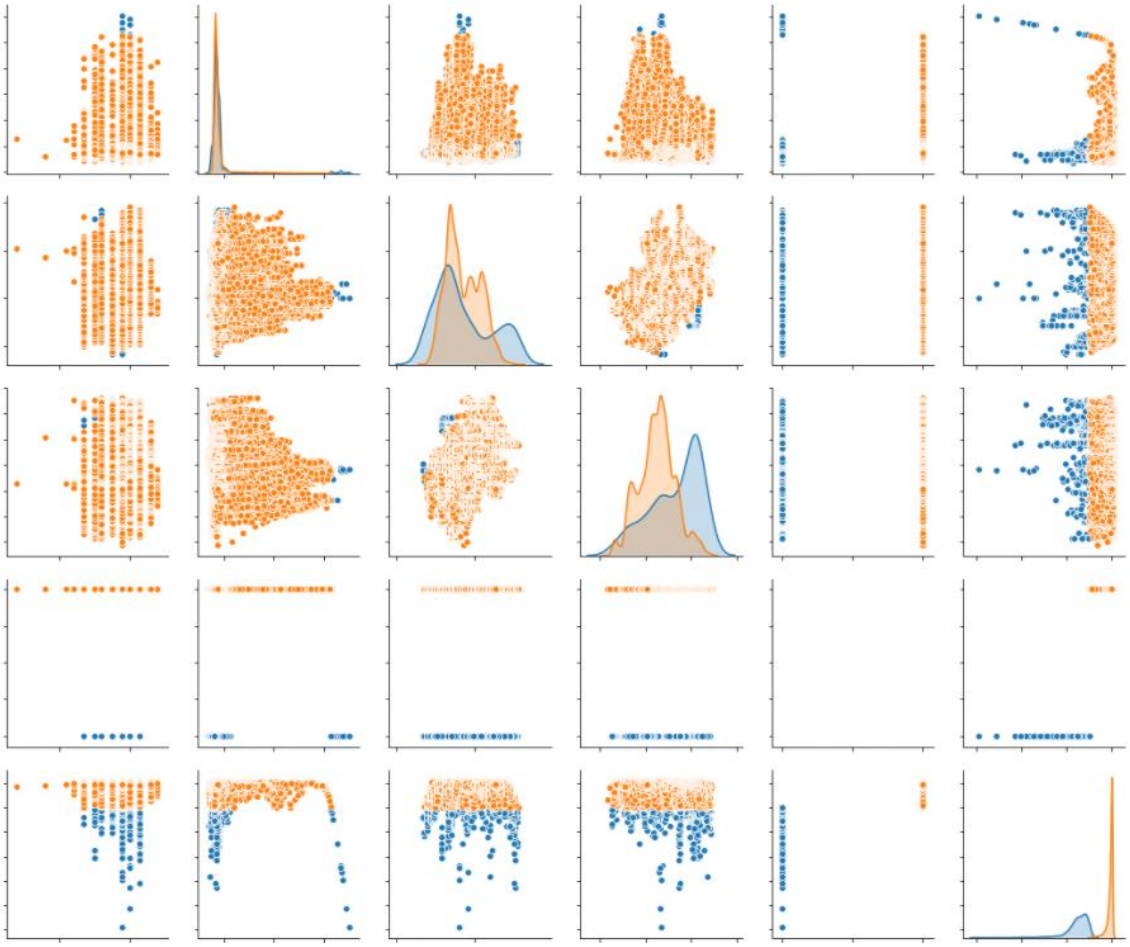
Algorithms for Validation of IoT Data



***Mads Johansen**, Specialist,*
IoT, Data & Services Innovation, FORCE Technology (majh@force.dk)

***Ashok Debnath,** Senior Specialist,*
IoT, Data & Services Innovation, FORCE Technology (asde@force.dk)

***Anders P. Mynster**, Head of Department,*
IoT, Data & Services Innovation, FORCE Technology (apm@force.dk)

***Alexandre Alapetite**, Principal Software Engineering Specialist,*
AI and Data Analytics, Alexandra Institute (alexandre.alapetite@alexandra.dk)

