PHYSICS-GUIDED MACHINE LEARNING FOR SMALL DATA SETS

DISSERTATION ZUR ERLANGUNG DES DOKTORGRADES DER NATURWISSENSCHAFTEN (DR. RER. NAT.)
DER FAKULTÄT PHYSIK
DER UNIVERSITÄT REGENSBURG

vorgelegt von

Andrea Spichtinger aus

Oberviechtach

im Jahr 2021

Promotionsgesuch eingereicht am:
Die Arbeit wurde angeleitet von: Prof. Dr. Elmar Lang

Prüfungsausschuss:

Vorsitzender:	Prof. Dr. Dieter Weiss
1. Gutachter:	Prof. Dr. Elmar Lang
2. Gutachter:	Dr. Stefan Solbrig
Weiterer Prüfer:	Prof. Dr. Klaus Richter

Datum Promotionskolloquium: 16.11.2021

To all the people,
who believed in me and supported me on my journey.

And to the ones,
who will follow.
In particular my godchild, who will be born, when I submit this thesis.

Contents

Abstract

In order to avoid costly machine breakdowns, proactive schedules are often put in place to substitute wear parts regularly. Currently, the contrary approach of Predictive Maintenance is receiving a lot of attention, as it promises needs-based maintenance. Currently, successful implementations are mainly found in highly standardized industries with a vast history of failure data. These conditions are not fulfilled for custom-built machines, namely here bottling machines. This thesis proposes an approach of combining machine learning with physical knowledge to compensate for missing error data. The approach is applied to bottle transport error cases in filling machines.

First, a physical intuition for the machine and the possible error cases is obtained by creating an analytical physical model, avoiding the need for extensive numerical simulations. Second, errors are detected via one-shot semi-supervised anomaly detection, guided by the physical intuition to narrow down suitable algorithms. The one-shot setup involves a particularly short training phase, with only a single healthy sample. The results of the scoring process are anomaly probabilities that are calculated by comparing new samples with the training sample. Samples with high anomaly probabilities continue into the third step, the classification. The anomalous patterns are compared to error sketches, which are drawn by domain experts and enriched by physical knowledge. This approach has so far not been reported in literature.

This thesis demonstrates that this strategy can pave the way to Predictive Maintenance for custom-built machines. It creates reliable results and allows transfer learning to similar machines naturally. It also allows feedback to domain experts in order to improve the machine construction.

1 Challenges of Machine Learning in the Bottling Industry

A long machine failure during production time is one of the worst scenarios for most manufacturing plants in the bottling industry. Water or soft-drink bottlers have very small profit margins on their products, and are thus creating profit by producing around the clock. Beer or milk bottlers risk changes in the product quality during long machine failures. Additionally, a lot of manufacturers hold contracts with supermarkets, which obligate timely delivery. In order to avoid this situation, a proactive maintenance schedule is in place. This implies that all wear parts are changed after specific predefined intervals, for example every 1000 operating hours. These intervals include a safety margin in order to guarantee as little failures as possible between maintenance windows. This concept is - for sure - not very sustainable, as most wear parts are exchanged much more often than actually necessary. Additionally, spare parts and long maintenance windows are costly. As a consequence, new solutions are trying to be found. If it was possible, to detect error cases before they lead to a machine stop, production time could be increased, and concepts like Predictive Maintenance could be set in place. Further on, in case of an unplanned machine failure, giving machine operators a detailed failure description and a troubleshooting guide could enable a faster repair, even by lower qualified staff. In order to find such solutions, the highest expectations are at the moment set on machine learning or “artificial intelligence”.

In other industries, such solutions are already in place. Though, notably, the vast majority of successful implementations have been for highly standardized machine parts like motors or turbines in power generators. The most common implementations are “intelligent” vibration sensors, which are able to detect bearings defects far ahead of the motor breaking [33, 59, 112, 94]. Other successful use cases can be found in industries with very high failure cost, which make highly specialized solutions financially viable.

At first surprisingly, very few of such solutions already find broad usage in the bottling industry. In contrast to the previously mentioned power generation in-

Figure 1.1: Overview over different Krones machines, clockwise from top left: Blower, Filler, Labeller, Palletizer, Brew House, Warehouse, Bottle Washer and Packer.[68]

dustry, a large variety of bottling machines exist. There are about 80 different machine types [68], of which each is highly specialized on its task. To give examples, a blowing machine for PET bottles works completely different from a glass bottle beer filler or a tray packer. Additionally, every machine is further customized for needs of the bottler to suit the intended filling and packing application. Thus, the machine specifications vary from production line to production line. Furthermore, every machine is assembled by hand, thus, even if the machine specifications were similar, exact duplicates don't exist. Thus, transferring a machine learning model from one machine to the next without retraining is not possible, or at least limited to very few exceptions. Instead, transfer learning methods, which allow the transfer of knowledge acquired from one machine to a similar machine, need to be put into focus.

In order to build a reliable machine learning or deep learning model for one machine, a large data set is usually needed, in which a variety of error cases is marked. As Krones (the data provider of this thesis) is the manufacturer but not the owner of the machines, data about "non-lab" error cases are rather rare. Even if data is collected, activities on-site are very difficult to track because not every crash or mechanical modification is documented in a structured accessible way. In fact, even for a number of producers in Germany, failure protocols are still written and shared on paper. Another challenge poses the regular modification of the

machines. Especially during the yearly overhaul, the whole machine is de- and reassembled in order to exchange all wear parts. Thus, the rate of mechanical modifications is in some cases higher than the actual appearance of error cases. As a consequence, a very flexible model is needed, which can be adapted very fast to machine modifications, and does not require any or very little error data.

This thesis aims to take the first steps towards building models, which fulfill those criterion. In order to compensate for the lack of error data, this thesis chooses the approach of combining physical knowledge with machine learning.

The chosen error case will be introduced and motivated in Chapter 2. Additionally, information about the data acquisition is provided, and a new approach for preprocessing the data is proposed. For a better understanding of the physical properties of the machine part and its error cases, Chapter 3 is devoted to the physics of the monitored machine parts. The first goal in this thesis is the reliable and stable detection of error cases and further modifications in the machine. For that, Chapter 4 provides an extensive introduction into semi-supervised anomaly detection and discusses possible downsides and improvements. This physical and algorithmic knowledge is applied on the use case in Chapter 5. In an intensive comparative study, the best anomaly method is searched, which just requires one healthy sample. The second goal of this thesis is providing a classification of the error case. Chapter 6 introduces a completely new approach, which is, to our knowledge, not yet reported in literature: the Physics- and Expert driven Error Sketch Recognition (PEESR). Error patterns are assigned to error descriptions by comparing them to expert error sketches, which are enriched with physical knowledge. The last Chapter 7 summarizes the thesis and gives an outlook for further developments.

2 Use Case and Data Processing

2.1 Analysis of most frequent Machine Errors

Every production line consists of a number of machines (usually ranging between six and fifty), which are connected via conveyors. Every machine operates independently of the others, except for weak coupling by the content of the conveyors.

This thesis will be based on the most essential, and the most widely distributed machine of every production line: the Filler. Additionally, the filler is especially interesting as it is usually in the TOP 3 machines, which cause the most and longest unplanned line stops. Analyzing the fault statistics of a brewery over the course of a year, the filler caused more than 12% of the faults, which is a lot considering the line setup with about 40 machines. One reason for it is that the filler is the slowest and most inflexible machine in the line and thus has a direct influence on the line performance. The filling process is very sensitive and cannot be accelerated or slowed down substantially. In contrast, other machines can compensate for short malfunctions as they are able to run faster than the line speed and have large conveyor buffers before/after the machine.

There are two major construction styles in the filler: The base-handling for glass bottles and the neck-handling for PET bottles. In this thesis, we concentrate on base-handling fillers for the reasons of availability of data and intensive contact to the bottlers. A photo and a schematic image of a base-handling filler can be found in Figure 2.1. The bottles follow a predefined path and are always held by a star or conveyor - depending if they are moved in a circle or a line. The bottles enter the machine on the “infeed conveyor”, are spaced apart by the “infeed worm”, and are then transported via the “infeed star” to the “main filler carousel”. The “main filler carousel” fills the bottles. The filled bottles are further transported by the “outfeed filler carousel” directly to the “capper” or “crowner” wheel (depending on the closure type), which close the bottles. The bottle leaves the filler via the “outfeed starwheel”. It is further transported via conveyors to the next machine. The names “star”, “starwheel” and “carousel” will be used in the following interchangeably. The name “station” refers to the slot of a star in which

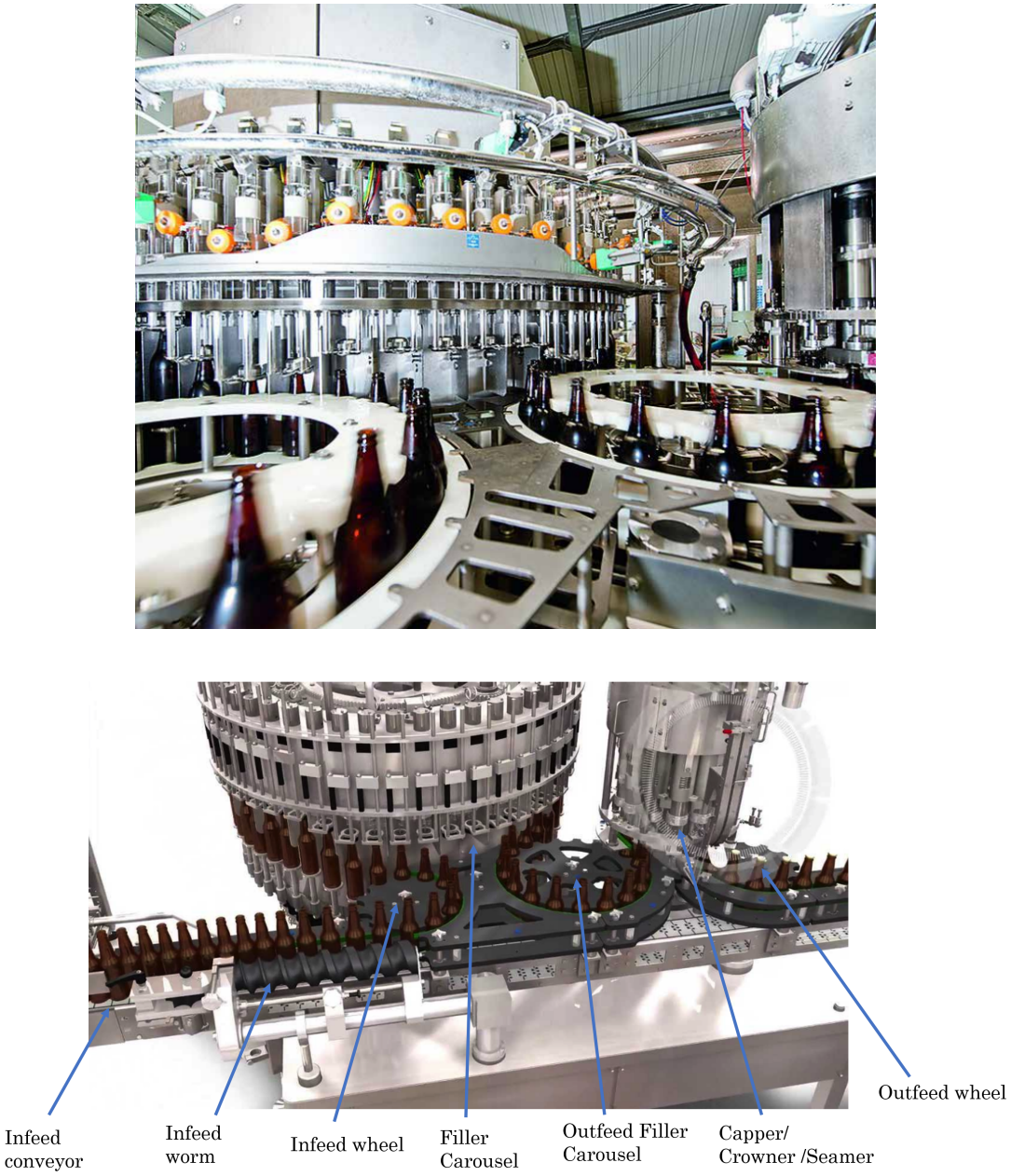


Figure 2.1: Photo [67] and schematic image of a Filler (modified from [68])

a bottle is held. Small stars usually have in the order of 26 stations, the larger filler carousel above 100.

Considering the mechanical properties of the depicted filler, every star has its own servo motor. These motors drive most of the mechanical parts with the exception of the filling and capping process. This makes them the perfect candidates for monitoring most of the machine's movement. They also have the advantage that there are already built-in sensors, which for instance record the motor current. Adding new sensors is usually associated with high costs, as the sensors have to fulfill high food industry standards. Also, the data quality from added sensors inevitably decays over time, as they are not directly needed for production, and thus are not part of the maintenance cycle.

Analyzing the most frequent error cases of the filler itself, the transport mechanism causes 47% of all occurrences, the biggest part by far, and also 42% of the total error time. Over the course of a year, in total 35.8 h of production time are lost in those error cases. In this time, about 1.4 million bottles could have been produced, resulting in lost earnings. Thus, detecting, predicting and classifying those error cases has the potential to save a lot of money.

2.2 Synchronization Error

There is a collection of possible transport error cases, as we will discuss later in Chapter 6. For context, we introduce here the most common error case, which is called "Synchronization Error".

A synchronization error appears when the stations of two stars (e.g. infeed star and filler carousel) are not synchronized perfectly. During the transfer process between the two stars, the bottle is moved from a station of the one star to a station of the next star. In the best case, the both openings line up perfectly and a smooth handover happens. In contrast, if the openings are slightly shifted against each other, a small crash happens. The machine cannot solve a synchronization error on its own, and thus, a crash is continuous happening with every handover of a bottle.

There are several causes of synchronization errors, of which we want to introduce two. First, it can be caused, when a bottle is lying on the infeed conveyor. It crashes into the infeed worm, blocking its rotation. This causes the infeed worm to rotate on its mounting, and thereby loose synchronization with the rest of the machine. An emergency stop will usually be triggered by this event, but this does not fix the resulting synchronization error between infeed worm and infeed star. The error persists even after removing the problematic bottle and restarting the system. Well-trained operators can detect this misalignment, but it is occasion-

ally overlooked. Second, product changes include exchanging some parts of the stars, as these are customized for each bottle type (for example 0.33l versus 0.5l). After product changes, the synchronization should be checked, but is sometimes forgotten due to time pressure.

A long lasting synchronization error has a huge influence on the health of the machine. Extremely high wear of stars, gears and motors increases maintenance costs and the risk of a sudden breakdown, leading to production loss.

2.3 Data Acquisition

Most data used in this thesis originates from built-in sensors in the servo motors, measuring for example the electrical current or angular position. The data cannot be acquired directly from the sensors, but is provided by the machine control. The main job of the machine control is running and controlling the machine in real-time. Thus, although the control accesses the data in a high resolution, it provides the data in a rather low resolution of 100-200 ms for external systems, in order to not endanger the main job. Receiving data in a higher resolution would imply an extensive project with mirroring all communication packages and ensuring that no packages get lost or delayed. In this thesis, we try to find solutions with the limited resolution that the control provides directly.

Some additional information such as high-level machine state (for example production or cleaning) is also provided by the control.

For long term data collection, an on-site edge device connects to the control, collects the data and sends it via a secured channel to encrypted storage in AWS Cloud Services.

2.4 Data Preprocessing: High Resolution via Statistics

For all following analysis, the acquired data is preprocessed in a two-step system.

In the first preprocessing step, all non-production times are removed from the data. Just times in which the machine is running and filling bottles, the so-called “production”, are interesting for detecting transport issues. Additionally, start and stop ramps are discarded, as during acceleration and deceleration a variety of forces are acting that are unrelated to the bottle transport.

The second - and by far more interesting - preprocessing step tackles the huge limiting factor of the sampling resolution. One revolution of an infeed star takes

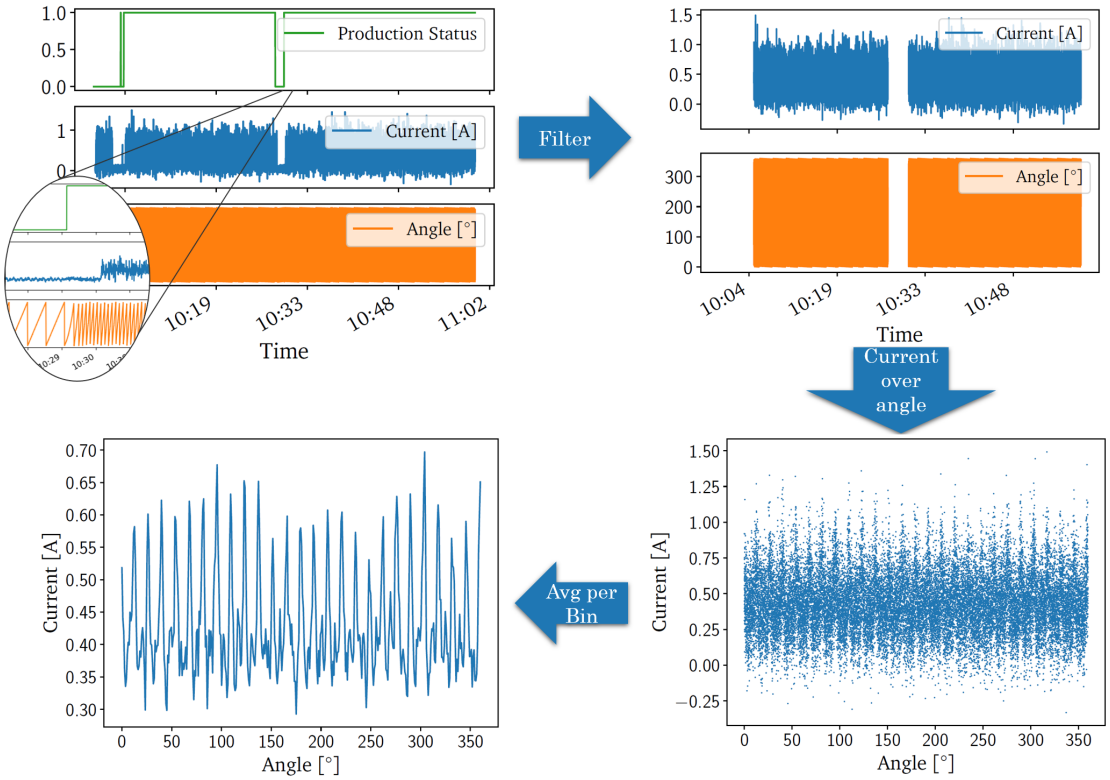


Figure 2.2: Transformation of low-resolution data into a high resolution pattern via preprocessing: Filter by production state with ignoring stop and start ramps (top right). Extract current-angle pattern (bottom right). Create statistical high-frequency pattern by binning and averaging (bottom left).

about 2.7 sec, and within that time 26 bottles are transferred. With acquiring only about 27 measurements in this time interval, detailed analysis of the data is restricted to statistical evaluations.

In order to recover an effective higher sampling rate, we propose following procedure: In addition to the sensor information, the control provides information about the carousel angle of that time point. With collecting the angle-sensor pair over a longer time period, a statistical evaluation can be made. As long as the sampling and revolution period have no common small multiples, a few thousand measurements (spanning 1-2 hours) will cover all possible angles with statistical relevance. The angle is binned into intervals, and for each interval the average sensor value is calculated. The number of bins is chosen in a way that every handover can be resolved with sufficient detail. A resolution of about 15 bins per handover was found to be a sufficient for characterizing the shape. This method can be applied to sensor data, which is not averaged in the control, such as the electrical current. The created curve is surprisingly close to real high-frequency data that is processed in the machine (which, as mentioned above, cannot be routinely measured for this application).

This trick enables transformation of low resolution data into a statistical high-frequency pattern. The preprocessing can extract pattern that appear in the course of the chosen angle, here the carousel round.

2.5 Model Deployment

All algorithmic models in this thesis are executed directly on the on-site edge device. This opens the possibility for inexpensive real-time scoring, as just the scoring result, and not the full data set, is sent into the cloud in real-time. However, this comes with the downside that all models must fulfill some rather strict requirement, as other processes on the edge-device must not be disturbed by the model evaluation. The resource usage has to be evaluated for every model.

