BE.services

Anomaly detection on a punching machine

Software.
Embedded.

Project name: Anomaly Detection
Version: 1.0

CONTENT

1 Introduction

1.1 Motivation and target

The aim is to evaluate the feasibility of an anomaly detection for punching machines based on machine learning algorithms. The drive is considered for this, as it is assumed that abnormal process behavior affects the motor data, such as active current or torque. By training ML algorithms with data from normal operation, a multivariable deviation from this can be recorded in a metric. This metric gives an indication of the normality of the process and, on that basis, control limits can be defined in order to issue warnings in the event of excessive deviations..

1.2 Procedure

For the experiment setup, the machine PLC embeds an OPC UA server, which outputs the following motor operating data:

- active current
- reactive current
- position
- speed
- torque
- temperature

These are read out every second by a client program and stored in a database. In order to keep costs low, both the OPC UA client and the database are implemented on a Raspberry Pi 4b single board computer.

Since the machine should not be operated for so long that real signs of wear occur, the abnormal operation is simulated by means of spring assemblies that influence the lifting movement. Depending on the stiffness of the spring, the influence on the stroke movement varies. The fact that the springs make it difficult to move downwards and make it easier to move upwards is intended to simulate the wear effect.

In a first test run, training data in normal operation and data from abnormal operation are collected, these only being used to evaluate the model and not being required for training. The test run is then carried out one more time, with the trained ML model being integrated and its anomaly value recorded.

2 Results

2.1 First test run

The structure of the first test run is described in Table 1. First, normal operation is simulated at different speeds. Then different spring assemblies are attached as disruptive factors.

Date	Start Time	End Time	Mode	Speed	Disruptive Factor	Phase
27.04.2021	10:15	11:20	Normal	500°/s	None	1
27.04.2021	11:20	12:15	Normal	1000°/s	None	2
27.04.2021	12:15	12:25	Normal	250°/s	None	3
27.04.2021	12:28	12:50	disorder	250°/s	Counterforce 4 spring type 1	4
27.04.2021	12:50	13:04	disorder	500°/s	Counterforce 4 spring type 1	5
27.04.2021	13:04	13:20	disorder	1000°/s	Counterforce 4 spring type 1	6
27.04.2021	13:30	13:47	disorder	250°/s	Counterforce 4 spring type 1	7
27.04.2021	13:47	14:00	disorder	500°/s	Counterforce 4 spring type 1	8
27.04.2021	14:00	14:15	disorder	1000°/s	Counterforce 4 spring type 1	9

Table 1: Structure of the first test run

The drive data recorded during the test run are given in Figure 1. The time is in UTC, so there is a 2h difference to local time. The beginning of each phase of the test run is marked by a vertical line. The green area marks the data that was used as training data. It can be seen that the values vary widely within the phases.

