How to use machine learning for anomaly detection and condition monitoring

Concrete use case for machine learning and statistical analysis

Vegard Flovik Follow
Dec 31, 2018 · 13 min read ★

In this article, I will introduce a couple of different techniques and applications of machine learning and statistical analysis, and then show how to apply these approaches to solve a specific use case for anomaly detection and condition monitoring.

Digital transformation, digitalization, Industry 4.0, etc....

These are all terms you have probably heard or read about before. However, behind all of these buzz words, the main goal is the use of technology and data to

increase productivity and efficiency. The connectivity and flow of information and data between devices and sensors allows for an abundance of available data. The key enabler is then being able to use these vast amounts of available data and actually extract useful information, making it possible to reduce costs, optimize capacity, and keep downtime to a minimum. This is where the recent buzz around machine learning and data analytics comes into play.

Anomaly detection

Anomaly detection (or outlier detection) is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data. Typically, anomalous data can be connected to some kind of problem or rare event such as e.g. bank fraud, medical problems, structural defects, malfunctioning equipment etc. This connection makes it very interesting to be able to pick out which data points can be considered anomalies, as identifying these events are typically very interesting from a business perspective.

This brings us to one of the key objectives: How do we identify whether data points are normal or anomalous? In some simple cases, as in the example figure below, data visualization can give us important information.

Figure 1 : Anomaly detection for two variables

In this case of two-dimensional data (**X** and **Y**), it becomes quite easy to visually identify anomalies through data points located outside the typical distribution. However, looking at the figures to the right, it is not possible to identify the outlier directly from investigating one variable at the time: It is the **combination** of the X and Y variable that allows us to easily identify the anomaly. This complicates the matter substantially when we scale up from two variables to 10–100s of variables, which is often the case in practical applications of anomaly detection.

Connection to condition monitoring

Any machine, whether it is a rotating machine (pump, compressor, gas or steam turbine, etc.) or a non-rotating machine (heat exchanger, distillation column, valve, etc.) will eventually reach a point of poor health. That point might not be that of an actual failure or shutdown, but one at which the equipment is no longer acting in its optimal state. This signals that there might be need of some maintenance activity to restore the full operating potential. In simple terms, identifying the “health state” of our equipment is the domain of condition monitoring.

The most common way to perform condition monitoring is to look at each sensor measurement from the machine and to impose a minimum and maximum value limit on it. If the current value is within the bounds, then the machine is healthy. If the current value is outside the bounds, then the machine is unhealthy and an alarm is sent.

This procedure of imposing hard coded alarm limits is known to send a large number of false alarms, that is alarms for situations that are actually healthy states for the machine. There are also missing alarms, that is situations that are

will still avoid going too deep into the theoretical background (but provide some links to more detailed descriptions). If you are more interested in the practical applications of machine learning and statistical analysis when it comes to e.g. condition monitoring, feel free to skip ahead to the “Condition monitoring use-case” section.

Approach 1: Multivariate statistical analysis

Dimensionality reduction using principal component analysis: PCA

As dealing with high dimensional data is often challenging, there are several techniques to reduce the number of variables (dimensionality reduction). One of the main techniques is principal component analysis (PCA), which performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the covariance matrix of the data is constructed and the eigenvectors of this matrix are computed. The eigenvectors that correspond to the largest

eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. The original feature space has now been reduced (with some data loss, but hopefully retaining the most important variance) to the space spanned by a few eigenvectors.

Multivariate anomaly detection

As we have noted above, for identifying anomalies when dealing with one or two variables, data visualization can often be a good starting point. However, when scaling this up to high-dimensional data (which is often the case in practical applications), this approach becomes increasingly difficult. This is fortunately where multivariate statistics comes to help.

When dealing with a collection of data points, they will typically have a certain distribution (e.g. a Gaussian distribution). To detect anomalies in a more quantitative way, we first calculate the probability distribution $p(\mathrm{x})$ from the data points. Then when a new example, x, comes in, we compare $p(\mathrm{x})$ with a threshold r. If $p(\mathrm{x}) < r$, it is considered as an anomaly. This is because normal examples tend to have a large $p(\mathrm{x})$ while anomalous examples tend to have a small $p(\mathrm{x})$

In the context of condition monitoring, this is interesting because anomalies can tell us something about the "health state" of the monitored equipment: Data generated when the equipment approaches failure, or a sub-optimal operation, typically have a different distribution than data from "healthy" equipment.