Further details about the overall architecture can be found in [124], which was published as a side project of this work.

2.6 Hardware and Software environment

All following calculations are performed on a Laptop with the operating system Microsoft Windows 10 Pro, an Intel(R) Core(TM) i7-6500U CPU processor with 2.50GHz, 2592 MHz, 2 cores, and 4 logical processors. The physical RAM is 32.0 GB.

All numerical calculations are implemented in Python 3.7 in an Anaconda environment (conda version 4.9.1). The calculations and visualizations make use of the python packages Numpy 1.18.5 [48], Pandas 1.2.1 [84, 111], Scipy 1.5.0 [115], Sklearn 0.23.1 [91], Matplotlib 3.2.2 [55] and Seaborn 0.11.0 [120]. Further packages are explicitly cited when used.

3 Physics of Bottle Transfer

To get a better idea about the system, we will consider the basic physical forces, which act on a bottle during the transportation process in a star. We will examine what happens during a normal handover and the effects that a misalignment of the stars can have. The results will be compared to measured data of the machine.

3.1 Conservation measure: Position-based Velocity

One main task of the control is keeping all stars of the machine synchronized at any time. The machine can only produce and transport bottles without crashes when all stars are in full synchronization. This synchronization is also necessary for a safe shutdown of the machine. In order to not endanger the life of any person close to the machine in the event of a defect, the machine has to be able to stop within about two seconds, and that must be possible without damaging the machine, and all bottles within. This can imply that some motors even have to accelerate briefly during the emergency stop in order to keep synchronization with the neighbor stars and to avoid crashes that would destroy the machine. This demonstrates how sensitive the synchronization must be.

To be able to keep synchronization, a control steers all stars. As the control cannot measure the synchronization directly, the stars are initially synchronized manually. The control is then told that the current state is synchronized, and it saves the rotational position of every star:

Claim 1. *At time $t = 0$, the position of each star - and thus the synchronization between the stars - is saved.*

Starting from that time point, the control has a reference point for the motor of every star. The only exception in which this claim is not fulfilled is during an huge-impact machine crash, during which the star shifts independently of the motor reference point, without the knowledge of the control.

In normal production, the control steers each motor by continuously passing a target position. The motor attempts to reach that target position in time, either

by accelerating or slowing down. This also applies for conveyor belts. Thus, the conservation measure is a position-based velocity:

Claim 2. *In a specific time Δt, the stations of every star rotate by the same circumferential target distance Δs.*

For example, in a constant time $\Delta t_{\text{handover}}$, two stars rotate the same distance, which is the distance between two adjacent stations $\Delta s_{\text{stations}}$. In this way, the stations always meet for handovers:

$$\begin{aligned} \Delta s_{\text{stations, Infeed Star}} &= \frac{2\pi r_{\text{Infeed Star}}}{n_{\text{Infeed Star}}} = \frac{2\pi \cdot 0.36\,\text{m}}{26} = 0.087\,\text{m} \\ \Delta s_{\text{stations, Filling carousel}} &= \frac{2\pi r_{\text{Filling carousel}}}{n_{\text{Filling carousel}}} = \frac{2\pi \cdot 2.16\,\text{m}}{156} = 0.087\,\text{m} \\ &\Rightarrow \Delta s_{\text{stations, Infeed Star}} \approx \Delta s_{\text{stations, Filling carousel}} \end{aligned} \tag{3.1}$$

with r being the radius and n the number of stations of the star. Notably, the rotated angle differs due to the different circumferences.

In addition to that very general rule that applies for every machine, the filler is also restricted to very specific production speeds:

Claim 3. *During production, the target filler speed is kept constant at very specific values.*

As already described, bottle filling is a very sensitive process, which cannot be substantially accelerated or slowed down in most cases. Usually, the filler is only able to drive two production speeds per product. As the filler is often the slowest machine in the line, the faster production speed is driven whenever possible. The lower production speed is mainly used for ramp down phases. Thus, for almost all chosen time windows, the target filler speed is constant.

The further physical investigation is based on these three claims.

3.2 Bottle Transport

We start by evaluating the radial transport process of a bottle in a star, ignoring the handovers to different stars for the moment. For that, we examine the forces which act on the bottle and the mechanical parts of the machine.

According to Claim 3, the speed of the star is constant, and because the bottle is fixed to the star the speed of the bottle is also constant. In the rotating frame of reference fixed to the star, both the star and the bottle are stationary. Newton's first law of motion has to be fulfilled: For a non-accelerating object, forces in all

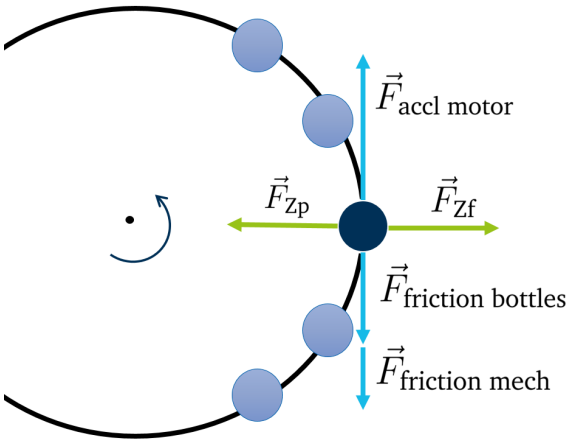


Figure 3.1: Schematic picture of the forces which act on the bottles (blue) while being transported in the star (black). For simplicity, just five bottles are sketched in the star.

directions are balanced. We split up the forces into the forces acting parallel and perpendicular to the tangential transportation direction (see Figure 3.1):

1. Radial forces:
   To keep the bottle balanced in radial direction, the centrifugal and centripetal forces have to be balanced.
   To get a better feeling for the magnitude of the acting forces, we perform a short quantitative estimation for an exemplary star. An outfeed star in a brewery transports filled beer bottles with a mass of $m = 0.85\,\text{kg}$ on a circular path of radius $r = 0.36\,\text{m}$ with a rotation time of about $T = 1.56\,s$:

$$\begin{aligned} F_{\text{Zf}} &= F_{\text{Zp}} \\ &= m\,\omega^2\,r = m\,\left(\frac{2\pi}{T}\right)^2 r \\ &= 0.85\,\text{kg} \cdot \left(\frac{2\pi}{1.56\,\text{s}}\right)^2 \cdot 0.36\,\text{m} \\ &= 4.96\,\text{N}. \end{aligned} \tag{3.2}$$

   Looking at the construction of the machine, the counterpart to the centrifugal forces is implemented via a railing, which does not allow the bottle to leave the star. In comparison to a centrifuge, the railing here is fixed in place and does not follow the rotational movement.

2. Tangential forces:
   Parallel to the transportation direction, friction forces act on the bottle and various mechanical parts. We introduce two main origins:

(a) In the small stars, the glass bottles stand on a metal mounting that does not move. The star bottoms are not actively lubricated, although the bottles carry over some lubrication from the conveyors. In addition to the friction of the bottom, the side of the bottle is pressed by the centrifugal forces against the railing and rubs along it. The railing - also called "wear profile" - is usually constructed from a polymer, which has a very small friction coefficient with the glass bottle.
For a rough estimation, we assume that bottles slide and do not rotate in the transport process. Literature proposes friction coefficients for glass on copper of $\mu_{k_1} = 0.53$ [8], and glass on the polymer PEEK of $\mu_{k_2} = 0.17$ [9].

$$\begin{aligned} F_{\text{friction bottle}} &= \mu_{k_1} F_G + \mu_{k_2} F_{\text{Zp}} \\ &= \mu_{k_1} m\, g + \mu_{k_2} F_{\text{Zp}} \\ &= 0.53 \cdot 0.85\,\text{kg} \cdot 9.81\,\text{m/s}^2 + 0.17 \cdot 4.96\,\text{N} \\ &= 5.26\,\text{N}. \end{aligned} \tag{3.3}$$

(b) All driving mechanical parts are affected by friction. The fast rotating motor, gears and bearing dissipate heat and wear off long term. Even with perfect lubricants, friction is inevitable.
An estimation is complicated due to a long list of influential parameters: material, lubrication, cleanliness of the lubrication, speed, temperature, magnitude of the load and the bearing type [26]. Information about most parameters is not available for this thesis due to internal domain knowledge of Krones and the motor producer. Additionally, the parameters change over the lifetime of the mechanical parts. In literature, the estimation of friction parameters is still an ongoing research topic [90, 39]. A further examination exceeds the scope and goal of this thesis. For simplicity, we will summarize those friction forces as

$$F_{\text{friction mech}} = F_{\text{friction motor}} + F_{\text{friction bearings}} + F_{\text{friction gears}}. \tag{3.4}$$

As noted above, these forces have to be balanced with the force of acceleration provided by the motor $F_{\text{accl motor}}$.

During production (Claim 2 and Claim 3 fulfilled), all mentioned forces are time independent. The only exception are the friction forces in the warming-up phase of the motor, which are not considered in this work. The time independent forces should lead to a stable motor current, and thus no particular pattern in the electrical current is expected due to the bottle transport alone (when ignoring handovers).

3.3 Bottle Handover

As a second step, we examine the physics of a bottle being handed over from one star to the next. We proceed in three steps: At first, we examine general properties of an handover, then we continue with the friction-less analysis, and close with a brief discussion of friction.

3.3.1 General Behavior

During a bottle handover, the bottle is pushed into the neighbor star on a mutual tangential trajectory. The base of the continuing star is always slightly lower than the base of the first star in order to allow an easy transfer. Due to the tangential trajectory, the transfer time can be approximated by taking the diameter of the bottle bottom $2 \cdot r_{\text{bottle}}$ and its speed v_{bottle} into account:

$$\begin{aligned} t_{\text{transfer}} &= \frac{2 \cdot r_{\text{bottle}}}{v_{\text{bottle}}} \\ &= \frac{2 \cdot 0.03065\,\text{m}}{1.44\,\text{m/s}} \\ &= 0.0426\,\text{s}. \end{aligned} \tag{3.5}$$

Putting this into relation with the time of a full revolution $T_{\text{small star}}$, this implies that the star is busy with handovers with each adjacent star about $70\,\%$ of the time:

$$\frac{t_{\text{transfer}} \cdot n_{\text{stations}}}{T_{\text{small star}}} = \frac{0.0426\,\text{s} \cdot 26}{1.58\,\text{s}} = 0.70 \tag{3.6}$$

As a consequence, the effects of an handover can be separated from the next.

Considering the receive and release process, those two processes always overlap. The degree of overlapping depends on the arrangement of the stars to each other.

3.3.2 Bottle Handover without Friction

Next, we examine a simplified handover without any friction. This allows us to develop a better feeling for the essential physics behind it. We assume a perfect handover, which implies that no unexpected crashes happen during the handover. Different aspects during the process of an handover are considered:

1. As sketched in Fig 3.2, the bottle enters and leaves each star or conveyor on a tangential trajectory. In this way, there is no abrupt change in velocity, but rather a smooth transfer between two radial arcs.

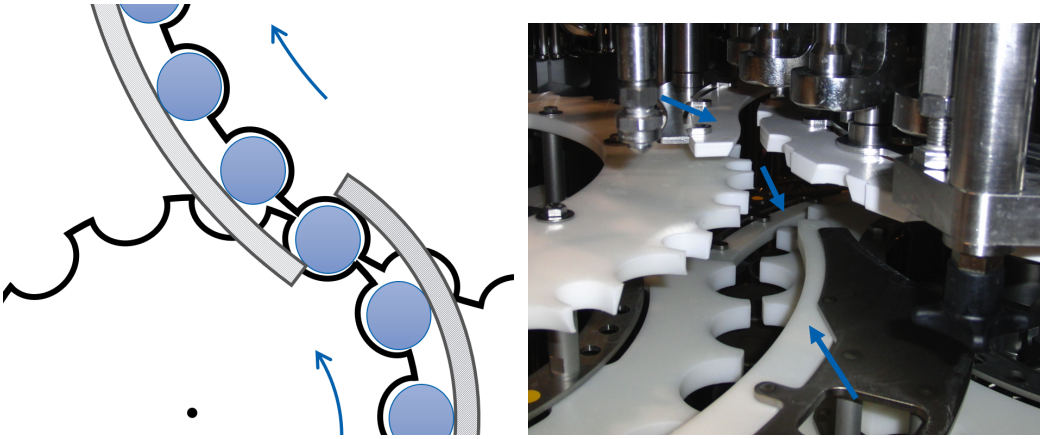


Figure 3.2: left: Schematic handover of a smaller star (bottom) to a bigger star (top). The gray bars symbolize the railings that keep the bottles in the circular paths.
right: Photo of an handover [68] with the railings marked with blue arrows. The first star is holding the bottle in the middle, the other at top and bottom. In this way, there is a position in which the bottle is held by both stars at the same time.

2. According to Claim 2, all stars move with the same circumferential speed. This implies that the bottle also moves with that speed, and that the speed of the bottle does not change in the process of an handover. The angular speed, in contrast, does vary, as the different stars have different radii.

3. The main effect during an handover is the change of path curvature. Similar to the above discussed bottle movement in the star, the railing introduces a centripetal force to the bottle to change its direction. In this way, the value and sign of the angular momentum of the bottle changes and the bottle feels forces perpendicular to the movement, similar to riding a roller coaster. As the interaction of the bottle with the railing is in this consideration frictionless, it has no effect on the speed of the bottle. The railing positioning is assumed to be perfect, which avoids frontal collisions with the bottle.

Summarized, there are no forces acting on the bottle in the direction of the movement. The bottle is completely decoupled from the star and the motor. The whole system can be reduced to a bottle with conserved speed and frictionless railings which define the direction. The motor does not feel any distortion from a bottle handover and thus variations in the motor current are not expected.

3.3.3 Bottle Handover with Friction

In the realistic handover process with friction, a number of processes influence the system. Even with ideal conditions and designs, friction cannot be avoided.

The bottle is never rigidly coupled to the star, which results in an uncontrolled movement of the bottle and small crashes with the star and the railing. Additionally, the synchronization is fixed manually, which makes small misalignment errors unavoidable.

Those effects are very similar to the synchronization error as described in Section 2.2, but with a by far smaller intensity. Thus, we will directly examine the physical properties of the error case, as the model will also cover the normal case of handover with friction.

3.4 Faulty Handover (mathematical description)

3.4.1 Introduction

As already briefly described in Section 2.2, the synchronization error is an error case that happens frequently in production. It describes the state in which two stars don't line up during an handover, which causes small crashes between the two stars at every handover. As a consequence of Claim 1 (the initial position is saved), an imprecise initial synchronization leads to persistent misalignment for the next several hours of production. As the control is not aware of the misalignment, it cannot be fixed automatically.

When an incorrect synchronization is saved, each handover involves a pair of opposing forces acting on the stars. On the one hand, an handover without alignment is not possible as both - the glass bottle and the stars - are rigid objects. Thus, alignment of the two stars is enforced by slowing down or speeding up one of the stars (usually the one with smaller mass). On the other hand, this enforced alignment increases the discrepancy between the target motor position given by the control and its actual motor position. In order to reduce this discrepancy (Claim 2), the motor tries to compensate by speeding up or slowing down.

As the synchronization error cannot be corrected by the machine, the two effects are constantly working against each other during production. The antagonists are active at every transfer of a bottle. A simplified mathematical model illustrates the effects.

3.4.2 System Modeling

The handover is modeled from a small star "A" to a high mass star "B", and we examine the influences on the small star (see Figure 3.3):

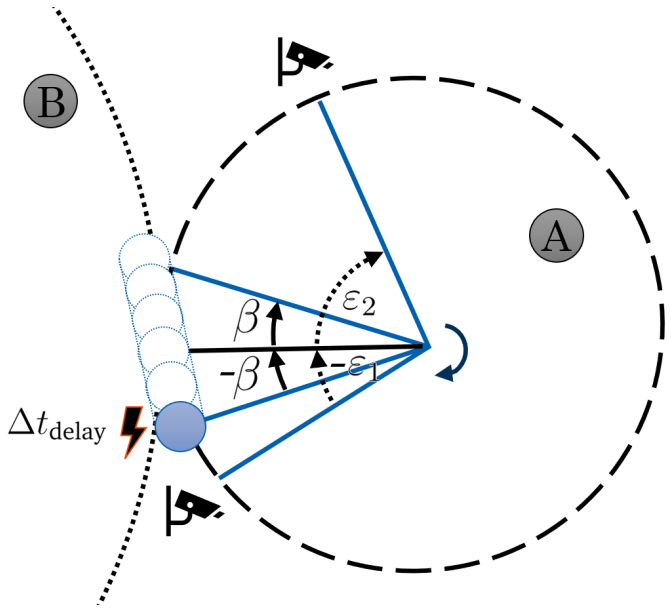


Figure 3.3: Sketch of the model setup for a handover from star "A" to star "B". The handover happens in the interval $(-\beta, +\beta)$. The misalignment is modeled by an abrupt stop of star A, involving a waiting time of Δt_{delay}. The angles $-\varepsilon_1$and ε_2 are checkpoints, at which the position of the carousel is forced to match the target position of the control. For better visualization, all angles are chosen larger than realistically.

- The bottle handover of star "A" happens in the angle area of $(-\beta, +\beta)$, which is fixed by the construction of the star.
- Due to the synchronization problem, a crash happens at the angle $-\beta$ and the star is forced to be stationary for a time delta of Δt_{delay} until the two stars are aligned. This time delta allows to adjust the severity of the synchronization problem. For a non-faulty handover, the time delta is rather small.
- During the handover, the two stars are coupled together, and they drive the same tangential speed. The speed in this phase is set to the fixed production speed v (Claim 3).
- For modeling the constant position-based velocity of Claim 2 of the control, two checkpoints at the angles $-\varepsilon_1$ and ε_2 are introduced. The star reaches those checkpoints at the times determined by the control.

As a side comment, the bottle and the handover process cannot be handled in the calculation as point-like because there is a substantial time period in which the bottle couples the two stars together. The approximation Eq. 3.6 showed that a bottle is being transferred 70% of the time.

3.4.3 Solution by Lagrangian Formalism

The equations of movement for the carousel are formulated by the Lagrange formalism. Transforming the system into cylindrical coordinates, we realize that the movement of the bottle is independent of the radius r and the height z. We can describe the movement simply by the angle α.

The conditions set up by the two antagonists (as described above) are modeled as the four constraints g_1 to g_4:

1. The first checkpoint $\alpha = -\varepsilon_1$ is reached at the time $t = 0$:

$$\begin{aligned} \alpha(t=0) &= -\varepsilon_1 \\ \Rightarrow g_1(\alpha, t) &= (\alpha + \varepsilon_1) \cdot \delta(t) = 0 \end{aligned} \tag{3.7}$$

   with $\delta(t)$ being the Dirac delta function.

2. The second checkpoint $\alpha = \varepsilon_2$ is reached at the scheduled time $t = \frac{\varepsilon_1 + \varepsilon_2}{v}$. The control assumes that the angle interval $[-\varepsilon_1, \varepsilon_2]$ is driven with constant angular speed v:

$$\begin{aligned} \alpha(t = \frac{\varepsilon_2 + \varepsilon_1}{v}) &= \varepsilon_2 \\ \Rightarrow g_2(\alpha, t) = (\alpha - \varepsilon_2) \cdot \delta(t - \frac{\varepsilon_2 + \varepsilon_1}{v}) &= 0. \end{aligned} \tag{3.8}$$

3. As the speed at the handover is taken to be constant, the handover in the angle range of $\alpha = [-\beta, \beta]$ can be modeled by two additional fixed points that are reached at the planned time plus the time delay $\Delta t_{\text{delay}} > 0$ representing the synchronization problem:

$$\begin{aligned} \alpha(t = \frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}) &= -\beta \\ \Rightarrow g_3(\alpha, t) = (\alpha + \beta) \cdot \delta(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}) &= 0 \end{aligned} \tag{3.9}$$

$$\begin{aligned} \alpha(t = \frac{\varepsilon_1 + \beta}{v} + \Delta t_{\text{delay}}) &= \beta \\ \Rightarrow g_4(\alpha, t) = (\alpha - \beta) \cdot \delta(t - \frac{\varepsilon_1 + \beta}{v} - \Delta t_{\text{delay}}) &= 0. \end{aligned} \tag{3.10}$$

   with $\varepsilon_1, \varepsilon_2 > \beta$.