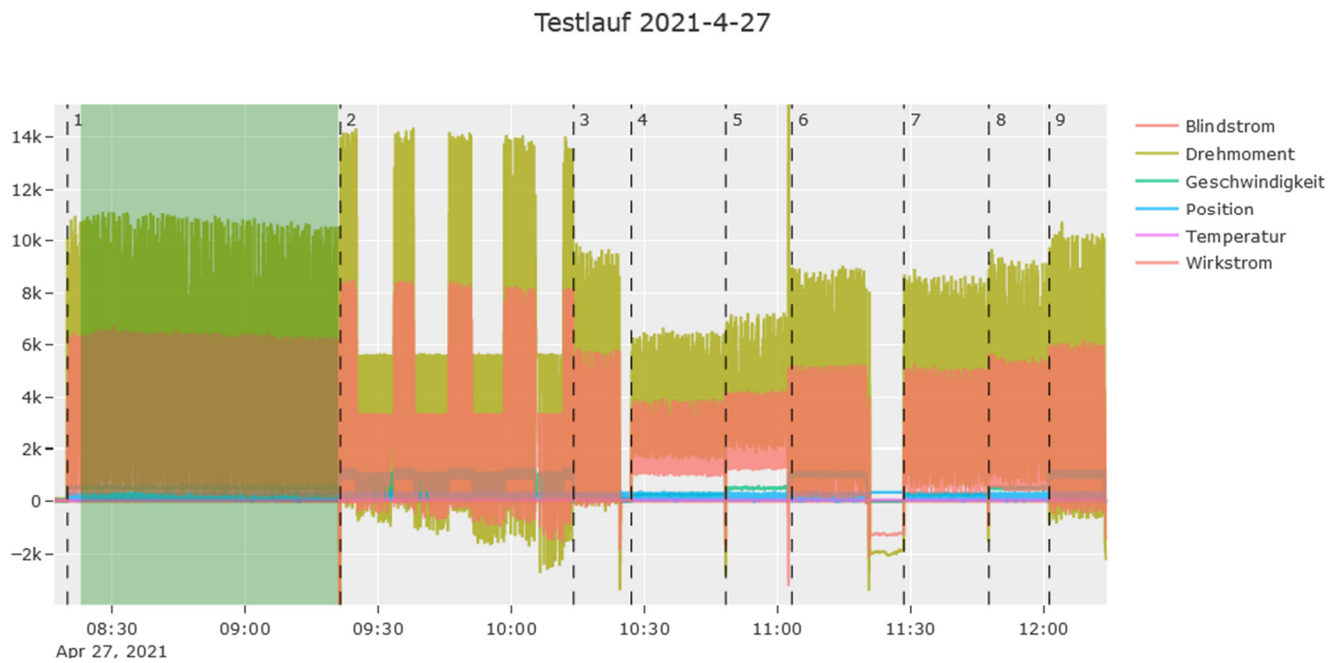

Figure 1: Graph of the raw data from test run 1

In phase 2 in particular, it can be seen that the active current and torque correlate strongly. A correlation matrix with all data points from the first test run is shown in Figure 2. In phase two, a change between two states can be seen, between which the speed fluctuates strongly, see Figure 2.

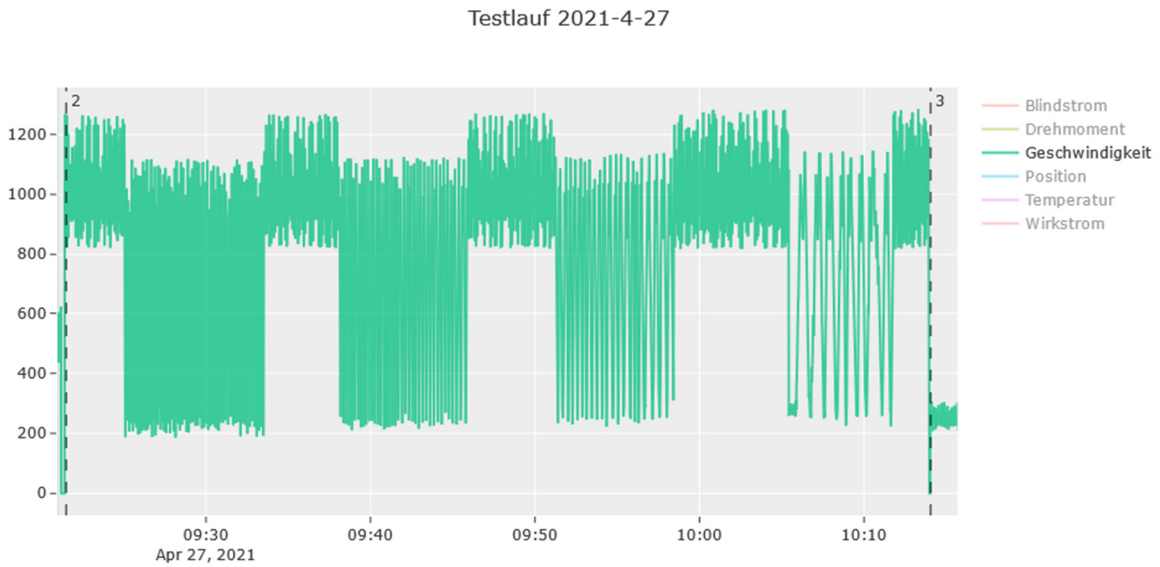

Figure 2: Speed in phase two in detail

The motor seems to fluctuate between two states in which the speed fluctuates within the normal range, but also significantly more. The cause of this behavior could not be precisely identified afterwards.

