease the accuracy are outliers or time shifts. Usually the accuracy of data is harder to measure than

the completeness, consistency, structure or documentation. Due to the heterogeneity of sensor data (regarding numerical values like production speed or temperature, not categorical values like on/off) for each datapoint it is difficult to detect which values are genuine and which are sensor errors and therefore outliers. [1]

Consistency

One example for consistency would be, if the data interval is equal. For example there should be a datapoint every ten seconds. As soon as two datapoints are more than ten seconds apart from each other the data is not consistent anymore. [1]

Structure & Documentation

If the structure of the data is not homogeneous, it is very difficult to analyze in an automated way. As a result the data either needs to be structured from the beginning or a process needs to be fabricated to structure the data automatically. Furthermore documentation is required in order to structure and preprocess data. Documentation of data might include data format (Comma Separated Values (CSV), parquet [3], Java Script Object Notation (JSON)), date format (e.g. ISO 8601 with UTC offset), valid value spans (e.g. temperature is only valid if it is between 100 and 400 °C) [1]

2.3 Improving data quality

This section will describe methods to improve data quality, based on the features elaborated in section 2.2.

Completeness

The most common methods to increase data completeness are statistical and deep learning based approaches. The goal of these methods are to fill in the missing values of a dataset. An example for a statistical method is DynaMMo [4]. For ANNs (artificial neural networks) Long Short-Term Memory (LSTM) (Long short-term memory) or Gated Recurrent Unit (GRU) (Gated recurrent unit) can be used to predict missing data. [2]

Accuracy

One approach to increase the accuracy of data is to define constraints for each value. E.g. When a machine cannot produce more than ten pieces per second, because it is physically not possible, the value could be limited to less or equal than ten. However limiting the values to a specific range might hide the fact that the machine has an error and is producing faulty products

at a rate of 15 pieces per second. This is one of the reasons why more sophisticated outlier detection methods are used. [2]

Consistency

To facilitate consistent data, statistical smoothing or forecasting methods can be used. Examples methods are: AutoRegressive Integrated Moving Average (ARIMA) (Autoregressive integrated moving average) or Gaussian Process (GP) (Gaussian Process). ANNs can also be used to unify the time series interval between datapoints. [2]

Structure & Documentation

The process of structuring heterogeneous and messy data is called data wrangling. In order to unify the structure of the data at least some documentation is required. Therefore the documentation of the data is fundamental in order to analyse or further process it.

3 Data cleaning & cleansing approaches

There are two main methods when it comes to data cleaning or cleansing. Ignoring faulty data or replacing it with a representative value. This paper will use the term data cleaning to describe the process of ignoring or deleting incorrect data and the term data cleansing to portray the process of replacing invalid data with representative values. Faulty, incorrect, invalid or wrong data is data which is inaccurate, incomplete or inconsistent.
Example sensor data: (Valid values for `production_speed` range from 0.00 to 2.00 meter(s) per minute)

`ID`	`timestamp`	`production_speed` (meter/minute)	`machine_running`
0	2021-12-01T12:00:00.000	1.56	True
1	2021-12-01T12:01:00.000	1.58	True
2	2021-12-01T12:02:00.000	3.50	True
3	2021-12-01T12:03:00.000	1.50	False
4	2021-12-01T12:04:00.000	1.50	True
5	2021-12-01T12:05:00.000	1.49	True

Table 1: Example of IoT sensor data

3.1 Data cleaning

As already mentioned the approach for data cleaning is to ignore or delete faulty data. Depending on the use case either the entire datapoint needs to be ignored or just one value. The process of data cleaning will be shown with the example data pictured in Table 1. The first incorrect datapoint has the `ID` 2. This row is incorrect, because the `production_speed` exceeds the maximum value of 2.00. Depending on the use case (e.g. summary of how long the machine has been running) it can make sense to just ignore the row `production_speed` and keep the value for `machine_running`. The second appearance of a faulty datapoint has the `ID` 2. This datapoint is incorrect since `machine_running` is False but the value of `production_speed` is not 0.00. In this case it does not make sense to keep either of those values for further analysis, because it is impossible to determine which of the two columns are incorrect. A possible result after the data cleaning is shown in Table 2