Having formulated the constraints, we will construct the equations of motion for this system. Due to the already fixed velocity in the time window of $\alpha \in (-\beta, \beta)$, the system can be split into two uncoupled parts with $\alpha \in [-\varepsilon_1, -\beta]$ and $\alpha \in [\beta, \varepsilon_2]$. For each of them, we solve the Lagrangian formalism:

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} + \sum_j \lambda_j \frac{\partial g_j}{\partial q_i} = 0. \tag{3.11}$$

$L = T - V$ is the Lagrangian, with T being the kinetic and V being the potential energy. Here, it is $q = \alpha$, $T = \frac{1}{2} m \dot{\alpha}^2$ and $V = mgz \overset{z=0}{=} 0$.

Starting with the angle interval of $\alpha \in [-\varepsilon_1, -\beta]$, this angle window is equivalent to the time window $t \in [0, \frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}]$. The corresponding constraints g_1 and g_3 yield

$$m\,\ddot{\alpha} = \lambda_1\,\delta(t) + \lambda_3\,\delta(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}). \tag{3.12}$$

Integrating over α leads the equation of motion of

$$\dot{\alpha}(t) = \lambda_1\,\theta(t) + \lambda_3\,\theta(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}) + b_1 \tag{3.13}$$

$$\begin{aligned}\alpha(t) =& \lambda_1\,t\,\theta(t) \\ &+ \lambda_3\,(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}})\,\theta(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}) \\ &+ b_1 t + b_2\end{aligned} \tag{3.14}$$

with θ being the Heaviside step function and using $\int dt\,\delta(t-a) = \theta(t-a)$ and $\int dt\,\theta(t-a) = (t-a)\,\theta(t-a)$.

The equation of motion can be solved by inserting Eq. 3.14 into the constraint 1 (Eq. 3.7)

$$\begin{aligned}\Big(\lambda_1\,t\,\theta(t) + \lambda_3\,(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}})\,\theta(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}) \\ + b_1 t + b_2 + \varepsilon_1\Big) \cdot \delta(t) = 0.\end{aligned} \tag{3.15}$$

This equation is fulfilled for all $t \neq 0$ due to the Dirac delta function. Considering the case $t = 0$, the first two terms in the bracket yield zero, as $\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}} > t$. This leaves us with:

$$b_2 + \varepsilon_1 = 0. \tag{3.16}$$

Inserting Eq. 3.14 into the third constraint Eq. 3.9 yields

$$\Big(\lambda_1\, t\, \theta(t) + \lambda_3\,(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}})\,\theta(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}) + b_1 t + b_2 + \beta\Big) \cdot\delta(t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}) = 0. \tag{3.17}$$

Similarly as before, this is fulfilled for all $t \neq \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}$, which leaves us with $t = \frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}$ and Eq. 3.16

$$\lambda_1\,(\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}) + b_1(\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}) - \varepsilon_1 + \beta = 0$$

$$\lambda_1 + b_1 = \frac{\varepsilon_1 - \beta}{\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}}. \tag{3.18}$$

The results of the two constrains Eq. 3.16 and Eq. 3.18 are inserted into the equation of motion Eq. 3.14, taking the angle area of $\alpha \in [-\varepsilon_1, -\beta]$ into account:

$$\alpha(t) = \frac{\varepsilon_1 - \beta}{\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}} t - \varepsilon_1 \tag{3.19}$$

$$\dot{\alpha}(t) = \frac{\varepsilon_1 - \beta}{\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}}. \tag{3.20}$$

The angle interval of $\alpha \in [\beta, \varepsilon_2]$ with the equivalent time window $t \in [\frac{\varepsilon_1 + \beta}{v} + \Delta t_{\text{delay}}, \frac{\varepsilon_1 + \varepsilon_2}{v}]$ works in an equivalent way. The full calculation can be found in Appendix 7. It yields the result

$$\alpha(t) = \frac{\varepsilon_2 - \beta}{\frac{\varepsilon_2 - \beta}{v} - \Delta t_{\text{delay}}}\,(t - \frac{\varepsilon_1 + \beta}{v} - \Delta t_{\text{delay}}) + \beta \tag{3.21}$$

$$\dot{\alpha}(t) = \frac{\varepsilon_2 - \beta}{\frac{\varepsilon_2 - \beta}{v} - \Delta t_{\text{delay}}}. \tag{3.22}$$

Combining these results, the angular velocity and position of the system can be described by:

$$\dot{\alpha}(t) = \begin{cases} \dot{\alpha_1}(t) = \frac{\varepsilon_1 - \beta}{\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}} & t \in [0, \frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}] \\ \dot{\alpha_2}(t) = v & t \in [\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}, \frac{\varepsilon_1 + \beta}{v} + \Delta t_{\text{delay}}] \\ \dot{\alpha_3}(t) = \frac{\varepsilon_2 - \beta}{\frac{\varepsilon_2 - \beta}{v} - \Delta t_{\text{delay}}} & t \in [\frac{\varepsilon_1 + \beta}{v} + \Delta t_{\text{delay}}, \frac{\varepsilon_1 + \varepsilon_2}{v}] \end{cases} \tag{3.23}$$

$$\alpha(t) = \begin{cases} \dot{\alpha_1}(t)\, t - \varepsilon_1 & t \in [0, \frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}] \\ \dot{\alpha_2}(t)\, (t - \frac{\varepsilon_1 - \beta}{v} - \Delta t_{\text{delay}}) - \beta & t \in [\frac{\varepsilon_1 - \beta}{v} + \Delta t_{\text{delay}}, \frac{\varepsilon_1 + \beta}{v} + \Delta t_{\text{delay}}] \\ \dot{\alpha_3}(t)\, (t - \frac{\varepsilon_1 + \beta}{v} - \Delta t_{\text{delay}}) + \beta & t \in [\frac{\varepsilon_1 + \beta}{v} + \Delta t_{\text{delay}}, \frac{\varepsilon_1 + \varepsilon_2}{v}]. \end{cases} \tag{3.24}$$

3.4.4 Comparison Model Results with Reality

The model is a strong simplification of reality. We want to challenge the result of the model in three different aspects: Is the solution physical, does the order of magnitude roughly fit with reality, and does the simulated pattern yield some similarities to real measured error cases?

Physicality

Starting with the physicality, the model is clearly a strong simplification of reality. The choice of the constraints leads to just three driven velocities with discontinuous transitions. The discontinuity at angle $-\varepsilon_1$ does not directly compromise the model, as this position should represent a rather sudden stop of the carousel. All other transition points have to be handled as approximations. In the further steps of this evaluation, all transitions are smoothed with a Gaussian filter in order to obtain a more physical solution.

Compare magnitude with reality

In a next step, we compare how well the calculated velocities represent reality.

For this, we record velocity data at a brewery (as described in Section 2.3) and ask the personnel to measure the corresponding synchronization offset during an error case. Combining this information with constructional details of the machine allows the calculation of the three velocities with Eq. 3.23, which can then be compared with the recorded velocity data.

Carousel The star has a radius of $r_{\text{carousel}} = 360\,\text{mm}$. Averaged over 40 turns, one turn takes $t_{\text{cycle}} = 2.7\,\text{s}$. This leads to an angular velocity of $v = \frac{2\pi}{t_{\text{cycle}}} = 2.3\,\text{rad/s}$. In one turn, $n_{\text{bottles}} = 26$ bottles are transferred.

Handover The two stars are coupled for the time interval it takes to transfer the bottle. In real life, this coupling is not as strong as assumed as the diameters

of glass bottles can vary up to $3.0\,\text{mm}$, and they need to have slack of at least $1.0\,\text{mm}$ on each side within the star. In this way, the strong coupling of the two stars lasts for about the bottle diameter $d_{\text{bottle}} = 61.3\,\text{mm}$ minus the slack of each of the two stars, with up to $d_{\text{slack}} = 2 \cdot 5.0\,\text{mm}$. This distance is used as the direct representation for the angle $\beta = \frac{d_{\text{bottle}} - d_{\text{slack}}}{2 \cdot r_{\text{carousel}}} = 0.07\,\text{rad}$. A handover takes about $\Delta t_{2\beta} = \frac{2\beta}{v} = 60.6\,\text{ms}$.

Misalignment The personnel of the brewery measured a misalignment of $l_{\text{delay}} = 3\,\text{mm}$. This corresponds to a delay time of $\Delta t_{\text{delay}} = \frac{l_{\text{delay}}}{v} = \frac{l_{\text{delay}} \cdot t_{\text{cycle}}}{2\pi\, r_{\text{carousel}}} = 3.5\,\text{ms}$.

Checkpoints We assume that each handover is not influenced by the previous or following handover. This restricts the maximum angle between the two checkpoints to $\varepsilon_1 + \varepsilon_2 \leq \frac{2\pi}{n_{\text{bottles}}}$. Additionally, the checkpoints have to be outside of the handover region with $\varepsilon_1, \varepsilon_2 > \beta$. In order to simulate different strengths of bottle crashes, we introduce the parameter $w_{\text{crash strength}} \in (0, 0.5]$ which represents the weighting of ε_1 and ε_2. The smaller the $w_{\text{crash strength}}$, the more rapid is the crash and - for simplicity - the longer the regeneration phase:

$$\begin{aligned} \varepsilon_1 &= \beta + w_{\text{crash strength}} \cdot (\frac{2\pi}{n_{\text{bottles}}} - 2\beta) \\ \varepsilon_2 &= \beta + (1 - w_{\text{crash strength}}) \cdot (\frac{2\pi}{n_{\text{bottles}}} - 2\beta). \end{aligned} \tag{3.25}$$

In order to get a better feeling for the numbers, the maximum time between the two checkpoints is about $103.0\,\text{ms}$. This leaves about $42.4\,\text{ms}$ for the regularization of the velocity.

Inserting all parameters into Eq. 3.23 results in the velocities depicted in Figure 3.4. The velocity curve is plotted for three different weighting factors $w_{\text{crash strength}}$. During the crash (between $-\varepsilon_1$ and $-\beta$), the velocity in the model is decreased drastically by $22\,\% - 36\,\%$ of the target speed. After slight physical smoothing, the decrease is still about $21\,\% - 25\,\%$. After the handover (between β and ε_2), the machine drives in average about $10 - 13\,\%$ faster to catch up with the target position. As soon as the process is finished for one bottle, it is repeated for the next.

For comparing the result to measured values, the motor speed is recorded, and processed with respect to the angle, as described in Section 2.4. For each angle, a minimum, maximum and average value is calculated. In order to receive

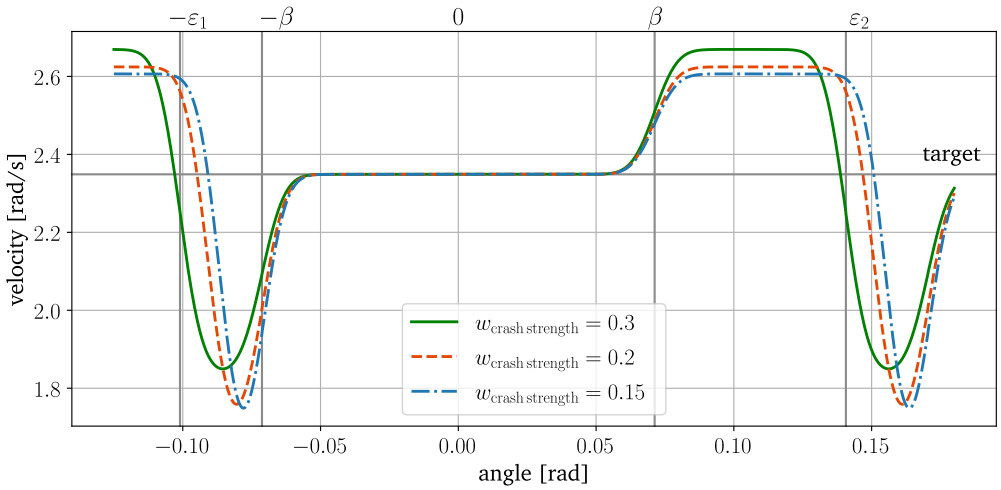


Figure 3.4: Calculated for a real life synchronization problem, the speed of the small star differs by up to $10 - 13\,\%$ from the target speed during the acceleration phase between β and ε_2 depending on the weighing factor $w_{\text{crash strength}}$ (here w).

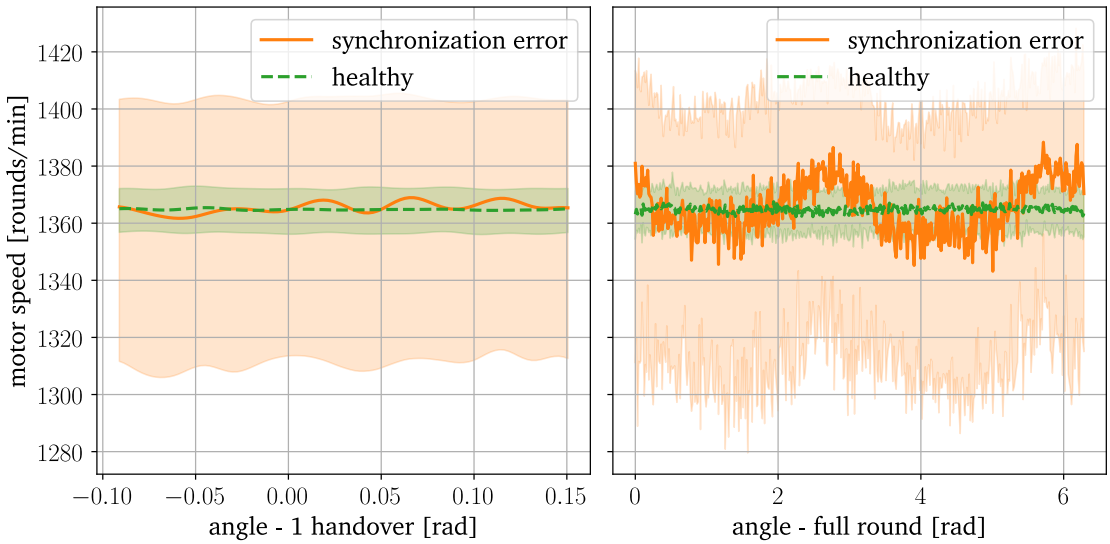


Figure 3.5: left: The measured motor speed relative to the angle during an handover, averaged over all handovers that occurred in 10 curves, each of which covers a 2 hour period of production. This is performed once in an healthy time period (green dashed) and once during a documented synchronization error (orange). Just a very slight variation over the course of the handover can be detected. However, the range between minimum and maximum values rises from about 16 rounds/min to 93 rounds/min in the error case, which is an increase of more than $500\,\%$.
right: The measured motor speed relative to the angle during a full round, averaged over the same 10 curves. A distinct pattern over the course of the full round can be detected.

a representative handover, 10 curves measured during production are averaged. This is carried out for an error case with similar strength as above, and a healthy period. The reported error case discussed above could not be used, as it was promptly detected and fixed before enough data could be collected. As illustrated in Figure 3.5 left, the averaged faulty curve (orange) just shows slight variations in comparison to the averaged healthy (dashed green) curve within one handover. It is expected that the variations are weakened, as an instantaneous speed cannot be measured in a machine, but is calculated as a position difference over time. More importantly, the speed is provided to our data recording device in a lower frequency ($500 - 600$ ms instead of 100 ms). The different acquisition time also results in a substantial lower accuracy, when assigning a speed value to an angle. Thus, every angle bin consists of speed information of a variety of angles, and the pattern is averaged out. It is promising that the average speed of the healthy state and the error case are very similar as this finding supports the Claim 2.

The minimum and maximum value also loose the pattern information, but in contrast, they still show the overall extremes of the curve. As depicted in Figure 3.5 left, the synchronization error has a strong influence on the minimum and maximum values; the amplitude increases from about 16 rounds/min to 93 rounds/min by more than $500\,\%$. In the healthy case, the maximum is $0.5\,\%$ higher than the average speed, and the minimum $0.6\,\%$ lower than the average. This is substantially increased in the error case, with $2.8\,\%$ for the maximum and 4.0% for this minimum. This increase/decrease fits qualitatively very well to the behavior of the model. Quantitatively, the model strongly exceeds the measured values by a factor of about four. This has several reasons. First, as already mentioned, speed can't be measured instantaneously, but is calculated over a time window, and is therefore averaged. This implies that the measured speed variations actually underestimate the true variations. Second, the model exaggerates the variations, as the available acceleration power of a servo motor and the regulating influence of the control are not taken into account. Additionally, the coupling between the bottle and the star is not as strong as assumed.

This result is consistent to the examination of the speed over the course of a full round in Figure 3.5 right. Interestingly, a distinct pattern in the frequency of half a round appears. The origin can not be explained with the model above and can only be conjectured.

As a last examination of the speed magnitude, we examine the effect of different synchronization offsets on the model. Therefore, we fix $w_{\text{crash strength}} = 0.2$ and vary Δt_{delay}. The variation between $\Delta t_{\text{delay}} = 1.2 - 11.8$ ms corresponds to a realistic lag of $l_{\text{delay}} = 1.0 - 10.0$ mm. Figure 3.6 illustrates that, in comparison to

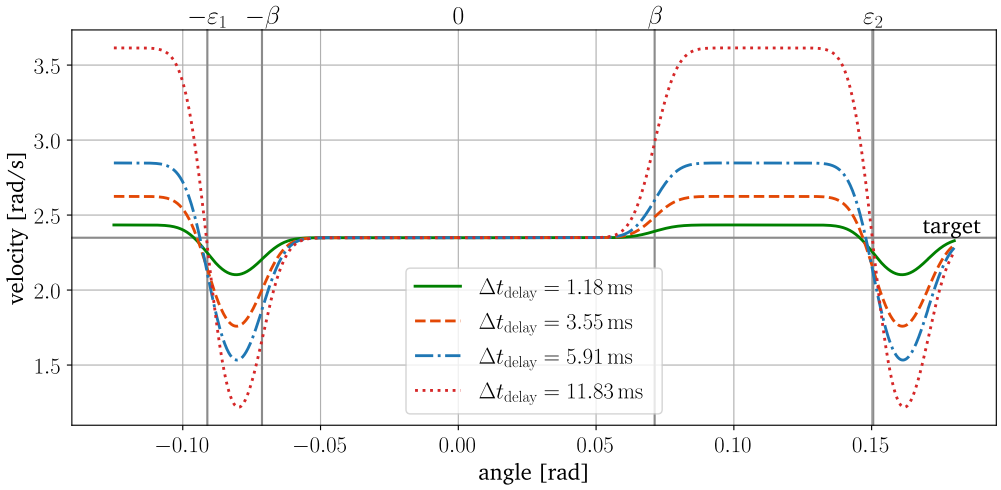


Figure 3.6: Illustration of the influence of different delay times Δt_{delay} on the model. The crash strength parameter is set to $w_{\text{crash strength}} = 0.2$.

the variation of $w_{\text{crash strength}}$, all transition points stay at the same position, but the variations of the speed strongly increase with higher values of Δt_{delay}.

Compare pattern with reality

In a final step, we compare the modeled speed pattern with an electrical current pattern acquired in a brewery. The electrical current is chosen as it is a measured signal which is not further processed in the control, and should thus show an angle-dependent pattern. For that - equivalent to above - we collect data from 10 labeled synchronization error cases, process the current as described in Section 2.4, and average over all handovers and error cases to produce one representative faulty handover.

In order to be able to compare the speed and the electrical current curve, we briefly have to go into detail how the motor steering works, as shown in Figure 3.7. Claim 2 states that the motor always tries to reach a defined time-dependent target position, which is set by the target generator in the main control. In order to reduce the error between the target and the actual position as quickly as possible, a motor control translates the error in an actuating signal, which defines the behavior of the motor, e.g. a current or frequency signal (depending on the type of motor). Thereby, the strength of the actuating signal depends on the size of the error term.