The correlations between the measured variables can be illustrated using Figure 3. For example, torque and active current are strongly linearly correlated, but it can also be seen that the speed correlates with temperature, active current and torque with coefficients between 0.24 and 0.29.

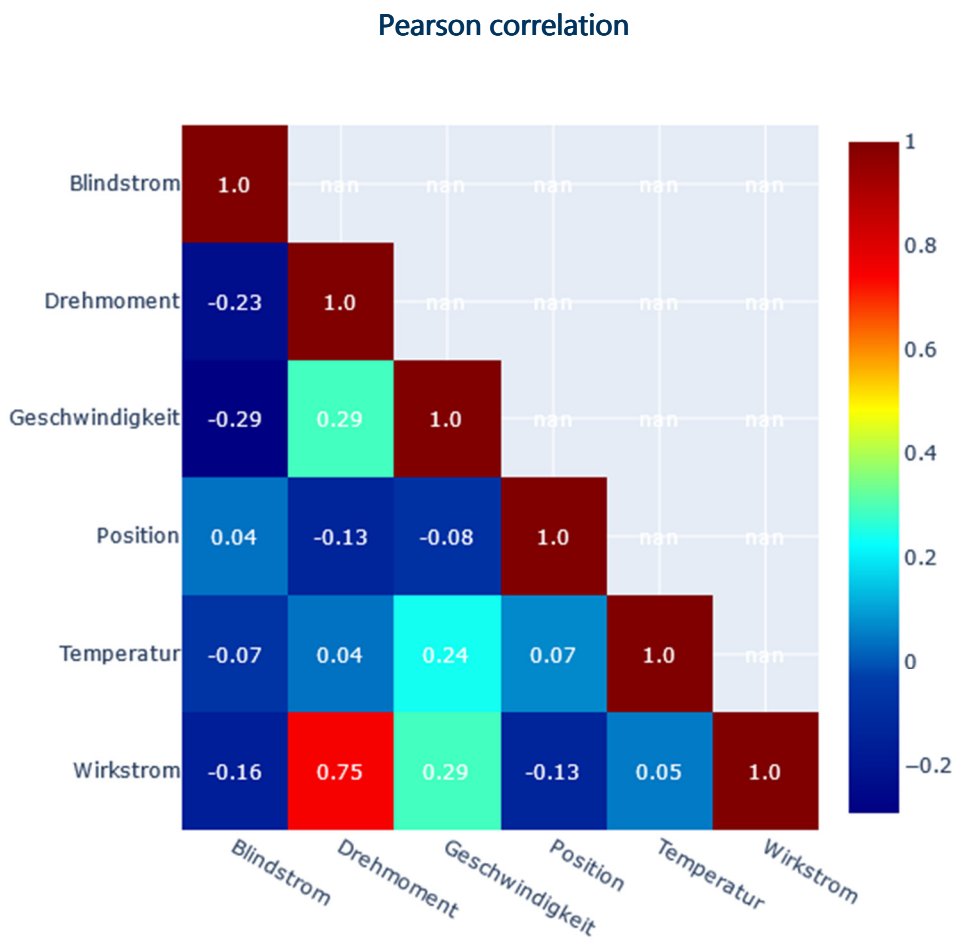

Figure 3: Correlation matrix

First, a starting point is created in which the raw data can be used directly without further processing to train and evaluate ML models. The predictions for the data are generated from a total of four models, all of which were trained with the same data. As an example, the predicted anomaly values of a local outlier factor model are shown in Figure 3, but the result for the other models was comparable or less clear. The more negative the value, the more abnormal this data point is; values around zero mean normal operation. It can be seen that phases one, five and eight differ little, the latter two being anomalies. On the other hand, it can be seen that phases without spring assemblies have much lower values due to their speed. This first attempt has two undesirable effects: on the one hand, the phases that differ only through the use of different spring assemblies at the same speed are not clearly recognized as an anomaly. On the other hand, phases with a deviating speed without interference from spring assemblies are classified as abnormal. These two weak points must be eliminated as far as possible by adapting the model and feature engineering (generating meaningful features).