***Bob Pepin**, Senior AI Specialist,*
AI and Data Analytics, Alexandra Institute (bob.pepin@alexandra.dk)

Table of Contents

Abstract 3

1. Introduction 4

2. Data Validation Steps 4

3. Data Format, Schema and Content 4

4. Schema validation 5

5. Aggregation 6

6. Valid Ranges, Distributions and Correlation 7

7. Anomaly Detection 9

7.1. Basic anomaly detection 10

7.2. Advanced anomaly detection 11

7.3. Clustering-based Anomaly Detection 13

7.4. Time Series Anomaly Detection 13

8. Signal processing 14

9. Conclusion 15

References 15

Abstract

IoT is growing and more and more devices are coming online. All these devices generate a lot of data but how do you make sure that the data is correct and trustworthy? In addition, as IoT systems grow, so does the amount of data that they generate. Previously it was possible to manually inspect and correct errors in the systems, but recently we have seen systems with 6, 600 and 60.000 million datapoints per year, making automated algorithms for data validation a necessity. This white paper aims to give an overview of different methods and algorithms for validating IoT data and making sure, that the gathered data is correct and trustworthy, i.e. useful. If the data cannot be trusted, then how can you trust the outcome of the analysis performed on that data. The paper goes through the different steps of the data validation starting from simple checks on data format and schema to algorithms for anomaly detection. Some algorithms could be viewed as part of the data analysis but in some cases, it will be necessary to analyze the data to a certain extend in order to validate it. After doing this validation the next step will be to look at the results and figure out why the data then looks the way it does, i.e. tracing the anomalies back to the root cause. Although that part will not be covered in this paper.

The IoT systems are being used in many diverse applications that are part of our life and their use is growing exponentially. So, over a period of time sensors are prone to failure, malfunction, rapid attrition, malicious attacks, theft, and tampering. All these conditions cause the IoT system to produce unusual and erroneous readings. Any faulty, erroneous, corrupted sensor readings corrupt the trained models, which thereby produces abnormal processes or outliers that are significantly distinct from the normal behavioural processes of a system. In this paper we propose different types of data validation algorithms for detecting different types of data faults on data collected from IoT sensors.

1. Introduction

IoT systems can generate huge amounts of data. All this data is gathered to gain insights, simplify tasks, improve performance of machines or processes and so on. This is part of the data driven decision trend, which is part of the general digitalization. There are many cases for gathering data, but when all this data is gathered what do you do with it? It needs to be analyzed to learn from it, that is the purpose of gathering the data.

When the data is going to be analyzed and used as argumentation for decisions or actions, it needs to be reliable, correct, trustworthy, and applicable. But how do you make sure that the data is correct, reliable, trustworthy, and applicable? The answer is data validation. The data needs to be validated to make sure it complies with all the requirements of the analysis. If the data is not validated any errors or anomalies could affect the data analysis to a degree where the analysis becomes useless or impossible to carry out. Furthermore, the amount of data gathered makes it time-consuming and difficult to manually review the data to check for errors or anomalies. This white paper aims to provide lists of algorithms for validating IoT data based on the data validation categories and activities from "IoT Data Validation: What is data validation and Why validate data?" FORCE Technology, 2021. As the topic is very broad and there are many aspects to cover as well as new algorithms being developed, the lists will not cover all possible algorithms but give an overview of the available options for doing data validation.

The white paper will follow a suggested order of the data validation steps by first giving a brief description of these steps. Then it will go through the steps where the data validation becomes more detailed, starting with overall data format checks and moving on to more advanced machine learning methods for anomaly detection. There will also be a section on signal processing at the end to describe certain cases where further processing is needed on the data as part of the validation, but also as processing/preparation before the validation with machine learning anomaly detection. Finally, the paper will be concluded.

2. Data Validation Steps

There are different steps or stages of data validation where different kinds of algorithms are used. The first step is to validate the schema and format of the data, next there is the possibility of visualizing the data in different ways. It could be to simply plot the data and investigate it manually.

There is also the possibility of investigating the distribution of the data in different ways or investigating the correlation between variables in the data. Distributions and correlations can be done both visually and with certain algorithms. When visualizing there is also the possibility of aggregating the data in different ways to simplify the visualization.

This can be useful when there is a lot of data. Finally, there is the anomaly detection algorithms for finding anomalies (also called outliers) in the data. There are several different algorithms for this which can also be separated into groups as will be described later.

3. Data Format, Schema and Content

Checking that the input data complies with a formal schema is a commonplace first step of a data quality process. The focus is to make sure the data is consistent and that its format doesn't change unexpectedly.

This for example means that the number of columns is the same, the data is readable by the system which is used for analysis, the data types, length and units are consistent through the data and that potential missing values are handled. Missing values could be fixed by for example imputation or interpolation, or they could be disregarded. The mentioned checks are summarized in Table 1.

Data format checks
Number of columns
Unrecognized characters
Data types
Data length
Units
Missing data values

Table 1: Different checks performed to ensure consistency in data format.