So far, we have modeled the motor control only as boundary conditions (start and end position), and ignored the detailed behavior. This approximation is justifiable, as the influence of the motor control can be neglected during mechanical

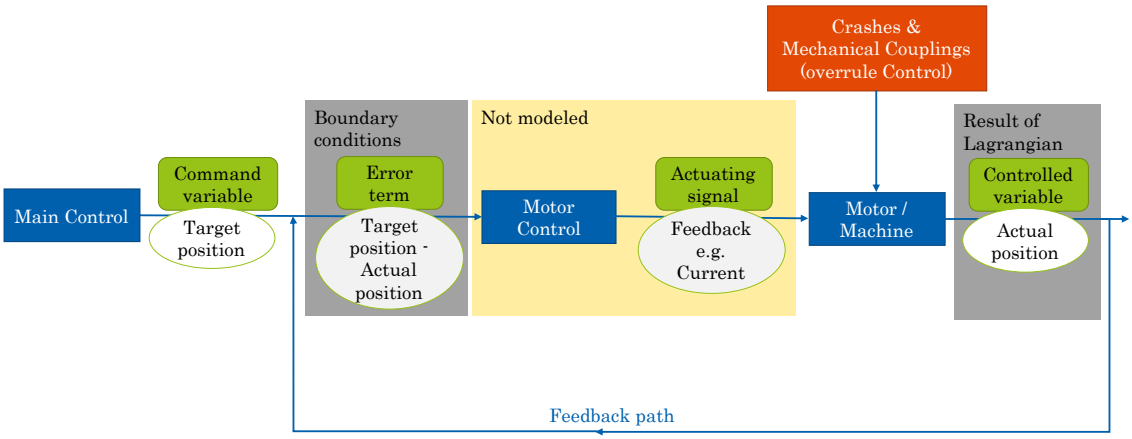


Figure 3.7: Details on the control system of the motor. The main control provides a command variable, which is continuously compared to the controlled variable. Depending on the error and its history, the motor control modifies the strength of the actuating signal. In the case of the servo motors, the position is controlled, and the actuating signal is, for instance, the current or the frequency. External influences like crashes can overrule the behavior of the motor control and modify the actual position strongly. So far, we have modeled the crashes, with the error term taken as a boundary condition.

error cases; during crashes or times of strong mechanical coupling, the physics define the behavior and overrule the motor control. Only in the acceleration phase, the motor control strongly influences the speed in order to minimize the error term. This phase is modeled above in a simplified way, and deviations to the measured signal can be expected.

Even if the actuating signal cannot change the behavior of the machine during crashes and coupling phases, the motor control nevertheless tries to react to the error term at all times. This reaction influences the electrical current. In the following, we will model the resulting actuating signal depending on the error term between target and actual position. The detailed response of the motor to the actuating signal strongly depends on the kind of the motor and thus we determine a generalized 'feedback signal', which should reflect the current in a qualitative way. The motor control is modeled as the popular PID-controller, which tries to minimize the error $e(t)$ by combining a proportional (P), integral (I) and derivative (D) feedback:

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_{t_0}^{t} e(t')\,dt' + K_\mathrm{d} \frac{de(t)}{dt}, \tag{3.26}$$

The three terms are summed with weighting factors K_p, K_i and K_d. The proportional term tries to minimize the present error, the integral term considers the past, and the derivative the future. In servo motors, the controller is usually

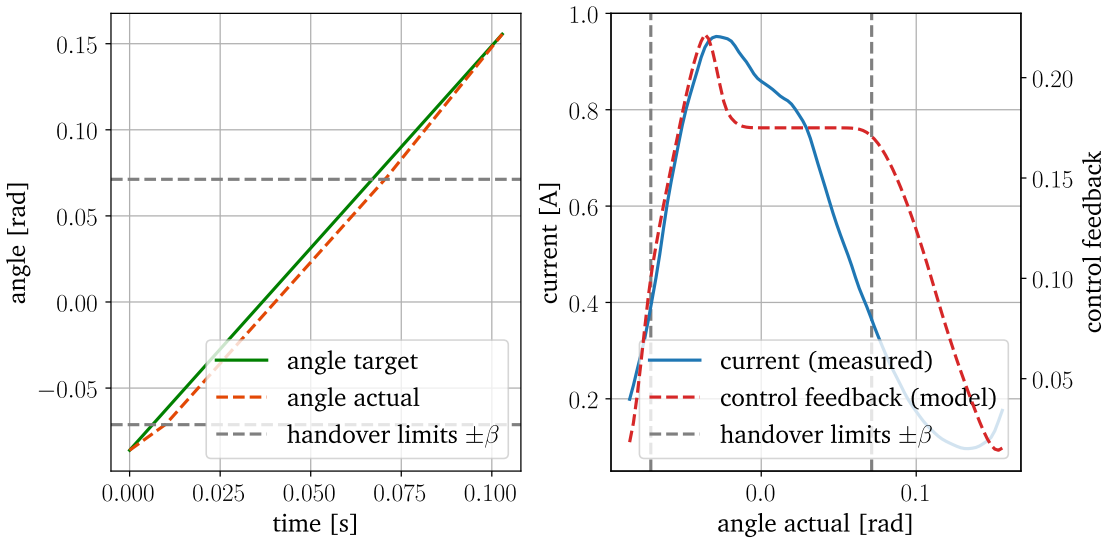


Figure 3.8: The two figures represent a single handover, consisting of a crash, strong coupling with the other star and, in the end, the acceleration phase. With respect to Figure 3.4, the plot shows the angle spectrum from $-\varepsilon_1 = 0.086\,\text{rad}$ to $\varepsilon_2 = 0.155\,\text{rad}$. left: The calculated actual position of the star (dashed orange) and target position (green) during a crash is shown with respect to time. At the start of an handover, a crash causes the target and actual position to diverge. During the handover, the angular difference stays roughly the same. After the handover, the actual angle catches up with the target. right: From the angle difference, an estimated PID-response of the control is shown (dashed red), and compared to the measured electric current pattern (blue). The PID response take a proportional (P), integral (I) and derivative (D) feedback into account. The two patterns show strong similarities, despite the highly simplified model.

constructed in a more complicated way with using a cascade loop of minimizing position, speed and current [116]. The here modeled PID-controller simplifies the system by reducing the number of weighting factors.

For simplicity, we choose the weighting factors such that the three terms contribute in similar strengths. Numbers are not given as the weighting factors have different units and thus are not comparable or meaningful without context. In addition, the integral and derivative parts are restricted to a small time window (about $20\,\%$ of the handover). Even without parameter optimization, similarities between the measured electrical current and the optimization response of the control can be seen in Figure 3.8 right. With optimized weighting factors and time window, even further similarities could be shown. This has not been performed at this point, as the real control settings are not known, and thus the optimizations cannot add any new insights.

Taking the highly simplified model into account, the two curves match surprisingly well. The initial steeper ascending part and slightly slower descending part are reproduced accurately. Additionally, the peak and the shoulder can be found in both curves. The main difference between the two curves is length of the shoulder, and thus the width of the curve. The model assumes a longer strong coupling phase between the two stars than the electrical current pattern shows. Thus, the coupling between the two stars seems to be overestimated by the model.

3.4.5 Potential Model Enhancements

The described model is based on a number of simplifications and assumptions, which restrict the model to qualitative interpretation. Some of the constraints could be improved upon:

- Lagrangian improvement:
  At the moment, the Lagrangian solution is discontinuous in velocity, and Gaussian smoothing is performed afterwards in order to achieve a more physical solution. Adding velocity constraints for all transition points t_i to the Lagrangian would improve the result:

$$\dot{\alpha}(t = t_i) = 0 \quad \text{for } t_i = \{t_{-\varepsilon_1}, t_{-\beta}, t_{\beta}, t_{\varepsilon_2}\}. \tag{3.27}$$

- System improvements:
  - As described above, the motor control is not considered in the model yet. In the final acceleration phase, the motor control could contribute an important factor for the curve shape. Modeling the cascade control of position, velocity and current controller would allow a more detailed

analysis. Choosing the weighting factors similar to the ones in the machine, could improve the output of the model.

  - The current model describes the handover process as continuous strong coupling between the two stars. In an more accurate model, the stars are just joined with a bottle, which transfers torque between the stars. As the bottle moves on the tangential of the star, the applied force on the neighbor star act with an angle at the beginning and the end. This should lead to weaker coupling. As improvement, an angle-dependent coupling strength could be modeled during the handover process.
  - Most stars perform handovers with two stars (receive and release). So far, just the faulty handover was considered, and the other one taken as negligible. In an improved model, the two handovers would be modeled with different intensities. The overlay of the two handovers could lead to interesting effects, like forwarding the error to the next star.

- Quantitative improvements:
  - Numeric simulations are one possibility to achieve quantitative results. Specifics of the motor, gears and control can be taken into account with realistic parameters. In particular, the resulting current can be directly calculated and compared to the measured data.
  - A different option is via improving the data quality. As control-control communication is possible in high resolution, adding a new control for data acquisition can increase the data resolution with minimal increase of the computational load on the production system. Otherwise, a high resolution can be achieved by mirroring the existing communication protocols between motor and gear. However, it has to be guaranteed that no packages are being lost or delayed. Both cases need a powerful computer in order to manage the data recording and storing.

3.5 Summary

In this chapter, we introduced the physical principles of transporting a bottle in a star and its handover to a neighbor star. Three observed claims guided the modeling of an handover. The equations of motion were determined via the Lagrangian formalism, and the resulting speed variations were qualitatively compared to the measured speed in a producing filler in a brewery. For comparing the pattern with the measured electrical motor current, a simplified model of a motor control allowed the translation of the speed pattern into an actuating feedback signal.

Despite the highly simplified models, both comparisons showed undeniable similarities on a qualitative level. Mainly, the assumed strong coupling of the two stars during an handover seems to be exaggerated. Additionally, it was shown in the model that the variations in speed are directly dependent on the crash intensity (modeled by Δt_{delay} and $w_{\text{crash strength}}$). Potential further improvements were discussed.

Thus, one can conclude that small crashes caused by a synchronization error lead to a pattern and behavior similar to the one measured in the electrical motor current. Without a quantitative analysis, the origin of the electrical current pattern cannot be uniquely assigned to the synchronization error. Nevertheless, during a synchronization error, the crashes seem to be a main contributor to the electrical current pattern. In the following, we will use that insight and the better understanding of the handover process for choosing anomaly measures in Chapter 5, and for creating error sketches in Chapter 6.

4 Semi-supervised Anomaly Detection - Methods

The last chapter established a physical understanding for the basic principles of bottle transfer in a filler. A strongly simplified model represented the reality in surprising detail. Nevertheless, in order to build a machine learning model, which detects error cases in an early stage, more detailed error data is needed. This cannot be achieved with a physical model on this approximation level.

An algorithmic approach for detecting error cases without the need of any error data, is offered by anomaly detection. With no or little error data abnormal deviations can be detected. Though, interestingly, anomaly detection is also affected by the No Free Lunch Theorem: "[...] averaged over all optimization problems, without re-sampling, all optimization algorithms perform equally well" [1]. Consequently, there is no algorithm, which fits all fields of application, but basic knowledge about the system and a rough idea of error characteristics is always needed for achieving the best result. In the following, this knowledge will be extracted from the physical model. Therefore, the simplified model of the last chapter should give a good intuition for the system and the error characteristics. This will be used for choosing suitable anomaly algorithms.

In the following, we start with an introduction of anomaly detection in general. Then, we establish a popular architecture of semi-supervised anomaly detection. In the following, a variety of state-of-the-art methods for feature extraction, anomaly algorithms and anomaly evaluation are introduced. We concentrate on time series data, and focus on methods which succeeded in comparable fields. In the end, some selected methods are evaluated, and improvements are proposed. The next chapter applies the introduced methods on the above described use case.

4.1 Definition of Anomaly Detection

Depending on the application, anomaly detection is known under a lot of different names: Outlier Detection, Novelty Detection, Deviation Detection, Exception

Mining or Change Detection [3]. Analogous to the different names, there is also a huge variation of slightly different definitions of anomaly detection. One rather widely accepted was published by Hawkins [49]: "An anomaly is an observation which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism". Esling [29] adds another important feature to the definition: "Anomalies [...] can be a surprising or unexpected behavior which is previously not known. It may or may not be harmful." This implies that anomalies are in the beginning always neutral, the origin of the new behavior can be of "unhealthy" origin or can just be a normal behavior which was not seen in the training data.

In the field of anomaly detection, setups and algorithms can be divided into three groups depending on the available training data [16, 42] (see visualization in Figure 4.1):

Supervised Anomaly Detection In the supervised case, normal and abnormal data is available and marked in the data. The main challenge in this case is the strongly unbalanced data set, as anomalies are rare events. In terms of algorithm, a variety of "standard" supervised machine learning methods can be used, a typical method is Support Vector Machine (SVM) [99]. The algorithms usually result in a classification of the categories "healthy" and "anomaly".

Semi-supervised Anomaly Detection In the semi-supervised case, only data of the category "healthy" is available. The goal is to classify significant deviations from the healthy data, often also named as reference, as anomaly. The data in this thesis belong to this category. An variety of different algorithms will be introduced in detail in the following sections. Depending on the model, the result will be either a classification or an anomaly score, which expresses the probability that an anomaly just occurred.

Unsupervised Anomaly Detection In the unsupervised case, data without any classification is available. Anomalies are declared on the assumption that the majority of the data is healthy, and anomalies are rare. Statistical Nearest neighbor based algorithms (e.g. K-Nearest Neighbors) or cluster based methods (e.g. Cluster-Based Local Outlier Factor (CBLOF)) can be used to detect this kind of anomaly [42]. Due to the missing reference, the result of the algorithm is usually a probability score.

Additional to the different data setups, the anomalies themselves can be classified in different kinds [16, 42]:

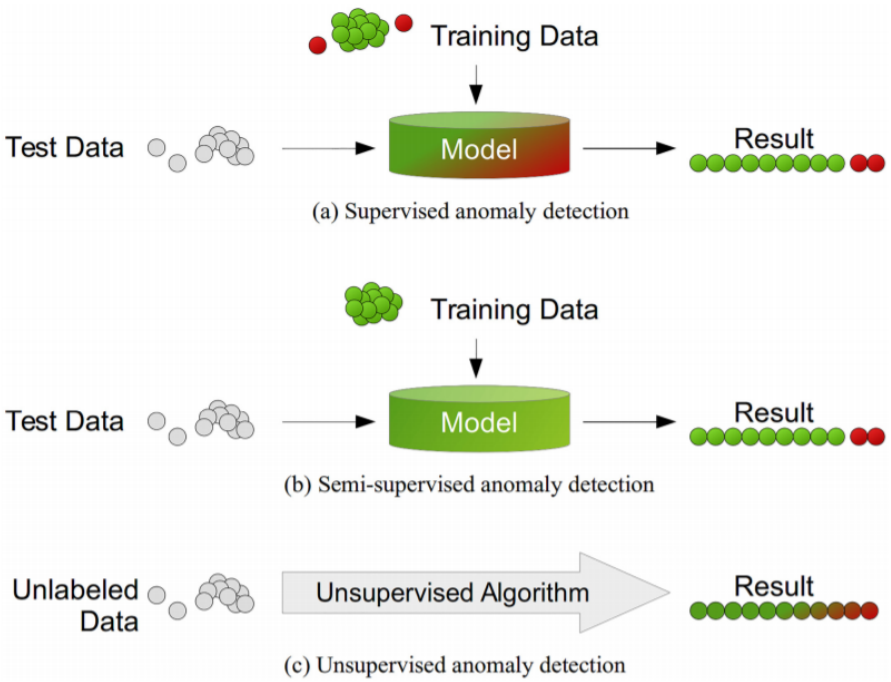


Figure 4.1: Different data setups of anomaly detection[42]

Point Anomaly: A single measurement behaves different than the majority or the healthy state.
For example, a body temperature above 39°C is a clear anomaly as it deviates from the normal body temperature.

Collective Anomaly: Not every data point has to be an anomaly by itself, but the combination or sequence of data points is marking an anomaly.
As example, one transaction of 99 Euro on the bank account is not suspicious. In contrast, having 100 transactions of 99 Euro within a few seconds can be a sign for a fraud.

Contextual Anomaly: Depending on the context the data point is classified as normal or anomaly.
Snow in winter would be normal, whereas in summer it would be definitely an anomaly.

The majority of anomaly detection algorithms is optimized on point anomalies. In order to detect collective and contextual anomalies, the data is often transformed in a way that those anomalies appear as point anomalies. For collective anomalies, point-like features can be extracted via methods like correlation, aggregating and grouping. Hence, patterns are encoded in the extracted features. For contextual

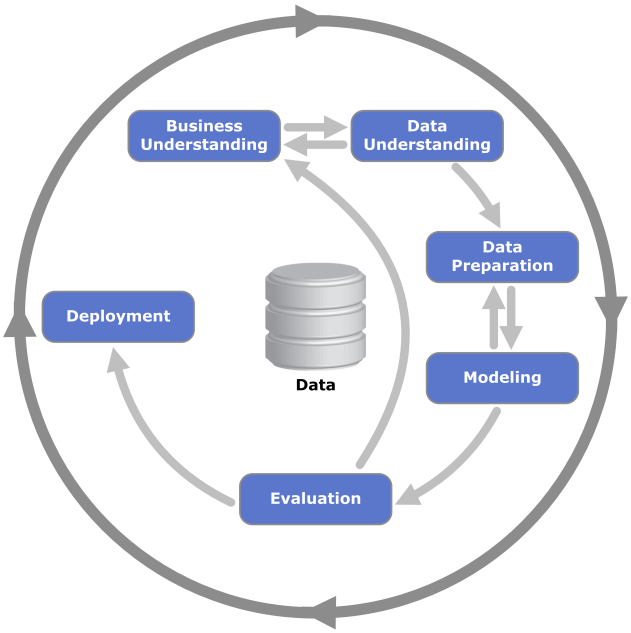


Figure 4.2: Visualization of the steps and iterations in the "CRoss-Industry Standard Process for data mining" (CRISP-DM) ([58] adapted from [57])

anomalies the context (here the season or month) can be added as additional feature.

Thus, it is important to notice, that depending on the transformations, specific anomalies can be detected or not. For every anomaly detection application, it has to be considered, which kind of anomalies should be detected.

In the following, we will concentrate on the semi-supervised anomaly detection. We will introduce the basic architecture, and then, in the following, a selection of transformations and anomaly algorithms. The words "healthy" and "reference" data will be used as synonyms, as will "failure" or "error" data.

4.2 Architecture of Semi-supervised Anomaly Detection

The architecture to set up a semi-supervised anomaly detection is very similar to the normal supervised machine learning setup, but differs widely in some details. As basic architecture the widely accepted "CRoss-Industry Standard Process for data mining" (CRISP-DM) [17, 122] is used (see Figure 4.2) and adapted with specifics for anomaly detection:

1. Business understanding:
   The first essential step is to understand the goal and motivation of the project. In the case of anomaly detection, this also includes collecting expert knowledge about possible characteristics of error cases, and understanding in which cases an alarm leads to added value. Small physical models like in Chapter 3 can provide additional insights.

2. Data understanding:
   Data Understanding includes collecting data and gaining basic knowledge about the data. It is closely linked to Business Understanding, and usually several iterations between Business and Data Understanding are necessary. As result of those two phases, the field of the anomaly detection (supervised, semi-supervised, unsupervised) and the kind of anomalies (point, collective, contextual) should be defined.
   As result of Chapter 2, the goal of this study is to detect collective anomalies in the angle curve with methods of semi-supervised anomaly detection.

3. Data preparation:

   (a) Define reference data:
   Semi-supervised anomaly detection is based on training data without any error cases. This implies that some criteria are needed to select suitable healthy reference data. The creation of this reference data set varies a lot from use case to use case. In some cases, questionable data points can be excluded by data cleaning, while in some others expert knowledge is necessary. Sometimes, the current state is simply defined as reference. This step marks a difference to supervised machine learning.

   (b) Preprocessing:
   Before the data can be passed to an algorithm, some preprocessing steps are needed. This can include a variety of different methods like filtering, normalizing, transforming, resampling or filling missing values.