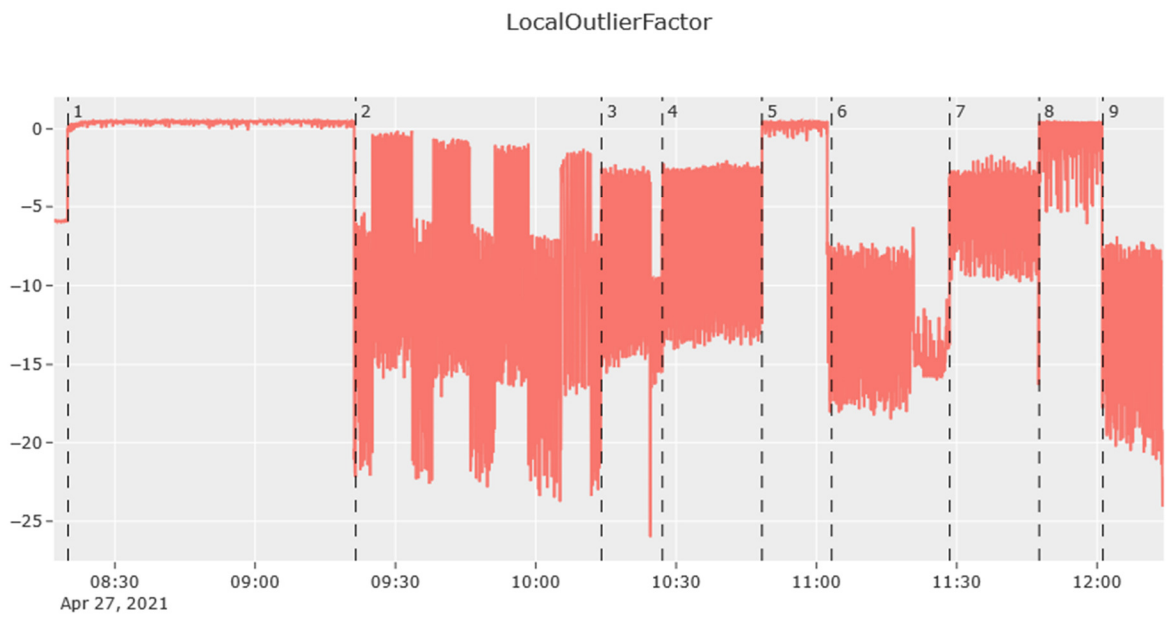

Figure 4: Prediction with raw data

The fact that phases with the same speed but different spring assemblies are difficult to distinguish from the normal state can be illustrated by their statistical distribution. A box plot in Figure 4 shows the distribution of the torque within the individual phases, with phase one in the bottom row describing the normal state.