4. Schema validation

When the need of validation becomes more complex than just checking for instance the number of columns, it quickly becomes necessary to use a more formal and automated method, such as based on schemas. [“IoT Data Validation: What is data validation and Why validate data?” FORCE Technology, 2021] There are multiple schema languages and technologies, the choice of which depend mainly on the following aspects:

- underlying data format (XML, JSON, CSV, RDF…)
  - for XML: DTD, XML Schema, Relax NG
  - for JSON: JSON Schema
- availability of existing schemas, especially when using standard data models
  - e.g., from FIWARE Smart Data Models[1]
- complexity of the validation rules to implement (e.g., dynamicity, cross-references)
  - e.g., Schematron for more sophisticated business logic rules
- size and velocity of the input data
  - e.g., XML Schema is well suited for very large documents, or data streams, which might not fit in memory
- integration with computer languages and frameworks
  - DTD being the most ancient but also most available technology, followed by XML Schema, and then Relax NG
- familiarity of the development team, or barrier to entry
  - e.g., with Relax NG being easier than for instance XML Schema, but XML Schema might already be used in other parts of the IT infrastructure such as SOAP services

Furthermore, the deployment of the schema validation process can also be done in a few different ways. In particular, online schemas (i.e., from an HTTP address) are quite usual for standard data formats, and while convenient for some quick tests, using online schemas can cause multiple problems, such as:

- scalability: when there is a need to validate many data samples, it would be too slow to fetch schemas from the Internet each time.
- stability: online resources might not always be available, or one might get banned if querying too many schemas.
- reproducibility: online schemas might change address or change version, which might break the validation process.
- security: retrieving and processing schemas blindly from the Internet might expose to threats such as denial of service, memory exhaustion, and even remove code execution.

[1] https://smartdatamodels.org

To mitigate those issues, a proper caching and versioning mechanism must be used, not only for the main schemas, but also for the potentially numerous references to other sub-schemas. Most validation engines have options to control the dereferencing of schemas to implement those recommendations.

Before the actual validation, two more steps might be needed:

- resolution: for endpoints receiving heterogenous data, it is necessary to identify the more precise schema to apply. Solutions using a declarative approach – e.g., based on XPath or XSL for XML inputs, and JavaScript-compatible expressions or JSONata for JSON inputs – and thus avoiding hardcoded rules inside a software, make the approach easier to maintain and scale.
- transformation: a data structure might have multiple possible serializations (textual or binary representations), in which case a transformation to the best form to be compatible with the corresponding schema. Tools such as XSLT (for XML inputs) and JSONata (for JSON inputs) can be helpful.

As an example, at the Nordic IoT Centre, for demos, prototypes, or even simple use cases, we sometimes use our open-source tool “json-multi-schema”[2].

5. Aggregation

When the data format is consistent it can be read by a program or other visualization tool to create graphs of the data. Many IoT systems generate data over time resulting in time series data, for example a temperature sensor measuring the temperature of a room every five minutes. The simplest way of validating this data is to simply plot it over time. But if there is a lot of data making it computationally difficult to handle or difficult to view in the graph because of the number of measurements then it might be an idea to aggregate the data. Aggregation reduces the number of data-points, making the data computationally lighter to handle, especially before plotting into different views. Using aggregation isn’t always the answer when validating data because it also reduces the details in the data. Thus, it is important to note that it is used to get an overview and find general errors or events in the data and not to find specific detailed errors.

Aggregation functions
Average
Min
Max
Median
Mode
Range
Sum
Count

Table 2: List of aggregation functions.

Because of the IoT data often being time series data, it is ideal to aggregate data over time. For example, if you have the temperature sensor mentioned before and have data for a few years then it will be infeasible to plot all data points in one graph. But if you aggregate the data such that you have one data point per day then it will be much easier to visualize. But what should you do with the data points within that time interval?

[2] https://flows.nodered.org/node/node-red-contrib-json-multi-schema

There are several different ways to aggregate data to gain different insights. Table 2 shows a list of different aggregation functions, where the most popular one is the average. This function will give an average of the data values in the given time interval. Similarly, for min, max, median, and sum. Mode is the value which occurs most often in the time interval, count counts the number of data points in the time interval, which could give an idea of lost data points or measurements, and range gives the difference between the minimum and maximum value in the given time interval.

As mentioned, aggregation reduces the details of the data and for example when using the average aggregation function the high and low values are lost which means outliers become hard to see. Therefore, one might have to do several kinds of aggregation functions since each of them has its strengths and weaknesses.