ID	timestamp	production_speed (meter/minute)	machine_running
0	2021-12-01T12:00:00.000	1.56	True
1	2021-12-01T12:01:00.000	1.58	True
2	2021-12-01T12:02:00.000		True
3	2021-12-01T12:03:00.000		
4	2021-12-01T12:04:00.000	1.50	True
5	2021-12-01T12:05:00.000	1.49	True

Table 2: Example of IoT sensor data after cleaning

3.2 Data cleansing

Data cleansing pursues a different approach. Incorrect data is not ignored, but substituted by a representative value. For example for the datapoint with the ID 2 there are several strategies that could be followed. For example the outlier value 3.50 could be replaced with the upper limit of the valid range, in this example 2.00, the value could also be replaced with the last valid value, in this example 1.58, or the value could be replaced with the average of the last n Values, for example with $\frac{1.56+1.58}{2} = 1.57$. For the datapoint with the ID 3 there are also different approaches. Either the machine was indeed not running then it would make sense, to set the production_speed to 0.0, if short downtimes for this machine are very unlikely then the machine_running value could be set to True. A possible result after the data cleansing is shown in Table 3 [5]

ID	timestamp	production_speed (meter/minute)	machine_running
0	2021-12-01T12:00:00.000	1.56	True
1	2021-12-01T12:01:00.000	1.58	True
2	2021-12-01T12:02:00.000	2.00	True
3	2021-12-01T12:03:00.000	1.50	True
4	2021-12-01T12:04:00.000	1.50	True
5	2021-12-01T12:05:00.000	1.49	True

Table 3: Example of IoT sensor data after cleansing

4 Outlier detection

Outliers can be categorized as point outliers or subsequence outliers.

Point outliers

A point outlier is a single datapoint that strongly varies from the usual trend of the datapoints. [6]

Subsequence outliers

Subsequence outliers are multiple consecutive datapoints that strongly vary from the usual trend of the datapoints. [6]

Furthermore outliers can be divided into local and global outliers.

Local outliers

A local outlier has a greater variance to its direct neighbouring datapoints (previous and next one) [6]

Global outliers

Whereas a global outlier varies more in regard to all datapoints. To do: Add charts that visualize point, subsequence, local and global outliers [6]

4.1 Outlier detection approaches

Outlier detection methods can be divided into the following groups

Statistical

For statistical outlier detection, historical data is taken to develop a model that pictures the expected behavior of the data. An example of a statistical outlier detection is the threshold based method described in section 4.2 [7, 8]

Distance based

For this approach a distance metric needs to be defined, (e.g. Euclidean distance). Then each datapoint is compared to the data preceding it. The greater the distance between the current and previous datapoints the greater the probability of an anomaly. [7–9]

Clustering

Clustering also requires a set of historical data in order to train the clustering model. Usually the data is clustered into two clusters: normal data and anomalous data. Depending on the distance of a new datapoint to the "normal" and the "anomalous" cluster it is classified. [7–9]

Predictive

In this approach a prediction model needs to be developed, based on previous data. The prediction of this model is then compared with the actual datapoint (new data, which was not used in training the model). If the actual datapoint differs too much from the prediction it is labelled as an anomaly. [7, 8]

Ensemble

as the word ensemble suggests, this is a collection of outlier detection methods that use a specific vote mechanism to determine whether a datapoint is faulty or normal. For example using the majority vote system and a statistical, distance based and predictive method to detect outliers. If at least two methods flag a datapoint as an outlier the ensemble reports it as an outlier as well. If only one method reports it as an outlier the ensemble does not flag it as an anomaly. [7]

4.2 Threshold based outlier detection

Threshold based detection methods are able to identify outliers based on a given threshold τ. These Methods can be described with the following formula

$$|x_t - \hat{x}_t| > \tau \text{ [6]}$$

Where x_t is the actual value and $\hat{x}_t$ is the expected value and τ is a given threshold.
Methods to calculate $\hat{x}_t$ will be described in the following sections. Furthermore $\hat{x}_t$ can be calculated using the entire data series or with subsets (of equal length) of the entire data series. This means $\hat{x}_t$ can be either calculated for the whole data series or for just a segment.
Depending on the sensitivity wanted for outlier detection an appropriate τ needs to be chosen. The greater τ is the fewer outliers will be detected. The smaller τ is the more outliers will be identified. [6]