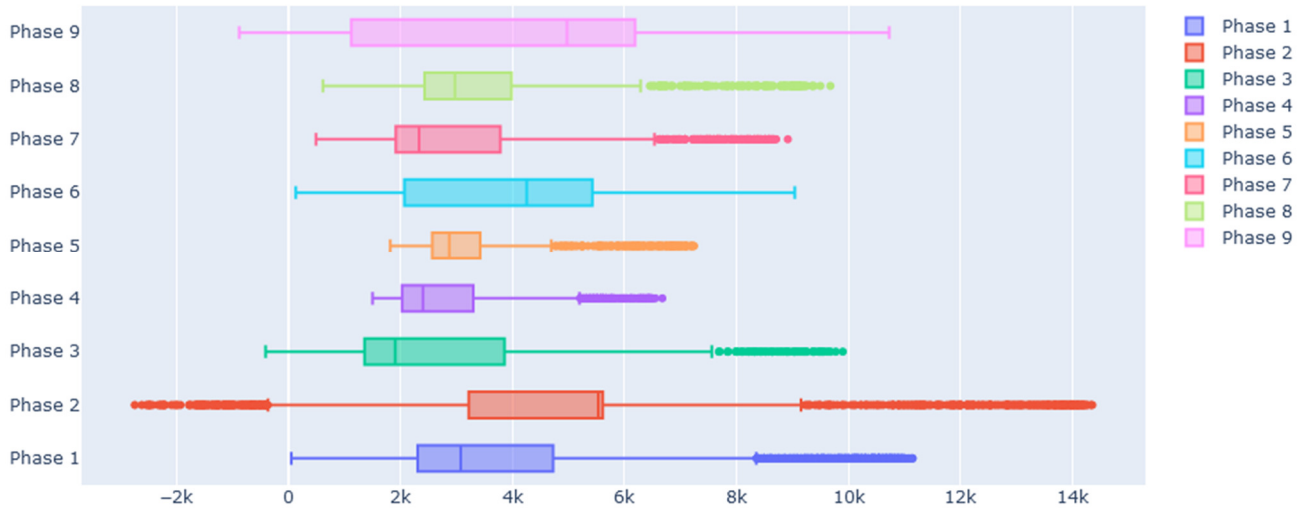

Abbildung 5: Distribution of the torque in the phases

It can be seen that the distributions overlap very strongly, so that it is difficult to separate them on the basis of the torque. This is also the case with the other variables, since, as already mentioned, these correlate in part with a strong linear correlation. In order to generate features that enable a better separation of the phases, a smoothing of the values over a time window of 60 seconds is first evaluated. Various statistical parameters are calculated over this time window: the mean absolute deviation (MAD), the RMS value and the peak-to-peak values. These characteristic values are used in addition to the raw data to retrain the model. The distributions of the peak-to-peak values, for example, allow a clearer separation of the phases, see Figure 5. It is therefore

obvious that an anomaly detection benefits from these additional values.