   (c) Feature extraction:
   Extracting the characteristics of the data before feeding them into a model can enhance the results drastically. There are many different possibilities, depending on the kind of data. The simplest methods are statistical methods like the average or standard deviation. More complex features like non-linear or blind source methods can also be used. Section 4.3 will give an introduction to that field for continuous time

series data and for patterns with fixed length. This step is especially necessary to detect collective anomalies.

4. Modeling:
   Based on the extracted features, a (or several) model is trained. In the case of semi-supervised anomaly detection, the model generally tries to learn a representation of the normal state. It also defines a measure to detect variations from the learned normal state in order to classify them as anomaly. Section 4.4 will give insights into different algorithms.

5. Evaluation:
   The evaluation of different algorithms is one of the trickiest parts of anomaly detection. Two different layers of complexity have to be considered:

   (a) High dependency on use case:
   Depending on the use case, the definition of an anomaly and the tolerance for false alarms or missed anomalies is different. In some cases, it is more important to detect and examine every anomaly, even if some of them are false alarms. This applies, for example, for medical applications. It is preferred to perform additional tests and prove the person actually healthy, rather than missing a person with spreading cancer. In contrast, there are fields that are very sensitive to false alarms, as every examination is very cost intensive. In a wind farm, some maintenance can be only performed by an expert flown to the wind turbine via helicopter, which makes false alarms prohibitively expensive. In this case, missing some anomalies has a smaller impact.
   As a consequence, in some cases, a specific amount of false alarms can be allowed. In other cases, multiple algorithms have to be very certain about the anomaly for a longer time before the alarm is triggered.

   (b) Availability of labels:
   Use cases differ in terms of availability of failure data. In the semi-supervised case, there is often no or very little failure data available. If no error data is available, the anomalies have to be evaluated with respect to use case specific criteria. One possibility is splitting the reference data into a training and test set. The number of anomalies, which are found in the test set, corresponds to the number of false alarms. Another approach is using further unclassified data, followed by a manual evaluation of the detected anomalies by an expert.
   If some failure documentation is available, this information can be used for choosing the best algorithm and tuning of the parameters. However,

failure documentation for anomalies (or in general predictive use cases) often cannot be handled the same way as labels in the supervised machine learning case. The failure documentation just gives information about the timing, when the failure was detected, whereas the goal of anomaly detection is to detect the misbehavior ahead of time. A possible approach to deal with this kind of failure documentation, will be discussed in Section 4.5.

6. Deployment:
   The best performing model (or the collection of models) is deployed and used for scoring. For every evaluation, the data runs through the same preprocessing and feature extraction as the training data. The chosen algorithm determines if the new data is normal or abnormal, and triggers the alarm when suitable.

7. Feedback cycle:
   The user feedback is very essential for anomaly detection. With the information, whether the triggered alarm was a false alarm or if actual alarms were missed, the model can be improved over time. As soon as there are enough high-quality labels for a specific error case, a supervised machine learning model can be trained on the failure data. For most cases, the semi-supervised model will still stay operational, as there are usually new error cases that are not yet contained in the training data.

In the following section, we will concentrate on the state-of-the-art algorithms for feature extraction, semi-supervised anomaly models and anomaly evaluation.

4.3 Feature Extraction

In anomaly detection, the feature extraction is a very important part of the analysis, especially in order to detect collective anomalies. This chapter reviews various state-of-the-art methods. At first, we will concentrate on two groups of features, which were proven successful in the field of machine failures, namely frequency and blind source separation based features. Then, we will introduce an approach, which is based on calculating a huge variety of features. Different packages will be discussed for this extensive feature extraction approach.

All presented features are global features, which imply that they take the whole time series into account with which they were fed. They can be easily transferred to local features, by splitting in the time series in several pieces.

4.3.1 Frequency based Features

Fast Fourier Transform (FFT)

The standard method to transform a signal into the frequency domain is the Fourier transform. In particular, the discrete Fourier transform calculates the discrete frequency spectrum $X(\omega_k)$ with frequencies ω_k from a uniformly sampled signal $x(t)$ of finite length N:

$$X(\omega_k) = \sum_{n=0}^{N-1} x(t_n) e^{-i\omega_k t_n}, \qquad k = 0, 1, 2, \ldots, N-1. \tag{4.1}$$

The most common numerical optimization, the Fast Fourier Transform (FFT), was published by Cooley and Tukey in 1965 [24].

The FFT has been proven successful in detecting machine failures, which show characteristic fault frequencies in the signal. The most prominent examples are bearing faults [43].

Empirical Mode Decomposition (EMD)

The Empirical Mode Decomposition (EMD) [53] - also called Hilbert-Huang transform - is an empirical method, which allows the extraction of the energy-time-frequency information of time series. In comparison to the time-independent Fourier spectrum, the EMD extracts time-dependent amplitudes, split up by different frequency bands (as visualized in Figure 4.3). Those bands are called Intrinsic Mode Functions (IMF). They form a complete and nearly orthogonal basis for the original signal.

A huge advantage of this approach is the possibility to handle signals with changing statistical distributions and frequencies over time - the so-called non-stationary signals. Additionally, the decomposition of non-linear signals is possible with the EMD.

The extraction of the IMFs is performed in a fully empirical process called "sifting":

1. Identify all local maxima and minima of the signal $x(t)$.
2. Determine the upper envelope $e_{max}(t)$ and the lower envelope $e_{min}(t)$ by cubic spline interpolation.
3. Compute the mean of the lower and the higher envelope: $m(t) = (e_{max}(t) + e_{min}(t))/2$.

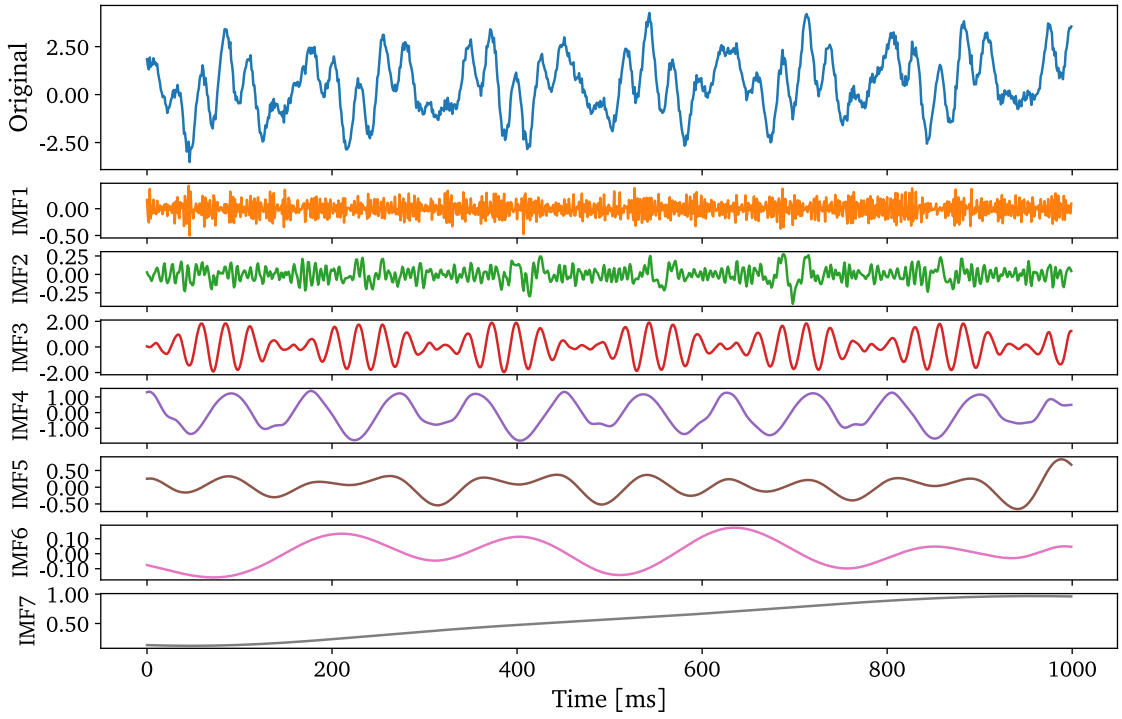


Figure 4.3: Example of the extracted IMFs (bottom rows) of the original signal (top row). The first two IMFs resemble the noise on the signal. IMF3, IMF4 and IMF5 represent the main frequencies with their modulations over time. IMF6 can be ignored due to the small amplitudes. The last IMF extracts the trend of the data. It can be noted, the empirical extraction process sometimes leads to modes with similar frequencies (IMF4 and IMF5), an effect which is called mode mixing.

4. Extract $d(t) = x(t) - m(t)$.

5. Iterate over the steps 1 to 4 with taking the residual $m(t)$. Stop, when $m(t)$ is a trend with just one extreme.

Ideally, all $d(t)$ fulfill the criteria of an IMF: First, the number of extrema and zero crossings must be either equal, or differ at most by one. Second, the envelope should be symmetric in respect to zero. The first IMF is usually extracting the noise of the signal, the last the trend.

The EMD comes with some limitations like end effects, mode mixing or a high computational effort. End effects imply that the EMD is not working properly at the start and end time points. Mode mixing refers to the situation when an IMF has components of different frequencies. In order to tackle those drawbacks, several improvements were developed. We want to mention here the approach of "Ensemble Empirical Mode Decomposition" (EEMD) [125]. The EEMD adds different

variations of white noise to the signal, calculates the EMD for each variation and averages over them. This methods allows to reduce edge effects and mode mixing.

For applying the EMD in the field of machine failures, specific features need to be extracted from the IMFs. The review of Lei [74] presents a large number of successfully proven features. Further features are still being discovered and evaluated, e.g. in [126]. In the following, we introduce three common feature groups:

- The energy content of each IMF E_i and the total energy of all IMFs E_{tot} are very regularly used features [22, 78, 94, 108]. They are defined as the sum of all squared values of the IMF

$$E_i = \sum_{n=1}^{N} x_i(n)^2 \text{ and } E_{tot} = \sum_{i=1}^{I} E_i \tag{4.2}$$

  with N being the length of the IMF and I the number of IMFs. In order to reduce the influence of the edge artifacts, the edges of the IMF are often ignored.

- With the help of the Hilbert transform H [23, 47] the IMFs can be transformed into analytic signals. This implies that a real signal $x(t)$ is being transformed into a complex signal with a non-negative frequency component:

$$x(t) \to x(t) + iH(x(t)) = A(t) \cdot e^{i\phi(t)}. \tag{4.3}$$

  Thereby, the Hilbert transform introduces a -90° phase shift to the original signal. The analytic signal allows to extract a time-dependent frequency $\phi(t)$ and an amplitude $A(t)$, the so-called instantaneous frequency and amplitude. The average and standard deviation of those two measures were proven as successful descriptive features. [22, 80]

- Some error cases show their characteristics in the high-frequency spectrum, mixed into noise. The Teager-Kaiser Energy-tracking operator (TKEO) [83] is a method to decrease the noise in the high-frequency IMFs, and thus extract the characteristic anomalies for a specific time t

$$TKEO_t = x_t^2 - x_{t-1}x_{t+1} \tag{4.4}$$

  with x_t being one of IMFs at time t.
  To detect high-frequency defects in roles or bearing, Tabrizi [107] calculated the TKEO for the first three IMFs, summed up each mode, and normalized the 3d-vector.

According to literature, the EMD features achieve very good results when the machine failures give advance notice in the frequency space, e.g. bearing, gear, or rotor failures.

To be noted, most studies extract the features for one long streak of data. This stands in contrast to the continuous monitoring that is necessary to detect failures ahead of time. In this case, the EMD can show an unfavorable behavior: Every time the EMD extracts the IMFs for the last time period, the IMFs can contain different or shifted frequency bands. When taking the feature values per IMF as a vector, this implies that also the extracted feature vector does not stay consistent over time, but specific frequencies can appear in varying vector entries. We will discuss this in further detail in Section 4.6.1 and provide a solution for it.

4.3.2 Blind Source Separation - Non-negative Matrix Factorization (NMF)

The next group of methods - Blind Source Separation (BSS) - are rather generic methods, whose goal is to separate influences of different origins in the data.

A popular example for visualization is the so-called “cocktail party effect”: Even if a lot of people are talking, our brain is capable to decompose the incoming signal into the underlying origins. This allows a listener in a crowded party to focus on one conversation, or to switch attention to a different one. In a similar way, the goal of the algorithms is to separate the different sources.

Basic Idea

Mathematically, n recordings of the system with the length m are summarized to the matrix $\mathbf{V}_{n\times m}$. Thereby, it is most important that the recordings differ slightly, and show effects of all different origins in different weightings. This can be achieved by measuring at different spots or times. The number of recordings has to be equal or larger than the number of sources k which should be separated. In order to separate the different origins, $\mathbf{V}_{n\times m}$ is being decomposed into the characteristic modes $\mathbf{H}_{k\times m}$ and the weights $\mathbf{W}_{n\times k}$:

$$\mathbf{V}_{n\times m} = \mathbf{W}_{n\times k}\mathbf{H}_{k\times m} \qquad \text{with } k \leq n \tag{4.5}$$

The k modes of $\mathbf{H}_{k\times m}$ are representational modes of the different origins. The weights matrix $\mathbf{W}_{n\times k}$ is the superposition weightings for each recording (see Figure 4.4 for illustration). The heart is the decomposition, for which different algorithms with different properties exist.

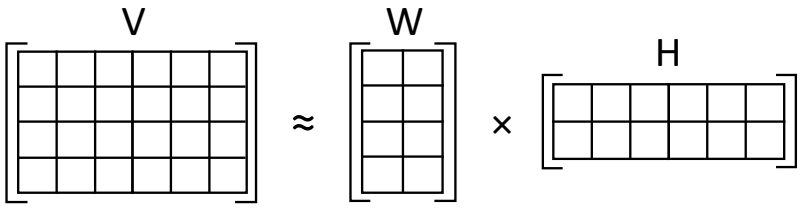


Figure 4.4: The four 6-dimensional measurements in matrix $\mathbf{V}_{4\times6}$ are decomposed into $\mathbf{H}_{2\times6}$ with two 6-dimensional characteristic modes and a weighting matrix $\mathbf{W}_{4\times2}$. (modified from [93])

In order to use BSS as features for semi-supervised anomaly detection, it is assumed that the weighting vector of the modes in the error case differs from ones in the healthy case. Thus, in the training phase, the modes $\mathbf{H}_{k\times m}$ are fixed and the weights of healthy samples are learned. In the scoring phase, the new measurement $\tilde{\mathbf{V}}_{1\times m}$ is decomposed in the fixed modes $\mathbf{H}_{k\times m}$ and the weights $\tilde{\mathbf{W}}_{1\times k}$, which are used as features. For that, $\tilde{\mathbf{V}}_{1\times m}$ is multiplied with the inverse of $(\mathbf{H}_{k\times m})^{-1}$:

$$\tilde{\mathbf{W}}_{1\times k} = \tilde{\mathbf{V}}_{1\times m} \cdot (\mathbf{H}_{k\times m})^{-1} \tag{4.6}$$

As the inversion $(\mathbf{H}_{k\times m})^{-1}$ often does not exist, usually the Moore-Penrose Pseudo-Inverse [92] is used.

Non-negative Matrix Factorization (NMF)

There are a lot of different algorithms grouped within the Blind Source Separation - mainly differing in the conditions the different characteristic modes $\mathbf{H}_{k\times m}$ have to fulfill. Here, we will just briefly introduce the Non-negative Matrix Factorization (NMF). This method stands out for its interpretive modes, which can be very empowering.

The defining characteristic of the NMF is that all three matrices $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ must be non-negative. This banning of negative values ensures physical properties. In a physical environment, each effect is usually non-negative and the total measurement is the positive superposition of all effects. In terms of numerical implementation, Lee and Seung's multiplicative update rule [73] is most popular.

In Literature, NMF is used for a variety of tasks. In mechanical engineering, bearing faults in labs [38] and copper ore crushers [123] are detected as the NMF learned the characteristic failure pattern of the frequency information. This approach is especially valuable when several error patterns start to overlap. Additionally, the NMF succeeded as feature selection method for machine fault diagno-

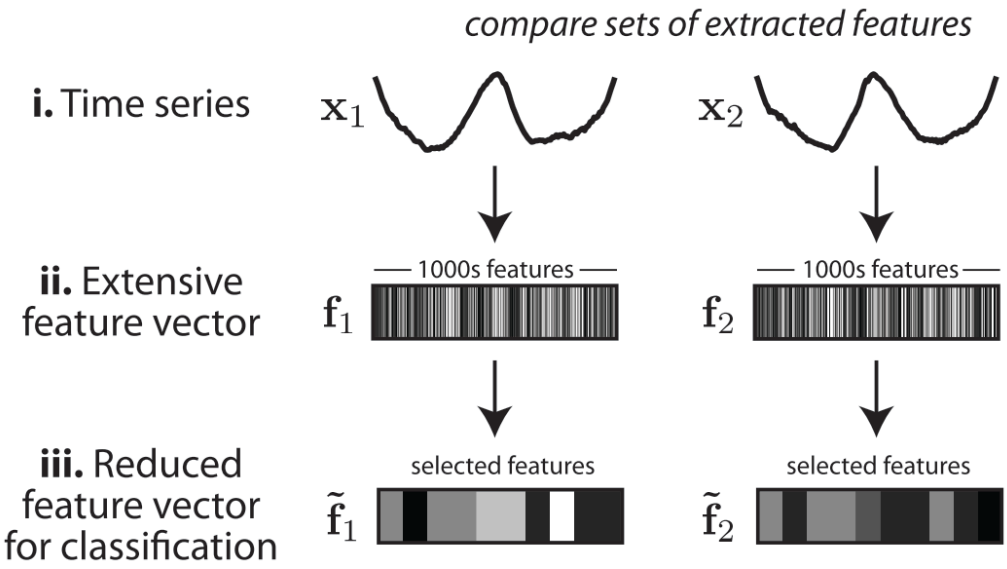


Figure 4.5: The "hctsa" package extracts more than 7700 features of each time series (in this example 1000 features). With the help of labels, the most characteristic features are selected, which allow an efficient differentiation of the two time series x_1 and x_2. (extract from [35])

sis [76]. Interestingly, for facial recognition, the NMF is able to extract different characteristic features like noses or eyes [72, 19].

4.3.3 Massive Feature Extraction

The most prominent work in the field of generic supervised feature based time series analysis comes from Fulcher [36, 35, 34]. Instead of focusing on very specific features, Fulcher's "hctsa" (Highly comparative time series analysis) matlab-toolbox covers a collection of more than 7700 different kinds of features. These include a large variety of statistical measures, linear correlations, stationary measures, entropy measures, and linear and non-linear model parameters. By providing a labeled data set, the most characteristic features can be found and used for classification. With interpreting the extensive feature vector as "DNA" of the data, the package even allows to find time series with similar behavior acquired in a completely different field [37].

This work inspired Lubba [79], who reduced the huge number of measures to a set of 22 features in the C-package "catch22". Those features were chosen in an automated manner by minimizing redundancy between the features. When being tested with over 40 labeled data sets, the package performed just slight worse than the hctsa package, but gained a speed increased of a factor of 600. Interestingly,

most of the features are not easily intuitively interpretative, for example “Change in correlation length after iterative differencing”.

The R package “tsfeatures” [56] has only 16 features that allow a by far easier interpretation, for instance “entropy”. It is slightly outperformed by the “catch22” package [79].

Finally, the package tsfresh [21] closes the gap in terms of programming language. The Python package has 63 feature algorithms implemented, which effectively provide in total 794 features considering different standard configurations. To give an example, the feature “percentile” is implemented and evaluated for four different percentages. This package also allows an automated reduction to the most important features, for labeled data sets.

So far, these toolboxes have been mainly used as basis for supervised classification algorithms. With the help of labels, the huge amount of features can be reduced to the most characteristic set. For classification tasks, this set can then be fed into later algorithm stages, such as a Random Forest [54, 63, 104].

In the case of semi-supervised anomaly detection - to our knowledge - none of the packages was used so far. The main reason is that feature reduction is not possible with unknown characteristics of error cases. Nevertheless, the catch22 features could lay a new foundation for features, and the ability to find time series with similar behavior in completely different fields could allow to determine features without any knowledge of the data.