6. Valid Ranges, Distributions and Correlation

This section focuses on checking distributions and correlation between variables. As the previous section many of these methods use visualization as a way of investigating whether the data fits the expectations. There are a lot of unknowns when it comes to IoT data. Of course, it is known what kind of device and/or sensor the data is coming from but there might be many unknown parameters regarding the environment in which the device is located. But in some cases, certain things are known, and this section includes several methods for validating data which is mostly useful when certain aspects of the environment is known.

First it is possible to perform range checks on the data. If the maximum and/or minimum possible values of the data is known, it is easy to apply these to see if any data points go beyond these thresholds. But even if neither the minimum nor the maximum is known it is possible to do some simple sanity checks on the data. For example, if a temperature sensor reports values of -1000ºC those values are certainly wrong and should be regarded as outliers and not only are they outliers, they are caused by errors in the hardware. In some special cases it might also be the case that the data is known to follow a specific function or known to be from a specific distribution. Then it will be possible to try and fit the function or distribution to the data and if it fits well this could be part of validating that the data is correct. If not, then the data might be affected by some unknown parameter. Fitting a function to the data can be used to see if the pattern of the data is as expected. If for example looking at outdoor temperature the values should be lower during mornings and evenings and higher during the day. This means the data follows a parabola like function over time or have an upside-down u-shape. When fitting functions different metrics can be used to assess the quality of the fit, such as R^2. R^2 is the coefficient of determination and describes how well the function fits. There are cases where it can be negative, but it usually ranges from 0 to 1, where it is 1 if the data fits perfectly and 0 if there is no fit at all. Of course, here it is necessary to establish some threshold which the R^2 value should exceed to be able to conclude that the data fits the given function. Furthermore, if the data changes too fast, e.g. a temperature dropping an unusual amount over a short period of time, a linear function could continuously be fitted on for example the latest 100 data points to see if the change is too steep, i.e. it can be used on a subset of the data.

Overall checks
Max
Min
Function

Table 3: Overall data checks.

There are different ways to check the distribution of the data. A histogram separates the data values into bins (or intervals) and counts how many values fall within each bin. This can be used to illustrate the distribution of the data. The Q-Q (quantile-quantile) plot is also a graphical method but instead of simply showing how the data is distributed it uses the quantiles to compare the distribution of two datasets or compare the distribution of a dataset to a theoretical distribution to see if it fits. If the points in the plot falls approximately on the diagonal (y = x) then the two sets are from

the same distribution (or the set fits the theoretical distribution). The farther from the diagonal the more likely it is that the distributions are different. A goodness of fit test can be used to calculate whether data fits a certain distribution. There is several different goodness of fit tests each with its own restrictions, but the most common one is the chi-square goodness of fit test. Some of the less common ones are for example Kolmogorov-Smirnov, Anderson-Darling, or Shipiro-Wilk test. A goodness of fit test like the chi-square test uses a null hypothesis stating that the data set (or sample) fits a given theoretical distribution. Depending on the result of the test the null hypothesis is rejected, meaning the data does not fit the distribution or the hypothesis is not rejected which means the data seems to fit the distribution, but it is not enough to conclude that the data is from the given distribution. The main methods for testing or investigating distributions are shown in Table 4 and an overview of different distributions are shown in Table 5, see for example [1] for further reading.

Testing distributions
Histogram
Q-Q plot
Goodness of fit

Table 4: Testing or investigating the distribution of data.

Examples of distributions
Normal (Gaussian)
LogNormal
Chi-squared
t-distribution
Weibull
Logistic
LogLogistic
Gamma
Binomial
Uniform
Exponential
Largest extreme value
Smallest extreme value
Poisson

Table 5: Examples of different distributions. [1]

Most of the methods described so far are univariate, which means they are performed on one column of the dataset (also called feature, attribute, or variable) at a time, but some of them can also compare features to each other like the Q-Q plot can help determine if the data from two features are from the same distribution. When looking at more features at the same time it is called multivariate analysis and can be used to examine the relationship between features.

Some of the simpler multivariate methods are used for finding correlations between features can be seen in Table 6. The scatter plot simply plots the data from two given features. It can then be used to see if there is a pattern or correlation between the two given features. For example, a linear correlation where values of the second feature increases as the values of the first feature increases. There is also the scatterplot matrix where a scatter plot is made for each combination of two features from the entire data set. This will give an n*n matrix where each cell is a scatterplot. In this plot the diagonal will plot the same features against each other which is not that useful, and all possible combinations of the features are shown on each side of the diagonal. All in all, this means only half of the scatterplot matrix is needed.