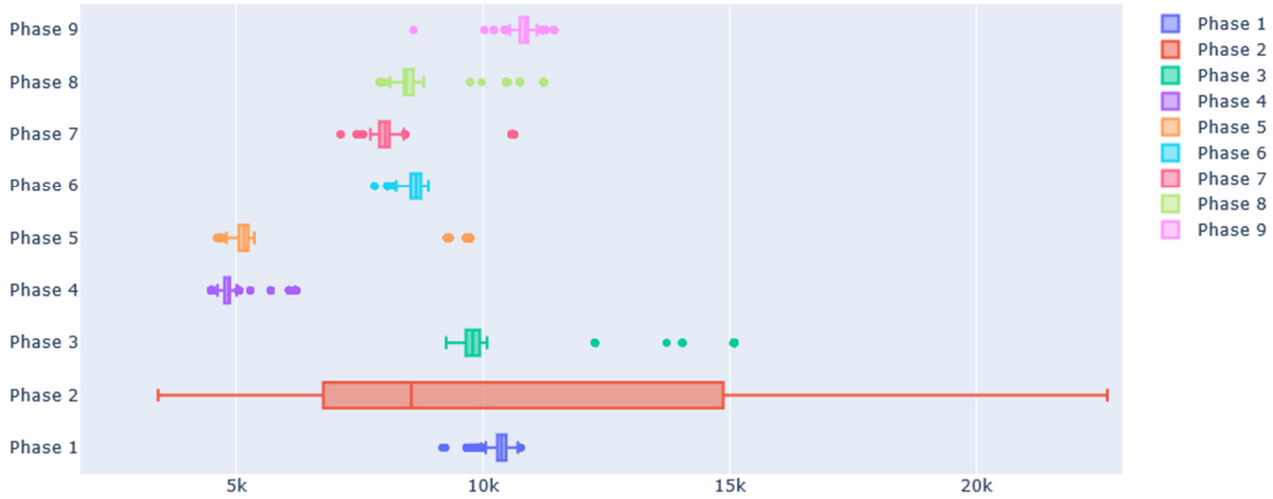

Figure 6: Distribution of the peak-to-peak values of the torque

As a second approach, in addition to the raw values, statistical values are calculated from the measurement data from the last 60 seconds and used to differentiate. The results of the evaluation can be seen in Figure 6. This time, phases five and eight must be clearly distinguished from normal operation. This model is saved and used for the evaluation of the second test run. However, it can also be seen that a change in speed is also accompanied by a large change in the anomaly value.

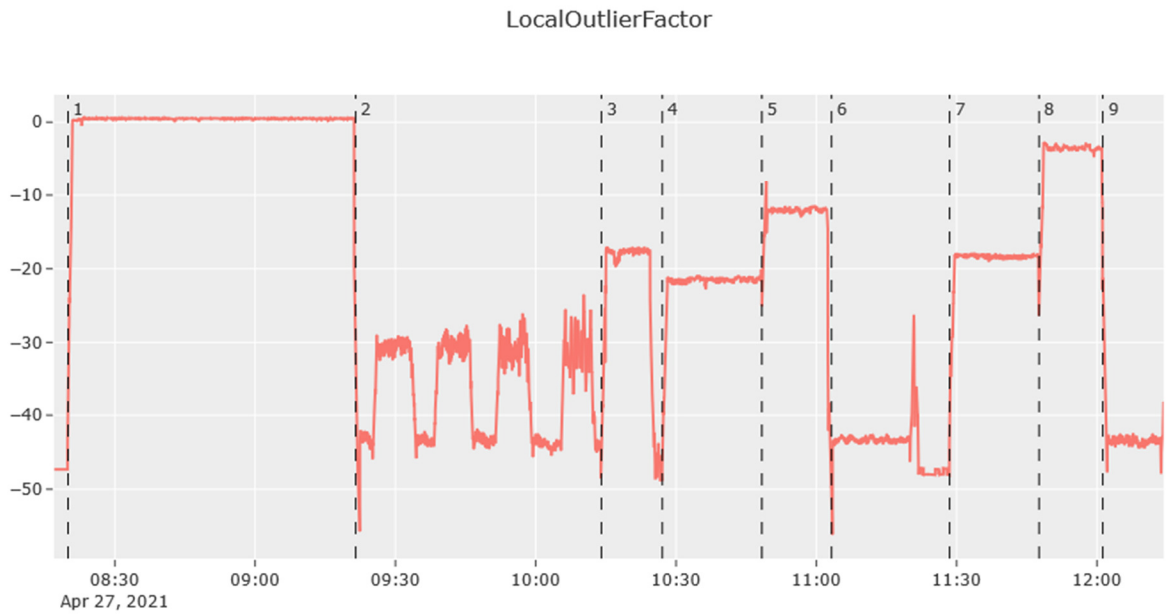

Figure 7: Prediction with additional metrics from sliding windows

2.2 Second test run

The second test run is intended to check whether the anomalies are also detected in live operation. The model calculates the anomaly value for each recorded data point and also saves it in the database. The anomaly value is displayed on a dashboard every five seconds together with a diagram of the drive data, see Figure 7.

Figure 8: Display of the model result on the dashboard

The structure of the second test run is based on that of the first and is given in Table 2.

