Machine Learning For Anomaly Detection and Condition Monitoring

Introduction

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? See Table 1 for an example table.

actc2	dbtm	dbtv	dmea	dver
bpos	ropa	hkla	hklx	woba
wobx	tqa	tqx	rpma	sppa
chkp	spm1	spm2	spm3	tva
tvca	mfop	mfoa	mfia	mdoa
mtoa	mtia	mcoa	mcia	stkc
lstk	drtm	gasa	spr1	spr2

Table 1: An example table.

Literature Review

Anomaly Detection Techniques

Simple Statistical Methods

The simplest approach to identifying irregularities in data is to flag the data points that deviate from common statistical properties of a distribution, including mean, median, mode, and quantiles. Let’s say the definition of an anomalous data point is one that deviates by a certain standard deviation from the mean. Traversing mean over time-series data isn’t exactly trivial, as it’s not static. You would need a rolling window to compute the average across the data points. Technically, this is called a rolling average or a moving average, and it’s intended to smooth short-term fluctuations and highlight long-term ones. Mathematically, an n-period simple moving average can also be defined as a ”low pass filter.” (A Kalman filter is a more sophisticated version of this metric; you can find a very intuitive explanation of it here.)

Density-Based Anomaly Detection

Density-based anomaly detection is based on the k-nearest neighbors algorithm. Assumption: Normal data points occur around a dense neighborhood and abnormalities are far away. The nearest set of data points are evaluated using a score, which could be Eucledian distance or a similar measure dependent on the type of the data (categorical or numerical). They could be broadly classified into two algorithms: K-nearest neighbor: k-NN is a simple, non-parametric lazy learning technique used to classify data based on similarities in distance metrics such as Eucledian, Manhattan, Minkowski, or Hamming distance. Relative density of data: This is better known as local outlier factor (LOF). This concept is based on a distance metric called reachability distance.

Clustering-Based Anomaly Detection

Clustering is one of the most popular concepts in the domain of unsupervised learning. Assumption: Data points that are similar tend to belong to similar groups or clusters, as determined by their distance from local centroids. K-means is a widely used clustering algorithm. It creates ’k’ similar clusters of data points. Data instances that fall outside of these groups could potentially be marked as anomalies.

Support Vector Machine-Based Anomaly Detection

A support vector machine is another effective technique for detecting anomalies. A SVM is typically associated with supervised learning, but there are extensions (OneClassCVM, for instance) that can be used to identify anomalies as an unsupervised problems (in which training data are not labeled). The algorithm learns a soft boundary in order to cluster the normal data instances using the training set, and then, using the testing instance, it tunes itself to identify the abnormalities that fall outside the learned region. Depending on the use case, the output of an anomaly detector could be numeric scalar values for filtering on domain-specific thresholds or textual labels (such as binary/multi labels).

Methods

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. 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.

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.

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.

Experiment

Data exploration and standardization. See Figure 1.

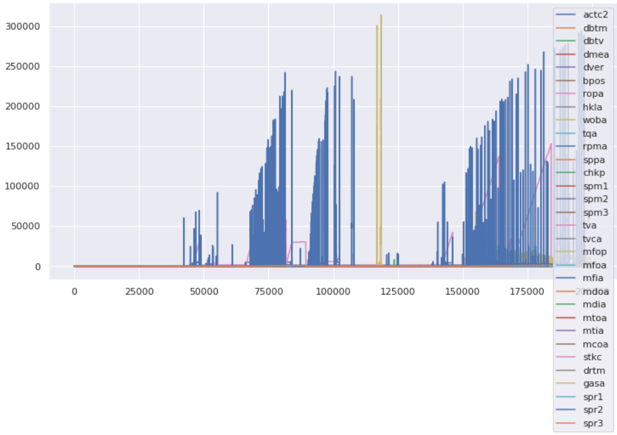

Figure 1: Here is a caption for a figure.

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 is illustrated in the figure below.See Figure 2.

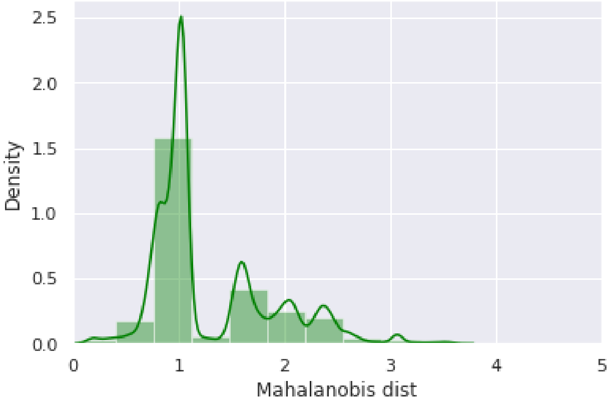

Figure 2: Distribution of Mahalanobis distance for test1

Using the distribution of MD, 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 leading up to the failure, as illustrated in the below figure 3.