4.4 Semi-supervised Anomaly Measures

Once features are extracted from a time series, the next step for semi-supervised anomaly detection is the definition of an anomaly measure.

As described in Section 4.2, the models generally try to learn a rich representation of the normal healthy state. A new incoming measurement is compared with the learned healthy state, and depending on the deviation, a 1d anomaly probability is assigned to it. Depending on the probability, the data point is classified as healthy or anomaly.

In terms of literature, there are barely any papers or reviews explicitly about semi-supervised anomaly detection, especially in comparison to the rather huge field of unsupervised anomaly detection [42, 60, 86]. Also survey papers about anomaly detection introduce the different setup types, but barely reference to them later-on [16, 127]. The main reason is that methods optimized for one group can be also used under specific conditions for a different group. To give an example, under the assumption that semi-supervised training sets are never completely clean of anomalies, unsupervised methods can be used by setting the anomaly ra-

Figure 4.6: Overview over different classes of point anomaly methods, and classification by the author if the algorithms are mainly thought or implemented for supervised, semi-supervised or unsupervised applications. (class structure from [16])

tio to a very small number. Similarly, there are few supervised algorithms, which can handle partly labeled data, even if just one class got labeled [6]. Even if the transfer is often possible, it is always important to be aware of the assumption the algorithm is based on, as the result strongly depends on them. For that reason, we will mark all semi-supervised algorithms, which were originally implemented for the unsupervised case - though this information is often not explicitly given in literature.

Anomaly detection is a very active research field with a lot of publications per year. In terms of algorithms, though, the leading survey was written by Chandola in the year 2009 [16]. Since then just few completely new ideas were added. The main research is about applying the existing methods in a variety of new fields [2, 28, 133] and optimizing them for special cases, like huge data sets, a very high number of dimensions or mixed data types [127].

As none of the extreme cases will apply to the data in this study, we will follow the main structure of Chandola [16] as shown in Figure 4.6. The section will present different classes of anomaly algorithms, briefly introduce their premise, and explain one or two algorithms each. We restrict ourselves to point anomalies as the preceding feature extraction already transformed the contextual and collective information into a feature vector. Just two methods for collective anomalies will be mentioned in the end.

4.4.1 Statistical Measures

The most intuitive measures are the statistical measures. The main assumption is that “normal data instances occur in high probability regions of a stochastic model, while anomalies occur in the low probability regions of the stochastic model.” [16]. There are two different classes of stochastic models:

- For the class of parametric techniques, the stochastic model is defined by a parametric distribution, e.g. a Gaussian distribution. Typical methods

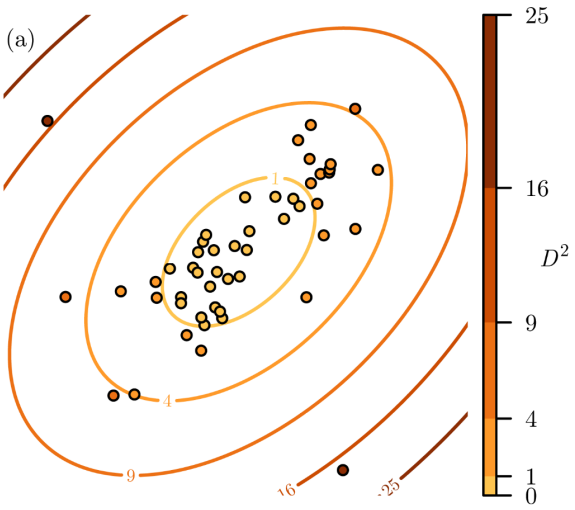


Figure 4.7: Illustration of Mahalanobis Distance. The larger the distance, the more likely the point is an anomaly. (extract from [31])

include the box plot rule [114], Grubb's test [44], student's t-test. [105, 106], $\chi 2$ - statistic [128] or Gaussian Mixture Model.

- For non-parametric techniques, the model structure is generated by histograms or kernel functions. A good overview can be found in [16].

The most important question for the success of the model is always the choice of the stochastic model: Can the normal data be estimated with Gaussian distributions, or does it follow a different distribution pattern?

In the following, we introduce an anomaly measure based on the Mahalanobis distance, which is a rather simple measure for n-dimensional Gaussian distributed data. Despite its simplicity, is shows great results in quite a few studies [14, 133].

The measure calculates the distance to a fixed reference point with taking the different variance in each dimensions into account. The reference point can be fixed in different ways. Popular choices are the average or median of the training data

$$\vec{\mu} = (\mu_1, \mu_2, \mu_3, \ldots, \mu_N)^T \tag{4.7}$$

with μ_i being the average or median of the i -th dimension in respective. The distance measure - the so-called Mahalanobis distance - is defined by the covariance matrix S, which takes the variances in the different dimensions into account:

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}. \tag{4.8}$$

This one-dimensional distance measure D_M can be interpreted as an anomaly probability. The further the point away, the more likely it's an anomaly. With the

help of the healthy reference data, a border between normal and anomaly can be defined.

This anomaly measure is very successful for Gaussian-distributed reference data, but fails for more complex structure. Another drawback is that there are cases, in which the calculation of the covariance matrix is tricky. To give an example, the covariance matrix cannot handle data, which is constant in one dimension in the reference data, though this dimension might be the most characteristic one for error cases.

The Mahalanobis distance is implemented in the python package sklearn [96], there called "Elliptic Envelope". The border is defined by specifying the ratio of anomalies in the training data. This allows the algorithm to be used in an unsupervised and a semi-supervised setup - depending on the chosen ratio.

4.4.2 Classification Measures

The next class of algorithms originates from supervised machine learning algorithms. In a supervised classification use case, every measurement is assigned to a specific class, e.g. cat, dog or parrot. In the case of anomaly detection, there are in principle just two classes: normal or anomaly. This allows the adaption of some supervised machine learning algorithms for anomaly detection. Methods reach from one-class classification algorithms over Bayesian networks to rule based algorithms [16].

In comparison to the statistical measures, most classification algorithms can handle reference data which form complex structures in the n-dimensional space. Some algorithms can also deal with categorical data. The main disadvantage is that no probabilistic score is being calculated. Additionally, there are cases, in which the reference data form several classes, often called multi-class anomaly detection. Though quite a few algorithms were developed for those cases [7, 27], they are tricky to use in a semi-supervised setup as labeling of the different classes within the reference data is needed.

In the following we will give an brief overview over different recent developments in one-class classification algorithms:

One-class Support Vector Machine (OC-SVM) In general the Support Vector Machine (SVM) uses the ability to lift the data points $x_i,\ i \in [1..n]$ into a higher dimensional feature space $\mathbb{F}$. In this space, non-linear decision boundaries can be created simply by hyper planes. In the case of the One-class Support Vector Machine (OC-SVM) according to Tax and Duin [110], the data is transformed into the feature space with the goal of finding circular boundaries that are laid around

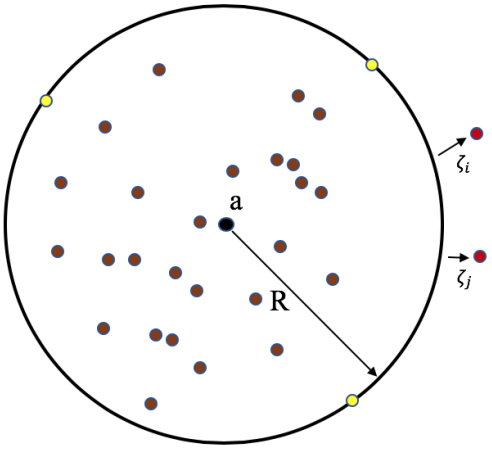


Figure 4.8: The One-class Support Vector Machine lifts data points in a higher dimension, in which the healthy data can be contained by circular boundaries with center a and radius R.[45]

the data. As visualized in Figure 4.8, the radius R between the center a and the outer boundaries is minimized with allowing a soft margin via a slack variable ξ and a penalty parameter C to avoid overfitting:

$$\min_{R,a}\left(R^2 + C\sum_{i=1}^{n}\xi\right) \tag{4.9}$$
$$\text{with constraints} \quad ||x_i - \mathbf{a}|| \leq R^2 + \xi_i, \;\; \xi_i \geq 0 \quad \text{for all } i = 1, ...n$$

According to Bengio [10], one-class SVM reach their limits when being applied to very complex and high dimensional data sets.

One-class Neuronal Network In the case of complex and high dimensional data sets, different approaches, which can be summarized by One-class Neuronal Network (OC-NN), are an ongoing research topic. In some cases, hybrid approaches use Autoencoder for deep feature extraction, which are then fed into OC-SVM [103, 28]. In other cases, OC-NN use a one class SVM as a loss function of the neural network [15, 97].

4.4.3 Clustering Measures

Clustering algorithms are in general unsupervised methods with the focus on finding cluster structures in the data. Under specific assumptions, those methods - or

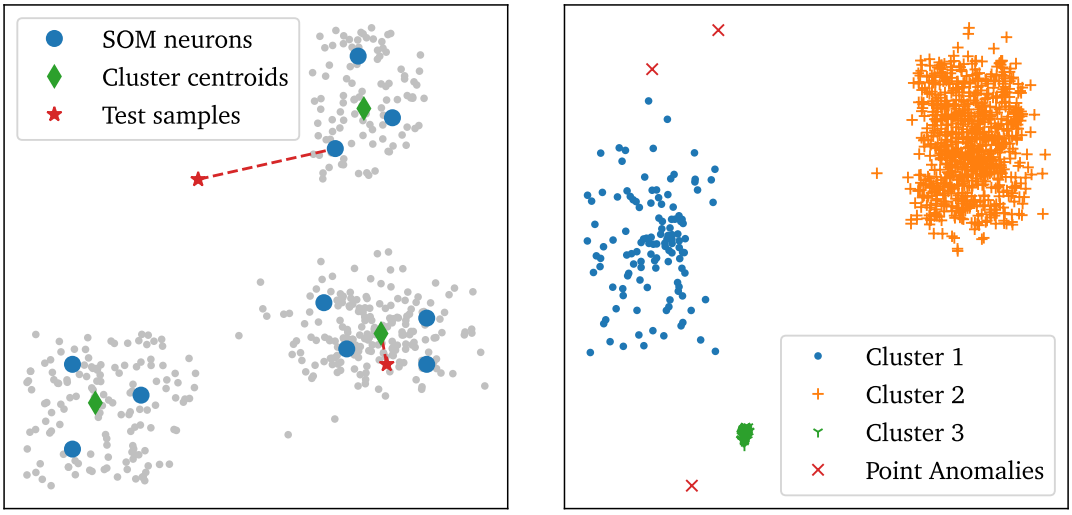


Figure 4.9: left: For a SOM, the distance to the closest cluster centroid or the closest neuron is taken as anomaly measure.
right: In Assumption 3, the large cluster 1 and the small cluster 3 would be defined as anomalies.

variations of them - can be also used for anomaly detection. The algorithms can be categorized in three different kind of assumptions [16]:

- *Assumption 1: Normal data instances belong to a cluster, while anomalies do not belong to any cluster.*
  Methods following this assumption are for example DBSCAN [30], OPTICS [5] or the FindOut algorithm [130]. Those methods are often computationally not optimized for anomaly detection. Additionally, their main focus is on unsupervised anomaly detection, as the main goal is to find anomalies in the data.

- *Assumption 2: Normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid.*
  One of the most popular and successful methods in this category are Self-Organizing Maps (SOM) [65]. The positions of the neurons are trained with the reference data. As illustrated in Figure 4.9 (left), for a new data point, the distance to the nearest neuron or the closest cluster centroid is taken as anomaly measure. This method works especially good in the semi-supervised setup.

- *Assumption 3: Normal data instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters.*

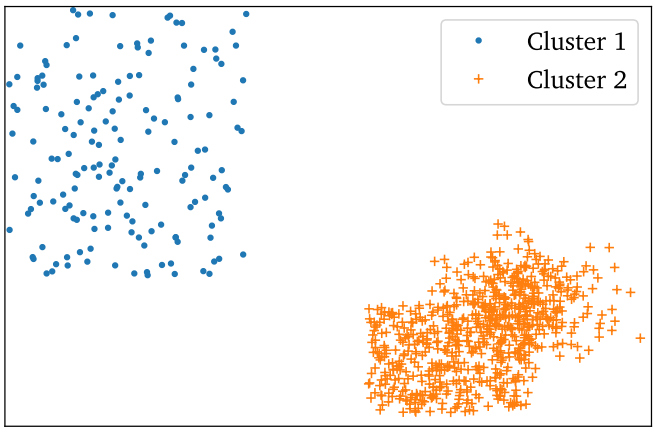


Figure 4.10: This image visualizes the difference between global and local nearest neighbor techniques: For global measure, the sparse cluster 1 will be classified as anomalies, whereas for local measures not.

The Cluster-Based Local Outlier Factor (CBLOF) [50] or k-d trees [18] make use of this definition. An example is shown in Figure 4.9 (right). Cluster 1 and Cluster 3 are declared as anomalies in this definition. Those methods are exclusively used for unsupervised anomaly detection.

4.4.4 Nearest Neighbor Measures

The nearest neighbor techniques are also mainly optimized for the unsupervised case. The assumption is that normal data group in dense neighborhoods, whereas anomalies occur far from their closest neighbor [16]. The premise is thus similar to some clustering algorithms, but the dense areas are not expected to form well-behaved clusters.

The methods can be split into global and local methods. The global methods take the density of all points into account for the anomaly classification. Methods are e.g. the distance to the k-th nearest neighbor or the average to the k-nearest neighbors. In comparison to that, local methods compare the density of the new point to the density of its neighbors. Methods are e.g. Local Outlier Factor (LOF) [12] or Connectivity- based Outlier Factor (COF) [109]. Figure 4.10 visualizes the difference between local and global measures, local measures will handle the imaged cluster Cluster 1 as normal, whereas global measures are likely to classify Cluster 1 as anomalies.

4.4.5 Spectral Measures

Spectral anomaly detection is one of the few methods, which is explicitly developed for the semi-supervised setups. They follow the premise that data can be transformed into a lower dimensional subspace, in which "normal instances and anomalies appear significantly different" [16]. We will have brief look into two methods:

- *Principal Component Analysis (PCA)*
  The PCA is usually known as dimensionality reduction method. The algorithm searches for a set of linearly uncorrelated variables called principal components, and orders them according to their variance. For reducing the number of components, the ones with the least variance are ignored. The error, which is caused by that reduction is called reconstruction error.
  In case of anomaly detection [100], the reference data is also projected on a lower dimensional set of principal components and the corresponding reconstruction error is noted. For new data points, the same transformation is applied and the reconstruction error is monitored. A high reconstruction error is a sign for an anomaly. Interestingly, for anomaly detection, one cannot just leave out the components with the least variance but also other sets like the ones with the highest variance. In this way, anomalies significant in different components can be also detected.
  This method is a very easy and fast anomaly measure but works best for ellipsoidal shaped training data.

- *Autoencoder*
  Autoencoder are neuronal networks which try to reduce the number of necessary neurons without loosing much information. It consists of two parts: the encoder compresses the data and the decoder tries to reconstruct the original again. As training measure, the reconstruction error is used. In this way, the goal of an Autoencoder is to find the balance between a very compact representation and a small reconstruction error.
  This network is perfectly suitable for semi-supervised anomaly detection. The neuronal network learns a compressed version of the healthy reference data. If the reconstruction error raises during scoring, the data point is declared as anomaly. There is a variety of different configuration of Autoencoder, especially good results were found with Variational Autoencoder [4].
  The hidden layers of an Autoencoder allow to learn complex structures of the training data. But this goes hand in hand with high computational complexity and a black-box model, which cannot be visualized as easily as most of the other methods.

Those methods have the big advantage that they can also handle high dimensions and complex structure. They are especially popular as a preprocessing step. Nevertheless it is often difficult to prove that the premise is fulfilled. Additionally, for Autoencoder, the high computational complexity cannot be neglected.

4.4.6 Information theoretic Measures

Anomaly detection techniques based on information theory assume that anomalies induce irregularities in the information content of the data set [16]. There are different possibilities how to calculate the information criterion. Here, three concepts will be introduced:

- In information theory, the "Kolomogorov complexity" [75] declares a group of measures, which describe the computational resources. An interesting example is using the size of the compressed data file as anomaly measure.
- The Spearman rank-order correlation coefficient is a non-parametric measure of the monotony of the relationship between training data set X and the scoring data set Y. For the calculation the data sets are each ordered in size and each value is assigned to a rank number rg_x and rg_y. The Spearman coefficient is based on the Pearson correlation coefficient (see next Subsection 4.4.7), but with using the ranks in place of the actual values [66]:

  $$r_s = corr(rg_x, rg_y) \tag{4.10}$$

  In comparison to Pearson correlation coefficient, the two data sets don't need to be normally distributed.
- The Kullback-Leibler (KL) divergence and the Jensen-Shannon (JS) divergence are entropy measures, which compare the distributions of two data sets. In the semi-supervised anomaly context the new data distribution is compared to the healthy reference data and provide fast information if the data is drawn from the same distribution.
  The KL divergence is used in a lot of different fields, and is also known as information gain. It is a measure how well information is being compressed. For discrete distributions $P(x)$ and $Q(x)$, the KL divergence is defined as [81]:

  $$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right). \tag{4.11}$$

  The JS divergence improves the KL divergence by transforming the score into a symmetric smoothed measure between zero and one:

  $$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M) \tag{4.12}$$

with $M = \frac{1}{2}(P + Q)$.

Those information measures can be powerful tools, but the result - for sure - depends strongly on the choice of the information measure.

4.4.7 Collective Anomaly Measures

So far, all introduced methods detected point-like anomalies, with ignoring the temporal or sequential information.

The number of methods for collective anomalies is limited as usually the data is being transformed by the above described feature extraction methods. In this way, collective anomalies show up as point anomalies.

To complete the picture, we introduce two direct anomaly methods for collective anomalies. They are based on the constraint that the patterns are aligned and have a constant length.

- One method is to learn the average curve and its confidence interval. As soon as the curve deviates too often or too far from the learned curve, an anomaly is present. This method is good if the confidence interval of the curve varies within the curve and small details are not of huge importance.

- Another method is the Pearson correlation coefficient. It gives a fast intuition about the linear correlation of the training curve X and the scoring curve Y [121]

  $$\text{corr}(X, Y) = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y} \tag{4.13}$$

  with $\text{cov}(X, Y)$ being the covariance of these two variables and σ the corresponding standard deviation.
  The correlation coefficient gives a fast and easy method to detect differences in the shape. In comparison to the above method, the correlation coefficient just takes one healthy curve. Thus, confidence intervals should be reasonably small. Additionally, just the shape is taken into account, the amplitude is ignored.

4.5 Anomaly Evaluation

In this section, we introduce a method for comparing the different anomaly algorithms for the case of Predictive Maintenance. The main challenge is the unknown timing when the failure actually started. Often, the only available information is the time stamp when the error was detected by the on-site staff and repaired.

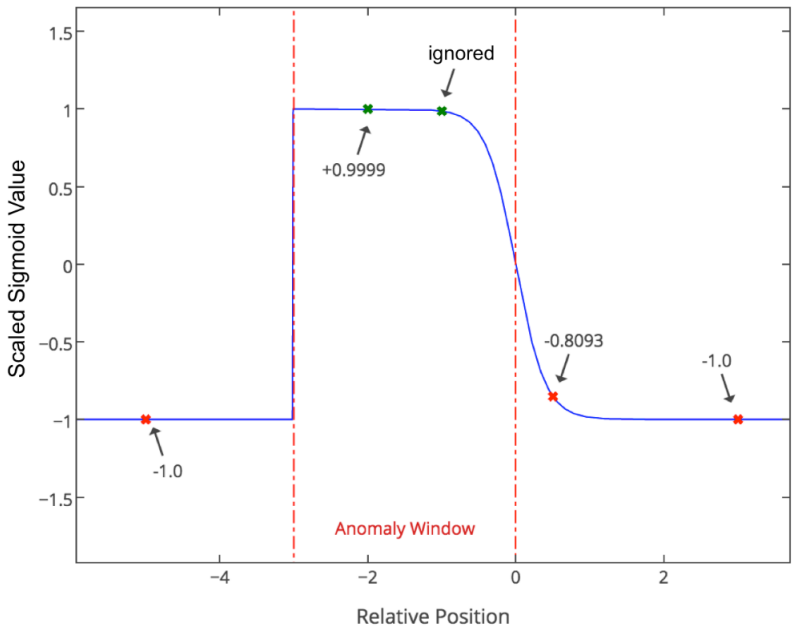


Figure 4.11: Scoring example of NAB score. The stars represent the detected anomalies by the algorithm, and the blue curve is the sigmoid weighting function for the error cases detected at $t = 0$. Here, the NAB score calculates as $-1.0A_{FP} + 0.999A_{TP} - 0.8093A_{FP} - 1.0A_{FP}$ with A_{TP} and A_{FP} being the weighting constants for True-Positive and False-Positive. [70]

This makes a special evaluation method necessary, which rewards early detection and punishes random anomalies. Commonly used evaluation scores like confusion matrix are not able to take those aspects into consideration and their results can be thus misleading.