Date	Start Time	End Time	Mode	Speed	Disorder	Phase
06.05.2021	10:15	10:45	Normal	500°/s	none	1
06.05.2021	10:45	10:56	Normal	500°/s	Adjustable spring package (first 4 small, then slipped and then 2 small)	2
06.05.2021	10:56	11:05	Normal	500°/s	Spring package 4 small	3
06.05.2021	11:09	11:15	Disorder	500°/s	Spring package 2 medium / 2 large	4
06.05.2021	11:17	11:26	Disorder	500°/s	Spring package 3 medium	5
06.05.2021	11:27	11:32	Disorder	500°/s	Spring package 3 medium / 1 large	6
06.05.2021	11:33	11:41	Disorder	500°/s	None	7
06.05.2021	11:42	11:46	Disorder	1000°/s	None	8
06.05.2021	11:47	11:53	Disorder	1000°/s	Spring package 3 medium	9
06.05.2021	11:54	11:59	Disorder	250°/s	None	10
06.05.2021	11:59	12:02	Disorder	250°/s	Spring package 3 medium	11
06.05.2021	12:03	12:05	Normal	250°/s	Adjustable spring package (first 4 small, then slipped and then 2 small)	12
06.05.2021	12:06	12:14	Normal	250°/s	Spring package 4 small	13

Table 2: Structure of the second test run

The results of the anomaly detection are shown in Figure 8. The individual phases are marked and numbered. Phases one and seven are the same settings as normal operation for which the model was trained. Accordingly, the anomaly values are close to zero during these phases. Thanks to the sliding window, a stable value is reached after 60 seconds and during the transition there may be peaks or the value gradually levels off, see phases four and five. The average anomaly values and their standard deviation are also given as a bar chart in Figure 9.