The correlation matrix is like the scatterplot matrix but instead of each cell being a scatter plot it is simply a value (coefficient) which describes the correlation between the two given variables. There are different methods of calculating a correlation coefficient and some of the more popular ones are Pearson's, Spearman's, and Kendall's correlation coefficient. Pearson's can only assess whether the relationship (correlation) between the variables is linear while Spearman's and Kendall's can assess whether the relationship is monotonic.[3] A monotonic relationship between two variables, e.g. x and y, means that y either never increases or never decreases as x increases, i.e. y goes either up or down but not necessarily at the same rate. Thus, if the relationship between x and y describes a parabola, it is non-monotonic.[4]

Correlation algorithms
Scatter plot (and scatterplot matrix)
Correlation matrix
Pearson correlation (linear)
Spearman correlation (monotonic)
Kendall correlation (monotonic)

Table 6: Correlation algorithms.

7. Anomaly Detection

Anomaly (or outlier) detection is a large subject and there are many different algorithms for finding anomalies. These algorithms can be separated into many different types which work on different kinds of data as they assume different properties of the data they are used on. Furthermore, there are three main categories of anomalies:

- Point anomalies
- Contextual anomalies
- Collective anomalies

A point anomaly is a single data point which lies far from the rest of the data points and is also known as a global outlier. Contextual anomalies are common in time series data as they are outliers in the context of the data points it is part of while it might not be an anomaly in a different context. For example, if a temperature sensor reports 28°C in a given city in the summer, it is probably a correct measurement. But if the same sensor in the same city also reports 28°C during winter it is likely an anomaly because of the time of year. So, the value is fine in general but not in the

[3] https://en.wikipedia.org/wiki/Spearman%27s_rank_correlation_coefficient

[4] https://blog.majestic.com/case-studies/majesticseo-beginners-guide-to-correlation-part-5/

given context. A collective anomaly is a collection of data points which differ from the rest as a group while the individual data points might not be anomalies themselves.[5,6]

There are two types of anomaly detection:

- Anomaly detection (outlier detection)
- Novelty detection

Usually, anomaly detection and outlier detection are used interchangeably. Anomaly detection is similar to novelty detection but differ in one important aspect. In anomaly detection the training data is polluted by outliers while in novelty detection the training data is not polluted by outliers.[7]

There are also different types of learning methods within machine learning. These relate to how the data is labeled when training the algorithms:

- Supervised
- Semi-supervised
- Unsupervised

In supervised learning the data has been labeled meaning each data point or input has been marked with a desired output. In unsupervised learning none of the data points have been marked and thus the outputs are unknown. Semi-supervised learning lies between these since it uses a small amount of labeled data together with a large amount of unlabeled data when training. The learning approach for many of the anomaly detection methods in this paper is unsupervised. This is due to the fact that labelling data can be a time-consuming process and since the data generated by the IoT system needs to be validated the only known thing is that there are normal data and data which are anomalies due to unknown parameters from the IoT system. The data validation is about locating these anomalies. Therefore, the validation process will rarely include labeled data.

7.1. Basic anomaly detection

Before going into the more advanced anomaly detection it is worth mentioning a couple of simple and common algorithms for finding outliers. These are for example the boxplot and z-score algorithms. These are both univariate which is why they are described here as basic algorithms.

Basic anomaly detection
Boxplot
Z-score

Table 7: Simple algorithms for visualizing and/or defining outliers.

The boxplot is made for a single feature in the data set and describes the distribution of the data but not in the same way as the other methods for finding distributions such as the histogram. It defines the median (50th percentile or Q2), 25th percentile (Q1) and 75th percentile (Q3). The range from Q1 to Q3 defines the interquartile range (IQR). Furthermore, the boxplot defines whiskers which are the minimum and maximum of the range of data which are not outliers. The minimum is Q1-1.5*IQR and the maximum is Q3+1.5*IQR. Everything below the minimum and above the maximum is defined as outliers. So, where the histogram simply illustrates the distribution of the data the boxplot both visualizes it and gives a definition of the outliers.