Numenta Anomaly Benchmark (NAB) [70] is a score which can tackle those conditions. It is based on the assumptions that anomalies are rare events, and that a maximum time window between the failure start and the detection can be defined, a so-called "anomaly window". The earlier the algorithm detects the error case within the anomaly window, the higher the found anomaly is rewarded. As pictured in Figure 4.11, every anomaly window is modeled by a sigmoid function with the zero-crossing at the event time y:

$$\sigma^A(t) = \big(A_{TP}\,\Theta(y < 0) + A_{FP}\,\Theta(y > 0)\big)\,\big(\frac{2}{1 + \exp(5y)} - 1\big). \tag{4.14}$$

$0 \leq A_{TP}, A_{FP} \leq 1$ are respective the weighting for True-Positives and False-Positives and y the relative position of the detection in the given anomaly window. Θ represents the Heaviside function. As every event should be just considered once, just the earliest detected occurrence counts. This implies that a failure de-

tected at the beginning of the anomaly window contributes with approximately +1, a failure detected at the same time as the on-site people contributes with 0, and a random anomaly without context of an event is rated with -1. For an overall score, all individual scores are summed up, with additionally adding the number of missed events f weighted by a false-negative constant $-1 \leq A_{FN} \leq 0$:

$$S^A = \Big(\sum_{y \in Y} \sigma^A(y) \Big) + A_{FN} f. \tag{4.15}$$

In order to receive a score between 0 and 100, the score is normalized with "perfect" detector S^A_{perfect} and the "null" detector S^A_{null}:

$$S^A_{\text{NAB}} = 100 \cdot \frac{S^A - S^A_{\text{null}}}{S^A_{\text{perfect}} - S^A_{\text{null}}} \tag{4.16}$$

A very essential part of the algorithm is fixing the weightings of True-Positives A_{TP}, False-Positives A_{FP} and False-Negatives A_{FN}. As already introduced in Section 4.2, use cases differ strongly in their tolerance for false alarms or missed anomalies. In cases in which false alarms cause high costs, the weighting of False-Positives should be increased. In cases in which undetected failures are not acceptable, the weighting of False-Negatives should be enhanced. This allows a flexible adjustment to different use cases.

As noted in [102], the respective formula for Eq. 4.14 was erroneous in the original paper [70]. Thus, it was corrected here according to example in the paper and the implemented code on which [70] is based on.

The NAB evaluation method was proven successful in other use cases [2], but was also harshly criticized from Singh [102]. Points of criticism were the difficult choice of the anomaly window size, the unexplained magic number of "5" in Eq. 4.14, and the implementation, which cannot handle non-equidistant data. Whereas the choice of the anomaly window can be fixed for every use case with expert knowledge, and small modifications of the implementation allow non-equidistant data, a major drawback was surprisingly not mentioned by Singh. Although the NAB score is explicitly designed for real-world data, it can run into problems for a labeled data set acquired in a non-academic context. For instance, some error cases are detected and repaired right after emerging, even before they can show any signature in the data. If this happens more often (which is a sign for well trained staff), the NAB score can prefer a random generator over any data-based anomaly score. We will discuss and illustrate this drawback in further detail in Section 4.6.2 and provide an improvement.

4.6 Improvements Methods

In the following, we will pick two of the above discussed methods and provide some improvements in terms of stability and usability for semi-supervised anomaly detection.

4.6.1 EMD for Semi-supervised Anomaly Detection

As described in Section 4.3.1, the Empirical Mode Decomposition (EMD) decomposes a signal into its Intrinsic Mode Functions (IMF), each of which contains information of a specific frequency band. Applying the EMD to a signal at separate time intervals, the EMD optimizes the IMFs for each time interval separately. Consequently, a specific frequency band is not always represented by the same IMF, but can vary in different time intervals. This happens in particular, when specific frequencies don't appear all the time (like error frequencies).

This leads to a major problem for semi-supervised anomaly detection. Taking the anomaly measure "energy per IMF" as an example, the energy vectors over time have the energy for specific frequencies at different positions in the vector. Comparing those vectors directly easily leads to misinterpretation. This can be seen in Figure 4.12, the IMF2 carries completely different frequency information on the left ($104\,\text{Hz}$) than on the right ($152\,\text{Hz}$). Additionally, the number of extracted IMFs can vary, which makes a comparison of vectors with different lengths very difficult.

This challenge has already been reported in a couple of papers in different contexts. Zhao [132] reports a similar issue for data from multiple sources and uses a multivariate EMD in order to tackle it. Faltermeier [32] aims to monitor time series in a continuous manner, and therefore proposes a new approach of a Sliding EMD.

In this thesis, a different approach is proposed for the case of semi-supervised anomaly detection. First, the EMD extracts the IMFs in a normal manner, then, a small matching algorithm assigns each IMF to a specific order number depending on its frequency:

1. Create Reference IMF (RIMF):
   In the training process, the EMD decomposes the signal during a healthy time period into its IMFs. Those are used in the following as Reference IMFs (RIMFs). The Hilbert-Huang Transform (HHT) extracts the instantaneous frequencies of each IMF. If the average frequency for an IMF differ less than 10% to another one, they are summarized to one IMF. For that the new IMF is the sum of the old IMFs, and the frequency is the mean of the old frequencies.

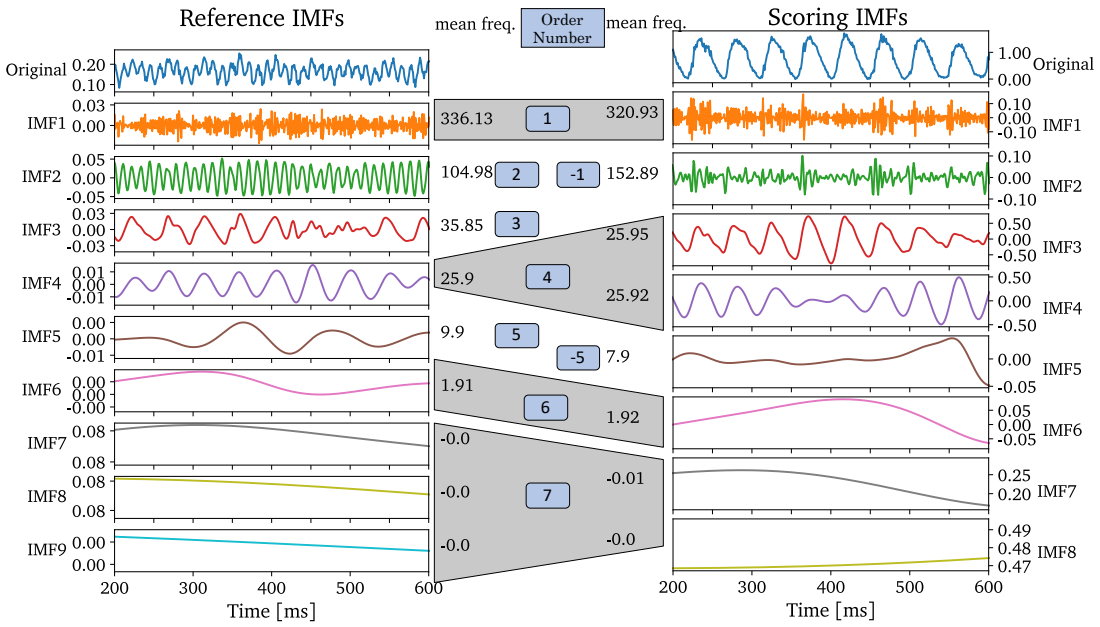


Figure 4.12: Extract from a Reference IMF and a Scoring IMF, and their matching via the average instantaneous frequencies. All IMFs, whose frequencies differ by maximum $10\,\%$, are matched together. Scoring IMFs, whose frequencies are not represented in the Reference IMF, receive the negated next higher order number. If several IMFs are mapped to the same order number, they are summed and considered as one IMF for the following feature extraction. This allows direct comparability between IMFs extracted independently for different time intervals.

Thus, RIMFs with similar frequency bands are summarized to one RIMF. Additionally, all IMF with negative frequencies are summarized to one IMF. According to the algorithm, negative frequencies are not possible, but small negative frequencies appear nevertheless due to numerical inaccuracies.
As result, each RIMF is mapped to a distinct frequency and an order number $[1, .., n]$, which sorts the RIMFs by frequency.

2. Map to Reference IMF:
During the scoring process of a new time interval, the new IMFs are mapped to the RIMFs. Equivalently to the training, the HHT extracts the mean frequencies of each IMF. If the frequency varies less than 10% to a reference frequency, the IMF is mapped to the according RIMF, and receives its order number. All IMFs that cannot be mapped to a RIMF, use the order number of the next higher RIMF, and negate it. It can happen that several IMF are mapped to same order number. In this case, they are summed to one IMF. Same as for the RIMFs, all IMFs with negative frequencies are summed to one IMF. Thus, every IMF is matched to an order number, which is positive in case the frequency band occurred in the RIMFs, and negative if the frequency lies between two frequency bands in the RIMFs.

This matching algorithm maps all similar IMFs to a specific order number, which allows comparison. It has to be noted that every IMF is assigned to an order number, but not every order number is assigned to an IMF. This has to be considered for the evaluation of the EMD features.

For visualization, the vector "average instantaneous amplitude per IMF" for the Reference and Scoring IMFs in Figure 4.12 is a vector of the length 13 with the order numbers $\{-6, .., -1, 1, .., 7\}$. All order numbers which do not appear, are replaced with zeros. Thus, the two vectors look like:

$$\begin{aligned}
\vec{a}_{\text{RIMF}} = \Big[&0.00 \quad 0.00 \quad 0.00 \quad 0.00 \quad 0.00 \quad 0.00 \quad 0.01 \\
&0.04 \quad 0.01 \quad 0.01 \quad 0.00 \quad 0.00 \quad 0.16\Big]^T \\
\vec{a}_{\text{Scoring IMF}} = \Big[&0.00 \quad 0.00 \quad 0.02 \quad 0.00 \quad 0.00 \quad 0.03 \quad 0.06 \\
&0.00 \quad 0.00 \quad 0.66 \quad 0.00 \quad 0.06 \quad 0.71\Big]^T.
\end{aligned}$$

These features vectors are directly comparable and can be used as basis for semi-supervised anomaly detection algorithms.

4.6.2 Improve NAB score for real-world Labels (iNAB)

As already mentioned in Section 4.5, the Numenta Anomaly Benchmark (NAB) score can run into problems for a labeled data set acquired in a non-academic context. We will introduce possible characteristics of such a data set, visualize the effects on the NAB score, and then provide an improved NAB score.

Data sets published in academia usually have rather high quality labels, as they are mainly used for comparing newly developed algorithms to a reference. This stands in contrast to labels acquired in industry. Particularly three aspects decrease label quality significantly:

Some error cases are not documented. The job of the machine staff is to keep the machine running, and not to provide a full documentation. In some stress situation, it can happen that documentation is missing or incomplete. Additionally, maintenance activities performed outside of production times are just fragmentary documented, as they are not needed for internal management reports.

Staff repairs error cases right after emerging. Faults often don't build up over a long time, but are caused by an unrelated machine crash. When checking the machine afterwards, the staff directly detects and repairs the error. Technically, those error cases should not be considered for any algorithmic evaluation. Distinguishing between sudden faults and faults which build up over time is often not possible by the labels. Even with the corresponding documentation, it is difficult to determine if only the crash, or the crash in combination with a pre-damage lead to the final error. Thus, there are always documented error cases, which - by principle - cannot be detected in the data.

Some error cases cannot be detected in the available data. As example, a motor failure can be caused by either a mechanical or an electrical origin. Whereas mechanical origins usually show characteristic patterns in the torque and temperature of the motor, an electrical short-circuit or a problem in the machine network leave those sensor data rather unaffected. Excluding those error cases from the evaluation is not possible. Even with a detailed error description, nobody can guarantee that the error case cannot be seen in the data without examining the data in great detail.

Those three effects can lead to undesirable consequences when comparing different anomaly algorithms via the NAB score. Figure 4.13 illustrates this by calculating the NAB score for a data-driven score and a Random Generator on exemplary data. Non-documented error cases (in the Figure case 2) decrease the

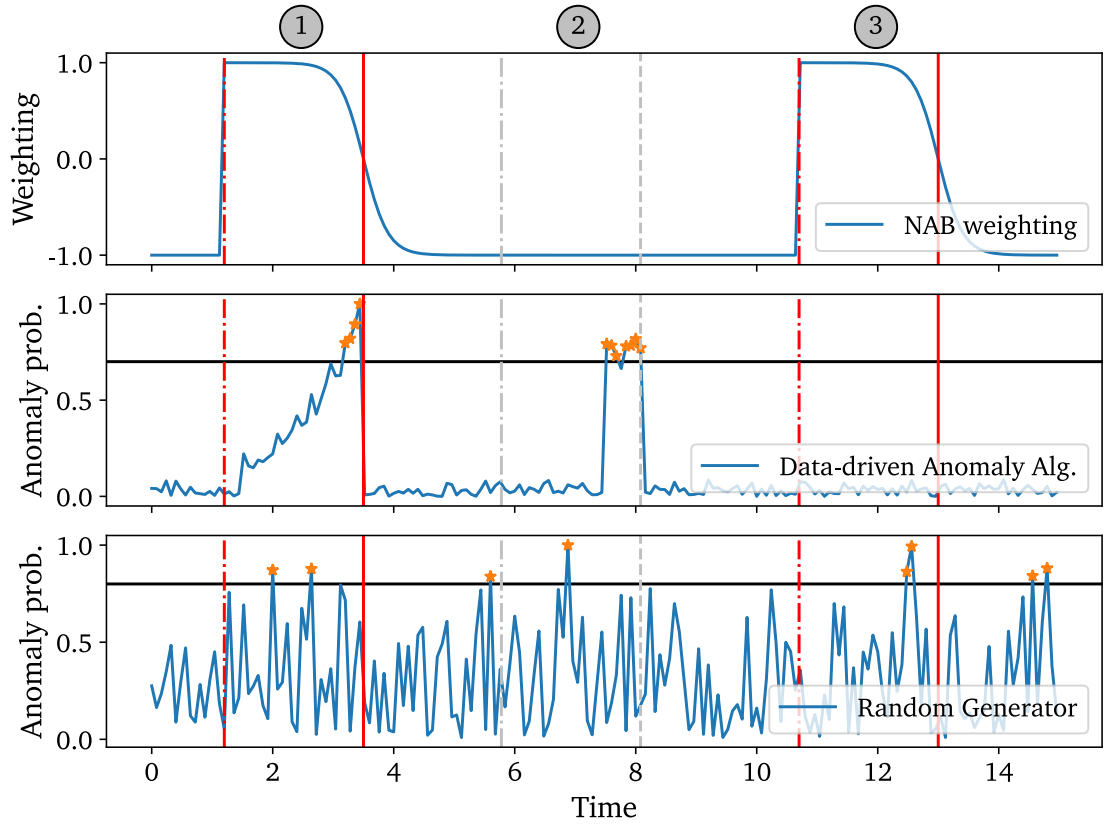


Figure 4.13: Visualization of the three possible cases, case 1 ($t = 3.5$) appears in the labels (red vertical lines) and the data, case 2 ($t = 8.0$) appear just in the data, case 3 ($t = 13.0$) just in the labels. The NAB score is evaluated exemplary once for an anomaly probability created by a data driven algorithm (2nd row), and once for a random generator (3rd row). Anomalies (orange stars) are extracted from the anomaly probability by defining a limit. As NAB weighting factors, the standard profile is taken: $A_{TP}, A_{TN} = 1$ and $A_{FP}, A_{FN} = -1$. The top figure shows the sigmoid function with which every found anomaly is multiplied. The anomaly window is chosen to $\Delta t = 3.3$. The data-driven anomaly score calculated via $0.64 \cdot A_{TP} - 7 \cdot A_{FP} + 1 \cdot A_{FN}$ yields to a score of -7.36. The NAB score for the random generator outperforms with $(0.99+0.86) \cdot A_{TP} - 4 \cdot A_{FP} = -2.15$ substantially the score for the data-driven anomalies. Nevertheless, both scores perform worse than the null reference $S^A_{\text{null}} = -2$, which leads each to the normalized score of $S^A_{\text{NAB}} = 0$. Consequently, by using the definition of Lavin [70], it can happen that the NAB score does not carry any relevant information.

score drastically, as every detected anomaly separately counts as false-positives, whereas every detected error case just counts as one true-positive. Thus, scores can loose their information content, and even worse, can be outperformed by random generators or the null reference.

In reality, the influence of the extreme punishment varies strongly with the data set, and usually a random generator is not explicitly used as anomaly algorithm. Nevertheless, some anomaly algorithms usually fail and their results resemble strong similarities with a random generator. Consequently, the NAB score cannot be fully trusted, and the result of the winning anomaly algorithm always has to be rechecked manually.

In order to receive a more reliable result, we suggest the following three improvements to the NAB score (iNAB):

Ignore single anomalies within anomaly windows. Only if at least N anomalies are detected in a row, the earliest one is used for the score. The parameter N has to be adjusted to the nature of the data. Due to statistics, even $N = 2$ or $N = 3$ has a large impact on the quality of the NAB score. This small change decreases the chance of a random generator to achieve a true-positive drastically. As side effect, anomaly algorithms producing a "smoothed" anomaly score will be privileged over noise scores, as they usually fulfill the additional rule faster. To enable noise scores also to win, single hits in the window are ignored and not punished.

Add data-driven error cases. The principle is based on majority voting. If the large majority of algorithms is pretty sure about an anomaly, or some algorithms are extremely sure, an additional error case is marked in the data. For this, we propose following procedure: Every anomaly score is normalized by its 98 percentile to guarantee comparability. In case the average score of all scores exceeds 80%, the final time point will be added to the documentation of error cases. This procedure can be performed only if a sufficient number of algorithms is evaluated, which additionally examine different characteristics of the data. As guideline, at least 20 different settings should be used. This should not pose a problem, as the NAB score is explicitly designed for comparing a number of different algorithms. Additionally, it is based on the assumption that the large majority of algorithms produce meaningful results and just a small minority acts as random generator.

Add artificially created reference scores for comparison. In order to get a better feeling how well the scores are really performing, we propose adding several references created by Random Generators, which are based on different distributions. All algorithms which perform similar or worse than any

of those references should be taken with caution. To be noted, the additional references should be added after determining data-driven error cases to not influence the result.

Figure 4.14 applies the improvements to the example above, and calculates the iNAB scores. Adding the additional data-driven error case increases significantly the data driven score. Additionally, ignoring single anomalies in anomaly windows has a large impact on the Random Generator, as not a single true-positive can be generated. Thus, the data-driven score is now able to significantly outperform the Random Generator.

Summarized, the NAB score [70] can fail in handling real-world label effects, though claimed in the publication. With three improvements, the improved NAB score (in the following iNAB) has the potential to be a by far more stable and meaningful score. In order to ensure the functionality on real-life data and labels, the two scores will be compared and evaluated in Section 5.4.