The Mahalanobis distance

Consider the problem of estimating the probability that a data point belongs to a distribution, as described above. Our first step would be to find the centroid or center of mass of the sample points. Intuitively, the closer the point in question is to this center of mass, the more likely it is to belong to the set. However, we also need to know if the set is spread out over a large range or a small range, so that we can decide whether a given distance from the center is noteworthy or not. The simplistic approach is to estimate the standard deviation of the distances of the sample points from the center of mass. By plugging this into the normal

distribution we can derive the probability of the data point belonging to the same distribution.

The drawback of the above approach was that we assumed that the sample points are distributed about the center of mass in a spherical manner. Were the distribution to be decidedly non-spherical, for instance ellipsoidal, then we would expect the probability of the test point belonging to the set to depend not only on the distance from the center of mass, but also on the direction. In those directions where the ellipsoid has a short axis the test point must be closer, while in those where the axis is long the test point can be further away from the center. Putting this on a mathematical basis, the ellipsoid that best represents the set's probability distribution can be estimated by calculating the covariance matrix of the samples. The **Mahalanobis distance** (MD) is the distance of the test point from the center of mass divided by the width of the ellipsoid in the direction of the test point.

In order to use the MD to classify a test point as belonging to one of N classes, one first estimates the covariance matrix of each class, usually based on samples known to belong to each class. In our case, as we are only interested in classifying "normal" vs "anomaly", we use training data that only contains normal operating conditions to calculate the covariance matrix. Then, given a test sample, we compute the MD to the "normal" class, and classify the test point as an "anomaly" if the distance is above a certain threshold.

Note of caution: Use of the MD implies that inference can be done through the mean and covariance matrix — and that is a property of the normal distribution alone. This criteria is not necessarily fulfilled in our case, as the input variables might not be normal distributed. However, we try anyway and see how well it works!

Approach 2: Artificial Neural Network

Autoencoder networks

The second approach is based on using autoencoder neural networks. It is based on similar principles as that of the above statistical analysis, but with some slight differences.

An autoencoder is a type of artificial neural network used to learn efficient data codings in an unsupervised manner. The aim of an autoencoder is to learn a representation (encoding) for a set of data, typically for dimensionality reduction. Along with the reduction side, a reconstructing side is learnt, where the autoencoder tries to generate from the reduced encoding a representation as close as possible to its original input.

Architecturally, the simplest form of an autoencoder is a feedforward, non-recurrent neural network very similar to the many single layer perceptrons which makes a multilayer perceptron (MLP) — having an input layer, an output layer and one or more hidden layers connecting them — but with the output layer having the same number of nodes as the input layer, and with the purpose of *reconstructing* its own inputs.

Figure 2: Autoencoder network

In the context of anomaly detection and condition monitoring, the basic idea is to use the autoencoder network to “compress” the sensor readings to a lower-dimensional representation, which captures the correlations and interactions between the various variables. (Essentially the same principle as the PCA model, but here we also allow for non-linear interactions between the variables).

The autoencoder network is then trained on data representing the “normal” operating state, with the goal of first compressing and then reconstructing the input variables. During the dimensionality reduction, the network learns the interactions between the various variables and should be able to re-construct them back to the original variables at the output. The main idea is that as the monitored equipment degrades, this should affect the interaction between the variables (e.g. changes in temperatures, pressures, vibrations, etc.). As this happens, one will start to see an increased error in the networks re-construction of the input variables. By monitoring the re-construction error, one can thus get an indication of the “health” of the monitored equipment, as this error will increase as the equipment degrades. Similar to the first approach of using the Mahalanobis distance, we here use the probability distribution of the reconstruction error to identify whether a data point is normal or anomalous.

. . .

Condition monitoring use-case: Gear bearing failure

In this section, I will go through a practical use case for condition monitoring using the two different approaches described above. As most of the data we are working on with our clients are not openly available, I have chosen to rather

Approach 1 : PCA + Mahalanobis distance

As explained in more detail in the “Technical section” of this article, the first approach consisted of first performing a principal component analysis, and then calculating the Mahalanobis distance (MD) to identify data points as normal or anomalous (sign of equipment degradation). The distribution of the MD for training data representing “healthy” equipment is illustrated in the figure below.