[5] https://blogs.oracle.com/datascience/introduction-to-anomaly-detection

[6] https://www.anodot.com/blog/quick-guide-different-types-outliers/

[7] https://scikit-learn.org/stable/modules/outlier_detection.html

The z-score is another univariate method for finding outliers. It is used to indicate how many standard deviations a value is from the mean and assumes a gaussian distribution. It is parametric since it assumes a gaussian distribution but if the data is described by a gaussian distribution it is possible to make transformations to the data such that the z-score can be used. The z-score can now be calculated for each value, but the threshold must be defined manually. The usual thresholds are 2, 2.5, 3, or 3.5 standard deviations. Then the outliers will simply be the values which lie beyond this threshold.[8]

Even though the boxplot and z-score methods seem similar they define the outliers differently. The outliers of the boxplot are defined based on the IQR which again is calculated using the quartiles and represents the middle 50% of the data. The outliers of the z-score are defined using standard deviation which is based on the mean of the data. The mean is sensitive to outliers meaning that potential outliers in the data affects the mean. The quartiles used to calculate the IQR are robust to outliers. Due to this the two methods are also referred to as a sensitive and resistant measure of spread.[9]

7.2. Advanced anomaly detection

Some of the more advanced anomaly detection methods include the Mahalanobis and the Minimum Covariance Determinant (MCD). These both use distance from a data point to the center of the data cloud to determine the outliers and are both multivariate. These are shown in Table 8. Other advanced anomaly detection methods include clustering-based algorithms which can be further subcategorized in certain kinds of clustering. There are also other algorithms using different approaches such as support vector machines (SVM) which are usually a supervised learning approach, but the One Class SVM is unsupervised. Then there is a density-based approach used by the local outlier factor, a tree-based approach used by isolation forest, and other approaches.

Anomaly detection
Mahalanobis distance
Minimum covariance determinant
Principal component analysis
Local outlier factor
Isolation forest
One class support vector machine
Elliptic envelope

Table 8: Advanced anomaly detection algorithms.

Principal component analysis (PCA) [2] is often used for reducing the feature space and usually used in machine learning to reduce the number of inputs. It takes the entire data set (all features) and transforms the data to a new coordinate system. It is done by first finding a vector (or line) through the data which captures the most variance, then finding a new line which is orthogonal to the first and again captures as much variance as possible. The third line must be orthogonal to the first two and so on. Each of these lines are the new coordinates called principal components. Since the principal components will be sorted by how much variance they describe, with the first principal component describing the most, it might happen that the first three components describe more than 90% of the variance in the data set and this is exactly why it is used for dimensionality reduction. Instead of using all the normal features from the data set one can use for example only three and still describe a lot of the variance.[10] Thus, it can be used as

[8] https://towardsdatascience.com/z-score-for-anomaly-detection-d98b0006f510

[9] https://online.stat.psu.edu/stat100/lesson/3/3.5

[10] https://towardsdatascience.com/not-1-not-2-but-5-ways-to-correlate-6ac92cf42f0f

preprocessing before using other machine learning algorithms, but it can also be used for anomaly detection in different ways. See for example [3] and [4].

The Mahalanobis distance is a more advanced way of measuring the distance from a data point to the mean, like the z-score. The Mahalanobis distance is closely related to PCA as both PCA and Mahalanobis distance requires the principal components of the data, i.e. transforming the data by changing the coordinate axes to be based on the variance in the data. If C is the covariance then the Mahalanobis distance between two coordinates, x and y, is equal to the distance between x and y divided by the square root of $C(x-y,x-y)$.[11] [5]

MCD is another outlier detection method. It resembles the Mahalanobis distance as they both use distances from the mean to find outliers, but they are finding the mean in different ways. The MCD is more robust to outliers as it in a way disregards them to find a more suitable boundary of the non-anomaly data points and thus finds more outliers. The Mahalanobis distance is more affected by the outliers in the data resulting in what is known as the "masking effect" where some data points which should be outliers are not marked as outliers. [6]

The local outlier factor algorithm looks at the density of a data point and compares it to the density of its k nearest neighbors, where density refers to how close the data point is to its neighbors. If the density of a data point is much less than the density of its neighbors, it is an outlier. The local outlier factor computes a score to determine this. If the score is around one, the density is similar to that of its neighbors. If it is less than one, the density is higher and if the score is more than one, the density is lower and if it is much more than one, it will likely be an outlier.[12]

The Isolation forest is an unsupervised anomaly detection algorithm based on isolation trees (similar to decision trees). An isolation tree is made upon the data by picking a feature and finding a random value within the range of values thereby splitting the range into two. This is done multiple times to make the isolation tree. The idea is then that an outlier will have a shorter path from the root compared to a normal data point. The reason it is called isolation forest is that the algorithm makes a lot of isolation trees, since each tree is randomly generated, and if a data point is categorized as an anomaly by a large amount of the isolation trees, then it is an outlier. This way a score is computed for each data point and the higher the score the more likely it is to be an anomaly.[13] [7]

The one class SVM works differently. It can create a non-linear decision boundary by projecting the data through a non-linear function to a space with a higher dimension. After the projection it fits a hyperplane through it to define the decision boundary. For example, if you have data in two dimensions which more or less forms a circle the one class SVM can lift the data using the non-linear projection to three dimensions and fit a two-dimensional hyperplane through it to create a circular decision boundary. The data points beyond this boundary will be outliers.