Mean

$$\text{mean} = \overline{x} = \frac{1}{n}\sum_{t=0}^{n} x_t$$

Where n is the total number of samples. Using the mean as an expected value is not robust to outliers, because the median is not as robust as the mean in hindsight to outliers. To calculate the mean all datapoints of a series must be summed up and then divided by the number of datapoints.

Median

If n is odd:

$$median(x) = x_{(n+1)/2}$$

If n is even:

$$median(x) = \frac{x_{n/2} + x_{(n+1)/2}}{2}$$

Where x is a dataset of n elements ordered from smallest to largest
($x_1 \leq x_2 \leq x_3 \leq \ldots \leq x_{n-2} \leq x_{n-1} \leq x_n$) [6] To calculate the median all values must be sorted from smallest to largest. If the number of datapoints is odd then the most center datapoint is the Median (e.g. if the series consist of 7 values the third value is the median). If the number of datapoints is even then the median is the mean of the two datapoints in the center.

Median Absolute Deviation (Mean Absolute Deviation (MAD))

The Median Absolute Deviation

$$MAD = median(|x_t - median(x)|)$$

MAD is a more robust (regarding outliers) way to calculate the deviation of a dataset. To calculate the MAD firstly the median of the dataset must be calculated. Then the absolute difference between x_t and the median of the dataset is calculated. The Median of all differences results in the MAD [10, 11]

5 Conclusion

This paper provides an overview of the topic anomaly detection. It provides a description for key features of data quality, and introduces the topic of data cleaning and data cleansing. Furthermore this paper provides general overview of outlier / anomaly detection approaches. Lastly the threshold based outlier detection is further elaborated.
There are countless methods to detect anomalies in data. There is not a go-to approach that suits all needs. It is required to assess different approaches for different applications, in order to get the best result. This paper should provide an overview of approaches to detect outliers / anomalies. It depends on the use case which method to detect outliers has the highest success rate.

Bibliography

[1] L. Cai and Y. Zhu, “The Challenges of Data Quality and Data Quality Assessment in the Big Data Era,” vol. 14, no. 0, p. 2.

[2] S. Song and A. Zhang, “IoT Data Quality,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. Association for Computing Machinery, pp. 3517–3518.

[3] “Apache Parquet.” [Online]. Available: https://parquet.apache.org/

[4] L. Li, J. McCann, N. S. Pollard, and C. Faloutsos, “DynaMMo: Mining and summarization of coevolving sequences with missing values,” in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’09. Association for Computing Machinery, pp. 507–516.

[5] J. I. Maletic and A. Marcus, “Data Cleansing: Beyond Integrity Analysis,” p. 10.

[6] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano. A review on Outlier/Anomaly detection in time series data.

[7] A. A. Cook, G. Mısırlı, and Z. Fan, “Anomaly Detection for IoT Time-Series Data: A Survey,” vol. 7, no. 7, pp. 6481–6494.

[8] F. Giannoni, M. Mancini, and F. Marinelli. Anomaly Detection Models for IoT Time Series Data.

[9] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” vol. 41, no. 3, pp. 15:1–15:58.

[10] C. Leys, C. Ley, O. Klein, P. Bernard, and L. Licata, “Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median,” vol. 49, no. 4, pp. 764–766.

[11] S. Mehrang, E. Helander, M. Pavel, A. Chieh, and I. Korhonen, “Outlier detection in weight time series of connected scales,” in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 1489–1496.

List of Figures

List of Tables

List of Abbreviations

IoT Internet of Things

CSV Comma Separated Values

JSON Java Script Object Notation

MAD Mean Absolute Deviation

LSTM Long Short-Term Memory

GRU Gated Recurrent Unit

ARIMA AutoRegressive Integrated Moving Average

GP Gaussian Process