Figure 3: Distribution of Mahalanobis distance for "healthy" equipment

Using the distribution of MD for "healthy" equipment, we can define a threshold value for what to consider an anomaly. From the distribution above, we can e.g. define a MD > 3 as an anomaly. The evaluation of this method to detect equipment degradation now consists of calculating the MD for all data points in the test set, and comparing it to the defined threshold value for flagging it as an anomaly.

Model evaluation on test data:

Using the above approach, we calculated the MD for the test data in the time period leading up to the bearing failure, as illustrated in the below figure.

Figure 4: Predicting bearing failure using approach 1

In the abo whereas
the red lin aly. The
bearing fa tted line.
This illust pcoming
equipmen he MD
crosses th

We can nc pproach,
in order tc .

Approa

As explained in more detail in the “Technical section” of the paper, the second approach consisted of using an autoencoder neural network to look for anomalies (as identified through an increased reconstruction loss from the network). Similar to the first approach, we also here use the distribution of the model output for the training data representing “healthy” equipment to detect anomalies. The distribution of reconstruction loss (mean absolute error) for the training data is shown in the below figure:

Figure 5 : Distribution of reconstruction loss for ”healthy” equipment.

Using the distribution of the reconstruction loss for “healthy” equipment, we can now define a threshold value for what to consider an anomaly. From the distribution above, we can e.g. define a loss> 0.25 as an anomaly. The evaluation of the method to detect equipment degradation now consists of calculating the reconstruction loss for all data points in the test set, and comparing the loss to the defined threshold value for flagging this as an anomaly.

Model evaluation on test data:

Using the above approach, we calculate the reconstruction loss for the test data in the time period leading up to the bearing failure, as illustrated in the figure below.

Figure 6: Predicting bearing failure using approach 2

In the above figure, the blue points correspond to the reconstruction loss, whereas the red line represents the defined threshold value for flagging an anomaly. The bearing failure occur at the end of the dataset, indicated by the black dotted line. This illustrates that also this modeling approach was able to

detect the upcoming equipment failure about 3 days ahead of the actual breakdown (where the reconstruction loss crosses the threshold value).

Results summary:

As seen in the above sections on the two different approaches for anomaly detection, both methods are successfully able to detect the upcoming equipment failure several days ahead of the actual breakdown. In a real-life scenario this would allow predictive measures (maintenance/repair) to be taken in advance of the failure, which means both cost savings as well as the potential importance for HSE aspects of equipment failure.

Outlook:

With the reduced cost of capturing data through sensors, as well as the increased connectivity between devices, being able to extract valuable information from data is becoming increasingly important. Finding patterns in large quantities of data is the realm of machine learning and statistics, and in my opinion, there are huge possibilities to harness the information hidden in these data to improve performance within several different domains. Anomaly detection and condition monitoring, as covered in this article, are just one of many possibilities. *(Article also available HERE)*

In the future, I believe machine learning will be used in many more ways than we are even able to imagine today. What impact do you think it will have on the various industries? I would love to hear your thoughts in the comments below.

Edit: This article on anomaly detection and condition monitoring has received a lot of feedback. Many of the questions I receive, concern the technical aspects and how to set up the models etc. Due to this, I decided to write a follow-up article covering all the necessary steps in detail, from pre-processing data to building models and visualizing results.

Other articles:

If you found this article interesting, you might also like some of my other articles:

1. Deep Transfer Learning for Image Classification
2. Building an AI that can read your mind
3. Machine Learning: From Hype to real-world applications
4. The hidden risk of AI and Big Data
5. AI for supply chain management: Predictive analytics and demand forecasting
6. How (not) to use Machine Learning for time series forecasting: Avoiding the pitfalls
7. How to use machine learning for production optimization: Using data to improve performance
8. How do you teach physics to AI systems?
9. Can we build artificial brain networks using nanoscale magnets?

AI workshop — From hype to real-world applications

Machine Learning | Data Analysis | Statistics | Predictive Analytics | Towards Data Science