Another important subject to mention in relation to machine learning algorithms is normalization of data. Some features in the data might range from zero to one while other features might have values ranging from 0 to 1000. This can create an instability in the data and consequently the developed machine learning model. To improve stability the data can be normalized which means bringing all features to a common scale. Several types of algorithms can be affected by the instability of the data if not normalized, for example the SVM-based algorithms. But other algorithms are not affected such as the Isolation Forest.[14,15] For more information on the elliptic envelope algorithm see [8].

[11] https://stats.stackexchange.com/questions/62092/bottom-to-top-explanation-of-the-mahalanobis-distance

[12] https://towardsdatascience.com/local-outlier-factor-for-anomaly-detection-cc0c770d2ebe

[13] https://medium.com/@parulsharma_8357/isolation-forest-62fcb1de4cf5

[14] https://medium.com/@jamesstradling/unsupervised-machine-learning-with-one-class-support-vector-machines-129579a49d1d

[15] https://towardsai.net/p/data-science/how-when-and-why-should-you-normalize-standardize-rescale-your-data-3f083def38ff

7.3. Clustering-based Anomaly Detection

Table 9 shows a list of different clustering algorithms. Clustering means that the algorithm will try to find groups (or clusters) in the data. Depending on the specific algorithm some data points are difficult to cluster or may be separated into a cluster of its own and this might indicate that the data point is in fact an anomaly.

Clustering algorithms
DBSCAN (density-based)
OPTICS (density-based)
k-means (centroid-based)
Gaussian mixture model (distribution-based) using expectation-maximization
Single-linkage (connectivity-based/hierarchical), also complete-, average- and centroid-linkage

Table 9: Clustering-based anomaly detection algorithms.

The DBSCAN method looks for high density areas in the data and assigns clusters to them. Data points in less dense regions are excluded from these clusters and marked as anomalies.[16]

The k-means algorithm separates the data into *k* clusters where the user has to input the number *k*, but how would you know how many clusters to separate the data into? This is one of the problems of k-means and one of the solutions is called the elbow method. Using the elbow method means running the algorithm on a range of values of *k*, for example 1 to 100. Then for each value of *k* the sum of the squared errors (SSE) is calculated. This is then plotted with *k* on the x-axis and SSE on the y-axis. When choosing *k* the SSE should be small but the SSE decreases when *k* increases, so when *k* is equal to the number of data points the SSE is zero. Having a cluster per data point is not useful. Thus, the value of *k* should be found where the rate of change of the SSE is still fairly high, but the SSE is fairly small which is often in the “elbow” of the graph. The k-means have other drawbacks which limits its use, but it is still mentioned here as it is a popular and simple clustering algorithm. The outliers could then be found as data points which lie far from the centroid of each cluster.[17] For further reading on the different hierarchical clustering methods see [9] and for Gaussian Mixture model see [10].

7.4. Time Series Anomaly Detection

Time series or sequential data anomaly detection is another type of anomaly detection where contextual anomalies are the main objective. Where the earlier algorithms are focused on point anomalies where the sequence leading up to the anomaly is not taken into account, the time series anomaly detection is about exactly that. As mentioned earlier this type of anomalies might not be anomalies looking at them individually but looking at the context in which they appear they might differ from the nominal data.

Time series anomaly detection
Long Short-Term Memory Autoencoder

Table 10: Time series anomaly detection algorithms.

[16] https://towardsdatascience.com/best-clustering-algorithms-for-anomaly-detection-d5b7412537c8

[17] http://www.mit.edu/~9.54/fall14/slides/Class13.pdf

The Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) which is made for sequences of data as they can remember information across a long input sequence. The LSTM can be used in an encoder-decoder architecture called an autoencoder. The autoencoder consists of an encoder taking an input and compressing it to a smaller feature space essentially learning the important parts of the input. The encoder is followed by a decoder which takes the compressed output from the encoder as input and tries to recreate it. The LSTM autoencoder then uses an LSTM as an encoder and one as decoder. An important thing to note here is that even though the autoencoder is an unsupervised learning algorithm it is trained using supervised method. They are usually trained on clean data without anomalies or assumed to not have anomalies and then tested on data with anomalies. This makes it more of a novelty detection algorithm and in a mix of supervised and unsupervised learning because the training data is assumed to be from one class making it supervised but then another class is introduced when testing and determined by looking different than the training data.