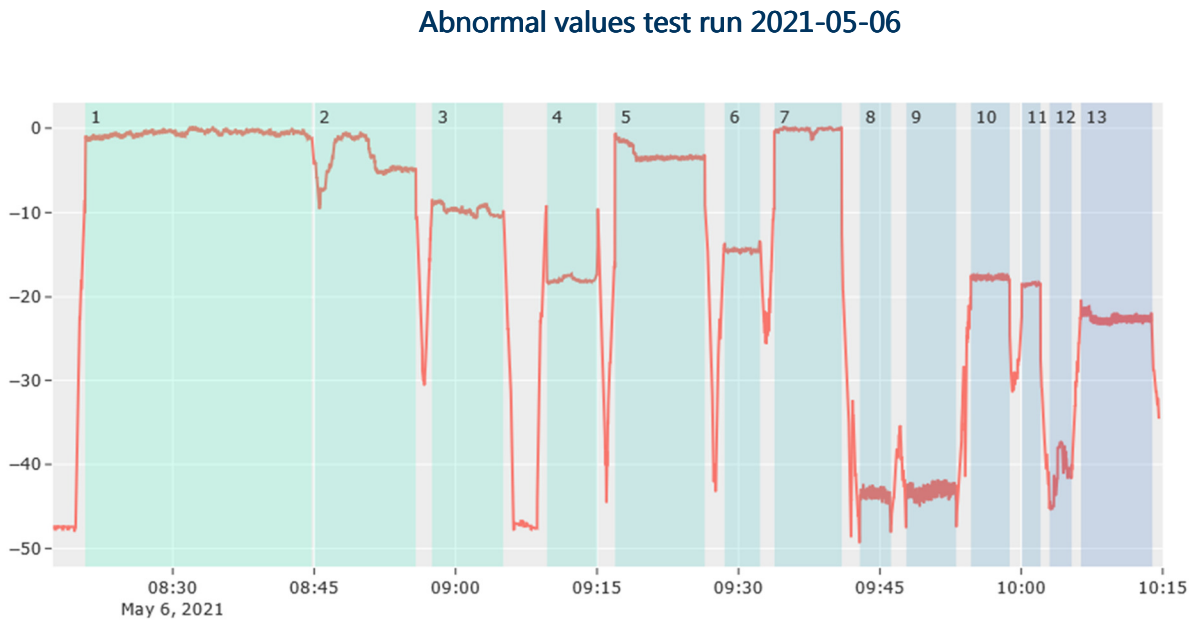

Figure 9: Anomaly values from the second test run

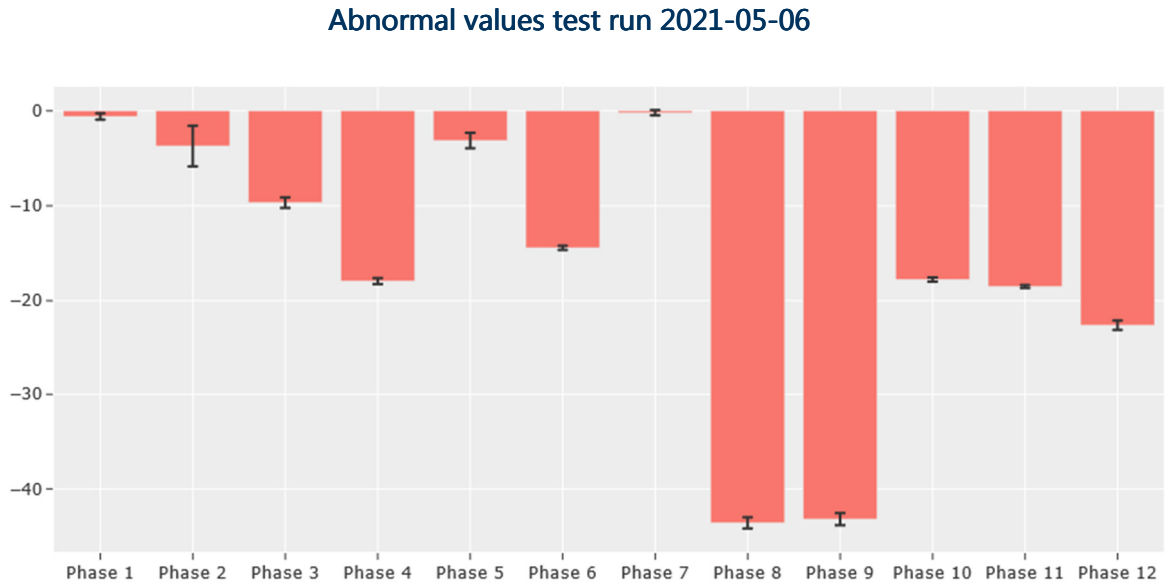

Figure 10: Anomaly values as mean value with standard deviation

It can be seen that all spring packages have a noticeable effect on the anomaly value and are thus detected. For example, phase two is interesting because the value drops noticeably here when the springs were placed, but after they slip back up towards zero until two springs are adjusted again and the value drops to a fixed level. Normal operation in phase seven is also recognized as such, with a small notch when the machine was hampered by muscle power. However, there is still the problem of the dependence on the speed, so phase eight and ten are not influenced by spring assemblies, but are recognized as a deviation due to the speed. The additional attachment of the spring assemblies has no clear influence, however, at 1000 ° / s the anomaly value increases minimally, while it decreases at 250 ° / s.

3 Conclusion

In principle, an anomaly detection can only be implemented on the basis of the motor data, but only under the condition that the target speed only fluctuates within a small range. In this case, each of the anomalies tested could be identified at the predicted value. However, the anomaly value is strongly influenced by a significant change in the setpoint speed and the further attachment of springs then no longer has a clear influence on the anomaly value and can lead to an increase or decrease in the value. Users of the machine set a constant speed so that model and ist implementation allow expected results.

What if users needed indeed to set different values for the speed?

One possible solution would be to retrain the model in the event of large changes in speed. Training times are short (<< 10 seconds), but a sufficient database must be available, in our case ~ 2h normal operation. How realistic this approach is depends on how often such changes would be necessary during ongoing operations.

Another approach would be a more in-depth search for features that enable a separation of the effects of engine speed and anomalies on the anomaly value. The integration of a measured variable that is independent of the motor, e.g. using a force sensor, would probably be the best solution here. It would also be conceivable to include the motor characteristics in order to determine the relationship between speed and active current and thus to isolate this influence from the effects of another source of interference.

The fact that there is a fundamental feasibility is a good result, keeping in mind that the number of available measured variables is relatively small and that these are sometimes strongly correlated.