In the above figure 4, the green points correspond to the calculated MD, 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 illus-

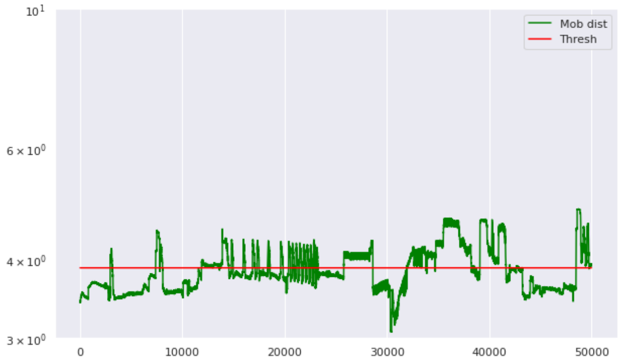

Figure 3: Predicting failure on test1

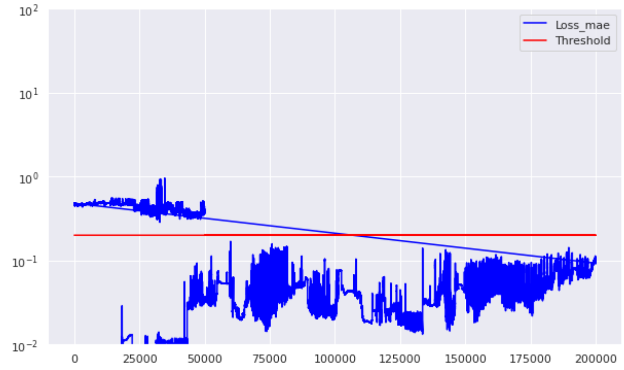

Figure 4: Predicting failure on combined data

trates that the first modeling approach was able to detect the upcoming equipment failure.

We can now go through a similar experiment using the second modeling approach, in order to evaluate which of the methods perform better than the other.See Figure 5

Approach 2: Artificial Neural Network 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 to detect anomalies. The distribution of reconstruction loss (mean absolute error) for the training data is shown in the below figure: 9

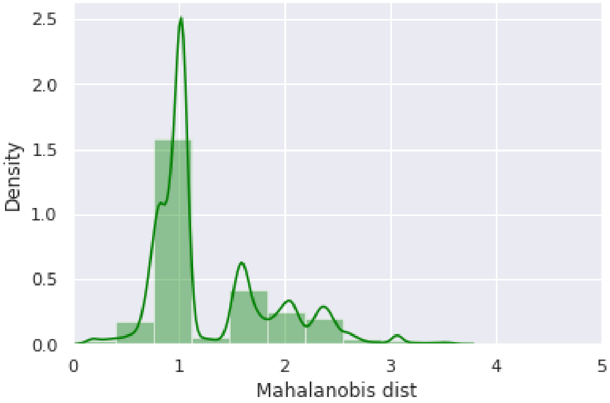

Figure 5: Distribution of Mahalanobis distance for test2

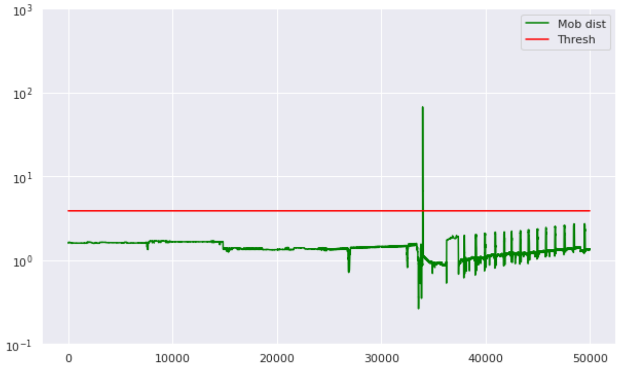

Figure 6: Predicting failure on test2

Using the distribution of the reconstruction loss, 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 abonomal data, as illustrated in the figure below. See Figure 10.