8. Signal processing

As an extension of the anomaly data validation algorithms, it might be necessary to do some processing of the data before doing the validation such as anomaly detection. In some cases, the data points will not be a single point represented by a number of features instead one data point (or sample) might consist of a sequence of data points from which it might be necessary to extract features before doing anomaly detection. Such a case could be within the area of signal processing. For example, one might have an accelerometer installed on a rotating machine to measure the vibrations. But for this kind of problem, it is not sufficient to only do one measurement with the accelerometer every few minutes. Depending on the use case it could be necessary to make a sequence of measurements for example 1000 measurements over one second. This sample of 1000 measurements will of course need to be validated but it is not always the best solution to do anomaly detection on the raw sample. Instead, this might need to be processed in order transform the sample of 1000 data points to one data point represented by certain features.

Signal processing algorithms
Fast Fourier Transform
Wavelet Transform
Kurtosis
Skewness
Crest factor
Root Mean Square
Clearance factor
Shape factor
Peak to Peak
Power spectral density

Table 11: Signal processing algorithms.

Signal processing can be done on a variety of things such as vibration and sound. Table 11 shows a list of different signal processing algorithms and signal characteristics. Fast Fourier Transform is an example of an algorithm for transforming the signal from a time domain to a frequency domain. There are other types of transformations as for example the wavelet transform. The rest of Table 11 are examples of different characteristics of a signal. These can be used as individual features to describe a sample measurement. The different characteristics for example describe how extreme the peaks are or the difference between the maximum positive or maximum negative amplitudes.

9. Conclusion

Doing validation of IoT data is important as there are many unknown parameters when IoT devices are gathering data, such as hardware faults or environmental parameters and unexpected events in the environment. Data validation ranges from basic checks in the data files making sure that the values are “legal”. In this step it is easy to catch errors where data values simply are not possible or if they do not adhere to known environmental parameters. Moving further into the validation the methods and algorithms become more advanced ranging from different visualizations of the data giving overall insights to more advanced algorithms for detecting outliers both in the form of point anomalies or even contextual anomalies when looking at sequences of data. There are a lot of algorithms to do different kinds of validation and the lists provided in this paper has given some of them but not all. Some of them has furthermore been elaborated to give an idea of what they can do and how they do it, in order to better understand when to use them. Since several algorithms, especially in the machine learning anomaly detection category, can be very complex and have weaknesses and strengths regarding what type of data it can be used on and what underlying assumptions they make about the data, it is difficult to give a full explanation of the use cases.

References

[1] M. Ciortan, "Overview of data distributions," 10 March 2019. [Online]. Available: https://medium.com/@ciortanmadalina/overview-of-data-distributions-87d95a5cbf0a. [Accessed 1 February 2020].

[2] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine,* vol. 2, no. 11, pp. 559-572, 1901.

[3] C. Callegari, L. Gazzarrini, S. Giordano, M. Pagano and T. Pepe, "A novel PCA-based Network Anomaly Detection," 2011.

[4] V. Flovik, "Machine learning for anomaly detection and condition monitoring," 23 April 2019. [Online]. Available: https://towardsdatascience.com/machine-learning-for-anomaly-detection-and-condition-monitoring-d4614e7de770. [Accessed 12 February 2021].

[5] P. C. Mahalanobis, "On the generalised distance in statistics," *Proceedings of the National Institute of Sciences of India,* vol. 2, no. 1, pp. 49-55, 1936.

[6] M. D. Mia Hubert, "Minimum Covariance Determinant," *Wiley Interdisciplinary Reviews Computational Statistics,* 2010.

[7] K. M. T. Z.-H. Z. Fei Tony Liu, "Isolation-based Anomaly Detection," *ACM Transactions on Knowledge Discovery from Data,* 2012.

[8] M. Alam, "Towards Data Science," 7 October 2020. [Online]. Available: https://towardsdatascience.com/machine-learning-for-anomaly-detection-elliptic-envelope-2c90528df0a6. [Accessed 12 February 2021].

[9] N. S. Chauhan, "KDnuggets," [Online]. Available: https://www.kdnuggets.com/2019/09/hierarchical-clustering.html. [Accessed 12 February 2021].

[10] C. Maklin, "Towards Data Science," 16 July 2019. [Online]. Available: https://towardsdatascience.com/gaussian-mixture-models-d13a5e915c8e. [Accessed 12 February 2021].