In the above figure, the blue points cor-

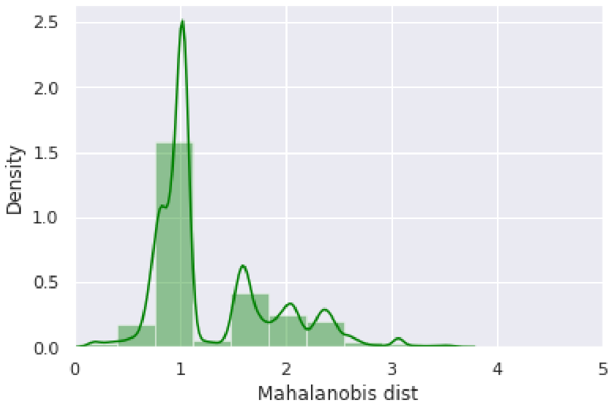

Figure 7: Distribution of Mahalanobis distance for test3

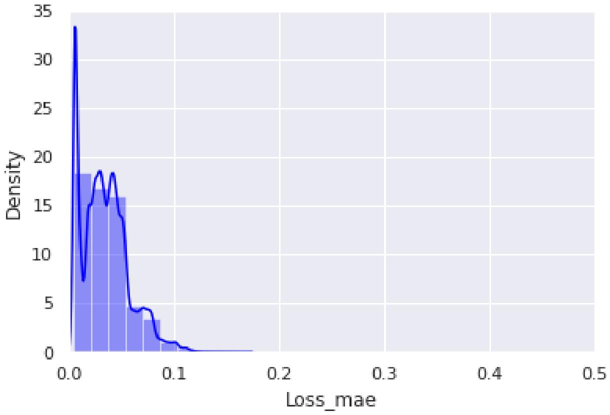

Figure 9: Distribution of reconstruction loss

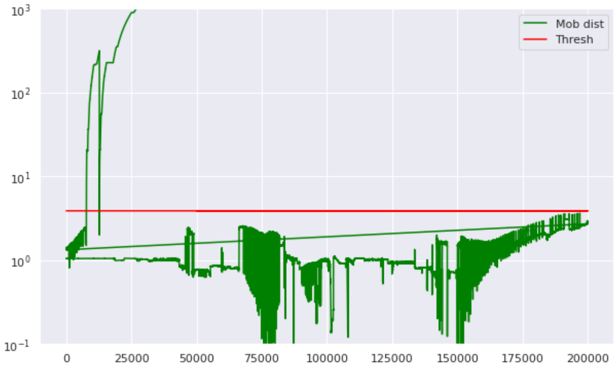

Figure 8: Predicting failure on test3

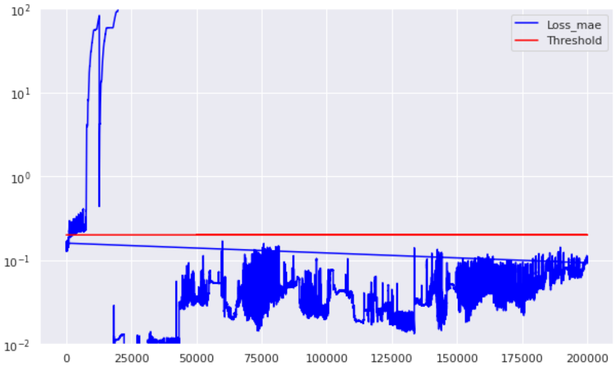

Figure 10: Predicting failure using approach 2

respond to the reconstruction loss, whereas the red line represents the defined threshold value for flagging an anomaly. The 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.

Results

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. 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.

Conclusion

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 problematic but are not alarmed. The first problem not only wastes time and effort but also availability of the equipment. The second problem is more crucial as it leads to real damage with the associated repair cost and lost production.

Both problems result from the same cause: The health of a complex piece of equipment cannot be reliably judged based on the analysis of each measurement on its own (as also illustrated in figure in the above section on anomaly detection). We must rather consider a combination of the various measurements to get a true indication of the situation.

Future Studies

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.

References

Achtert, E., Kriegel, H.-P., Reichert, L., Schubert, E., Wojdanowski, R., Zimek, A. 2010. Visual Evaluation of Outlier Detection Models. In Proc. International Conference on Database Systems for Advanced Applications (DASFAA), Tsukuba, Japan.

Aggarwal, C.C. and Yu, P.S. 2000. Outlier detection for high dimensional data. In Proc. ACM SIGMOD Int. Conf. on Management of Data (SIGMOD), Dallas, TX.

Angiulli, F. and Pizzuti, C. 2002. Fast outlier detection in high dimensional spaces. In Proc. European Conf. on Principles of Knowledge Discovery and Data Mining, Helsinki, Finland.

Arning, A., Agrawal, R., and Raghavan, P. 1996. A linear method for deviation detection in large databases. In Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD), Portland, OR.

Barnett, V. 1978. The study of outliers: purpose and model. Applied Statistics, 27(3), 242–250.

Bay, S.D. and Schwabacher, M. 2003. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD), Washington, DC.