4.7 Summary

In this chapter, an introduction into the field of anomaly detection was given, with a focus on the semi-supervised kind. After defining the meaning of anomaly, the different steps of a study were explained on the basis of the CRISP-DM cycle, and differences between a machine learning study and an anomaly detection study were pointed out. After that, algorithms for different steps of a study were introduced. For extracting characteristics of the data, a variety of feature extraction methods was presented, which already proved successful in related use cases. Based on the extensive review study of Chandola [16], an exemplary selection of semi-supervised anomaly methods was established with emphasizing the different premises. For comparing the different anomaly methods, an approach called NAB score was introduced, and critically analyzed. In the last section, a new approach was discussed, which allows using the EMD in a semi-supervised anomaly setup. Additionally, improvements on the NAB score were suggested and proven successful on a small exemplary data set.

This chapter was just able to give a brief overview over the very broad field of anomaly detection. The choice of algorithms depends a lot on the use case, and the kind of anomalies which should be detected. A contribution to the further development on anomaly detection was made.

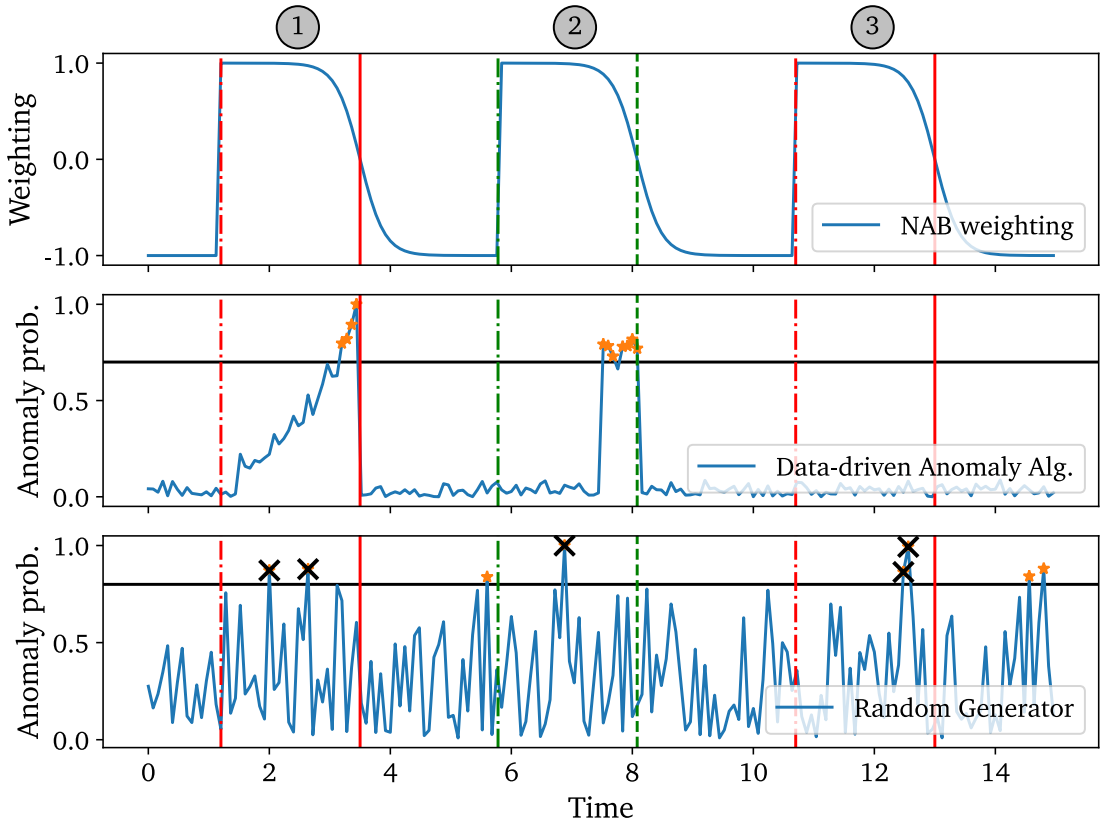


Figure 4.14: Evaluation of the NAB improvements on the example in Figure 4.13, all parameter are set accordingly.
First, the second error case is detected in the data and added to the list of error cases. Second, all single anomalies in an anomaly window are ignored with $N = 3$ (black crosses). This improves the NAB score for the data driven anomalies significantly from -7.36 to $(0.64 + 0.89) \cdot A_{TP} + 1 \cdot A_{FN} = 0.53$, leading to $S^A_{\text{iNAB}} = 58.8$. The random generator cannot generate a hit for any error cases, and looses with $3 \cdot A_{FP} + 3 \cdot A_{FN} = -6$ and $S^A_{\text{iNAB}} = 0$. The small example demonstrates the effect of the improvements, and show their effectiveness. The data-driven anomaly score is now clearly the winner.

5 One-shot Semi-supervised Anomaly Detection - Study

This chapter will link together all previous chapters. The anomaly detection algorithms introduced in the last Chapter 4 will be applied on the data of bottle transport error cases, which were motivated in Chapter 2. The physical understanding of the system of Chapter 3 will assist for choosing a selection of feature extraction methods and anomaly algorithms.

The approach of anomaly detection in the field of special mechanical engineering is very promising, as no data of error cases is required, and it allows easy transfer between similar machines that differ in details. Although anomaly detection is already established in a lot of fields, there is a special challenge in this case. The goal is to find an algorithm, which detects anomalies in patterns reliably with using as little resources as possible. The training data is restricted to one healthy pattern as manual labeling of longer times for every new machine is not feasible. This leads to the new concept of one-shot semi-supervised anomaly detection. Additionally, resources for scoring new incoming patterns are strongly restricted as well. The calculation has to be performed on an edge device without interfering with other services running on the device. This excludes approaches like multi-model solutions with majority vote.

In this study, a variation of different algorithmic setups will be challenged. The winning algorithm will be compared to a score based on the physical properties discovered in Chapter 3, and the concept of Transfer Learning will be tested. Additionally, the functionality of the iNAB score, introduced in Section 4.6.2, will be examined on real data.

5.1 Structure Study

The study is structured in the CRISP-DM cycle as introduced in Section 4.2. For better orientation, Figure 5.1 visualizes the main steps with including a small image of each step.

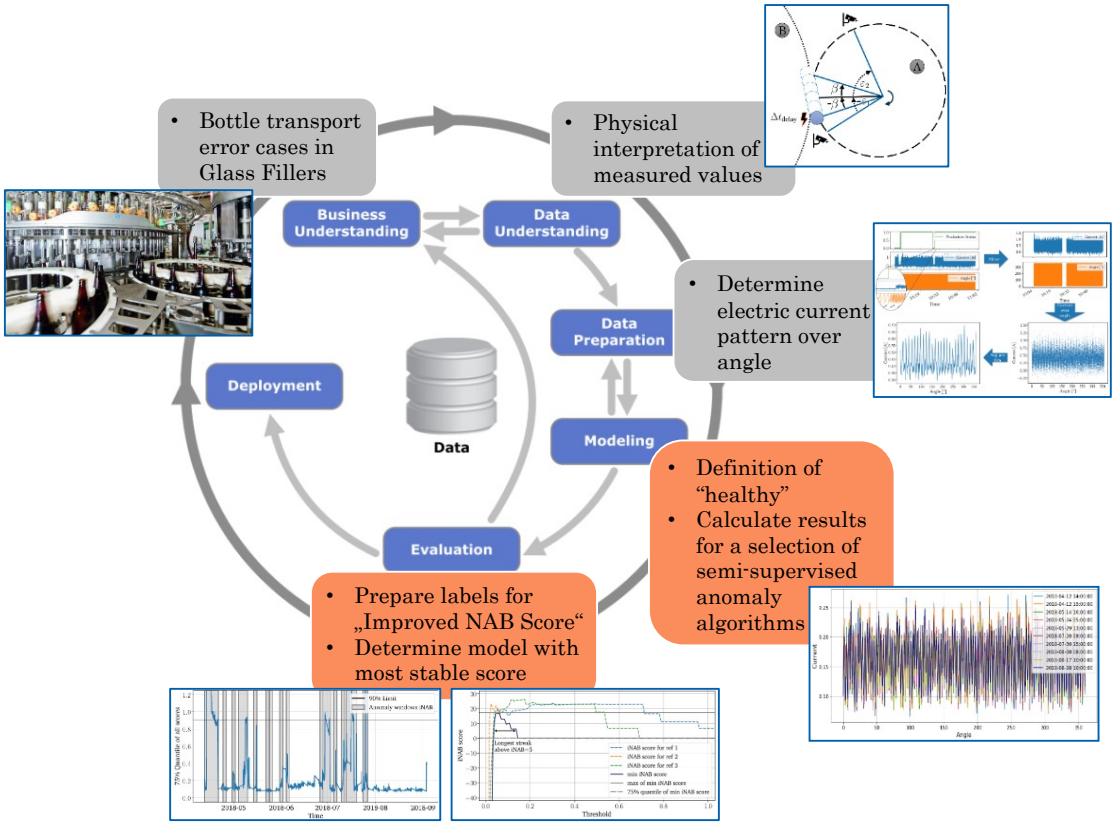


Figure 5.1: Schematic structure of the study based on the CRISP-DM cycle (modified from [58]). This chapter will concentrate on the fields of modeling and evaluation.

The first three steps of the cycle - Business Understanding, Data Understanding and Data Preparation - were already discussed in Chapter 2 and 3. The most important insights can be summarized in the following way:

Business Understanding Bottle transport errors in Filler stars were recognized as one of the main causes of loss of production time.

Data Understanding Considering the physical behavior of the machine, built-in sensors in motors, which drive the transport stars, seem rather sensitive to error cases. Taking the implementation of the control into account, the electrical current is the most suitable candidate as it responds rapidly to changes in load (in comparison to, for example, temperatures), and the recorded values are raw values, which are not yet processed or averaged in the control.

Data Preparation As the sampling rate of the recorded data is too low for detailed analysis, a trick was developed within this thesis. Additionally to the current, an angle position of the star is logged. Taking the two measures into relation, an average current per angle interval is calculated, and a high-resolution pattern can be extracted. This routine is performed every hour on all production data.

The two following steps, Modeling and Evaluation, build up on those findings. In the Modeling Section 5.3, a collection of semi-supervised anomaly detection algorithms will be selected in order to find anomalies in the data. In the Section 5.4, the stability of the newly developed iNAB score is tested. In the Evaluation Section 5.5, the different algorithms are finally compared to each other with different stability criteria based on the iNAB Score.

As already mentioned in the introduction, the main challenge will be the stability of the algorithm despite strongly restricted resources:

Restricted training data The training data is very limited. The expert marks just one sample as healthy.

Restricted scoring resources Running the calculation on an edge device, the computational power is limited. It is not possible to calculate several resource-intense algorithms and perform a majority vote.

Stability of result The result has to be very stable and should not depend intensively on the training data as there are variations of the healthy state which should be all classified as healthy. This generalization is one of the main challenges in one-shot learning.

5.2 Details about Data and Labels

The study is performed on data, which was acquired by a filler in a beer brewery. The machine is running with a production rhythm of always two production days and one maintenance day. Data was recorded over the course of five months from April 2018 to September 2018 with a resolution of $100\,\text{ms}$. During that time, 15 transport errors were documented by the machine operators. All of them affected the infeed starwheel. Thus, the focus will be on this particular transport star. It has to be noted that just errors that lead to a loss of production time of at least 10 minutes were noted, and that there is no documentation available about repairs in maintenance windows. The error cases are documented on a daily granularity without any information about the exact timing of the event.

5.3 Modeling: One-shot Semi-supervised Anomaly Detection

As introduced in Section 4.1, semi-supervised anomaly detection always consists of two steps. First, some data is labeled as a healthy reference. This data should not contain any error cases. Here, due to the one-shot setup, just one healthy state is defined. Second, all further unlabeled data is compared to the reference in order to find anomalies. The characteristics of the data are often extracted via feature extraction, which precedes the calculation of the anomaly measures.

5.3.1 Label Healthy States

The first very essential step is labeling a healthy state. For this study, the healthy state is defined by a Krones machine expert. He chooses a time window in which no error case is documented, and then checks the angle-current curve by eye. With experience and knowledge about the machine setup, he assesses the curve.

As the final goal of this study is to transfer the results to further motors and bottling plants in a larger scale in future, some practical aspects are highlighted. When choosing the healthy state, the expert usually cannot access months of data, but just one to two weeks. This implies that the data is limited, and for practical reasons just one healthy state is defined. Additionally, depending on the timing, the defined healthy state may vary, as the machine is naturally subject to small fluctuations. For the anomaly detection, it is very important that the result is not strongly influenced by those variations. In order to test this stability, the Krones

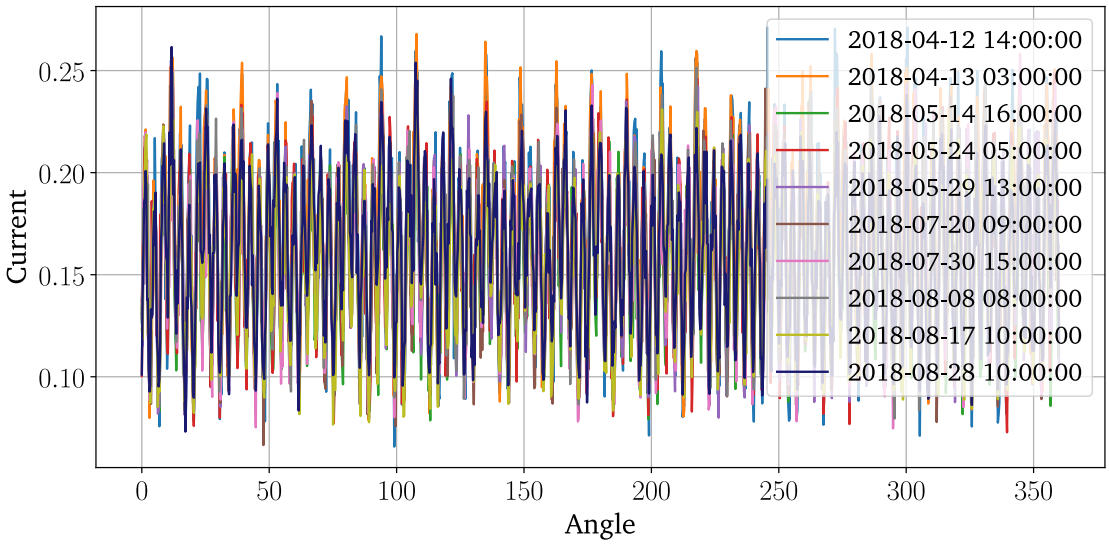


Figure 5.2: Those ten curves were classified as healthy states by an expert. Each of them will be separately used as reference curve.

expert chooses for this study in total 10 healthy curves, spread over almost five months.

Comparing the chosen healthy curves in Figure 5.2, they show in general a similar behavior with variations in amplitude. This is encouraging as it seems that the fluctuations over time are small and the machine returns to a similar healthy state after error cases.

5.3.2 Feature Selection

Section 4.3 introduced a variety of feature selection methods that proved successful in detecting machine failures. In this comparative study, most of them will be used in order to analyze their behavior. In the following, the specific settings are defined.

Frequency based Features

Both, the physical examination of the system, and the strongly oscillatory behavior of the reference curves, suggests the usage of frequency based features. The Fast Fourier Transform (FFT) and the Empirical Mode Decomposition (EMD) will be used in the following.

The FFT is used as a data transformation in the frequency space, and the resulting frequency-amplitude information is used as a long feature vector. Additionally, taking the knowledge of the physics chapter into account, once the amplitude

of the handover frequency, and once the sum of the handover frequency and its multiples, are taken as separate features.

For the EMD, the ensemble variant EEMD is used with averaging over 100 settings. To ensure comparability, the mapping algorithm proposed in Section 4.6.1 is used. As reference IMFs, the IMFs of the above defined reference curves are taken. The mapping is performed for every reference curve separately, thus in total 10 times. In terms of features, the following eight are chosen (definitions can be found in Section 4.3.1):

Energy content: Total energy of all IMFs E_{tot}, Energy vector of all IMFs $\vec{E} = [E_{-n+1}, .., E_n]$ with n being the number of IMFs for the reference

Local amplitude: Vector with the average of the local amplitude per IMF, Vector with the variance of the local amplitude per IMF

Local frequency: Vector with the average of the local frequency per IMF, Vector with the variance of the local frequency per IMF

Teager-Kaiser Energy-tracking Operator (TKEO): TKEO vector of all IMFs, TKEO vector of IMF -3 to 3 (similar to [107])

For calculating the EEMD, the package PyEMD of Laszuk [69] is used.

NMF

As introduced in Section 4.3.2 the Non-Negative Matrix Factorization (NMF) separates patterns from different origins in different components. Having a lot of different moving parts in the machine that are driven by the same motor, the NMF is expected to separate those different origins efficiently.

For the training of the NMF, several curves are needed. As the labeling only provides one healthy curve, twenty neighbor curves are taken for the training. This adds some randomness to the extracted modes, depending on whether the healthy curve is embedded in healthy curves or faulty curves. This should not compromise the anomaly analysis, as the weighting factors are always compared to the healthy curve.

Additionally, the parameter “number components” has to be set, which specifies the number of influences. As the number of influencing components is not known, the NMF is calculated for a set of two, three, five and ten components.

The NMF implementation of scikit-learn [91] is used in the following.

Global Features

Three sets of global features will be used for the study.

The first set is the collection of standard statistical features: mean, standard deviation, minimum, 25% percentile , median, 75% percentile and maximum.

As second set, the large number of features defined in the package 'tsfresh' [21] are used. This package was introduced in Section 4.3.3. Without having error data, the dimensional reduction offered by the package is not possible. Ignoring features that are mainly zero, reduces the number of features from 670 to 514.

In contrast to this extensive feature set, the third set is the reduced set of in total 22 features of the package 'catch22' [79].

It will be interesting how those three contrary approaches will perform in comparison to each other.

Noise Reduction (Optional)

Every described method can be combined with an optional process of noise reduction ahead of the feature extraction. Reducing the noise has the potential to enhance the significance of the extracted features, assuming the noise does not contain any significant information.

The EEMD, as introduced in Section 4.3.1, is used as noise reduction algorithm [69]. By removing the first IMF from the signal, the high-frequency information is eliminated.

5.3.3 One-shot Semi-supervised Anomaly Measure

In Section 4.4, a variety of anomaly measures was introduced. Taking the one-shot setup with only one single "healthy" reference into account, just a small set of algorithms is feasible for this use case. In the following, the five most promising anomaly measures of different categories are chosen.

Statistical Measures: This set of measures are based on the assumption that the distribution of healthy data can be described with a statistical model. With just one sample, it is not possible to estimate a distribution. As a trial, nevertheless, the *Mahalanobis distance* will be evaluated. Similar to the NMF, in order to estimate the covariance matrix, in total 20 curves before and after the reference curve are taken into account. This adds an uncontrollable factor to the evaluation, as the labels of the other 19 curves are unknown. It is assumed that a time point declared as healthy is embedded in a healthy time window, and the effect should be small. Nevertheless, variations between the 10 chosen reference samples are expected. The Mahalanobis distance

will be used for global feature sets, in which the different components are not comparable and need scaling.

Classification Measures: Most classification measures, which are suitable for semi-supervised anomaly detection, for instance OC-SVM, depend strongly on a sufficiently big labeled data set, and tend to act rather sensitive to incorrect labels. Thus, an approach of choosing further random curves similar to the Mahalanobis distance is not suitable here.

Clustering Measures: Clustering algorithms rely on large amount of data. This stands in contrast to the prerequisite that the anomaly detection should start working after just 1-2 weeks of data. Additionally, the algorithms usually require a large history of data for the evaluation, which conflicts with the limited scoring resources.

Nearest Neighbor Measures: Equivalently to the clustering measures, nearest neighbor distance can be usually only calculated with large amount of data.

Spectral Measures: Those measures show their true potential for high dimensional spaces. This criterion is fulfilled here, with for example taking every angle window as its own dimension. Nevertheless, those methods also rely on a large number of healthy samples in order to find a suitable representation. This is not fulfilled, and thus no spectral measure is taken for this study.

Information theoretic Measures: In contrast to the measures ahead, the *Spearman rank-order correlation coefficient* and the *Jensen-Shannon (JS) divergence* work with one sufficiently long healthy sample. Both compare the distribution or monotony of two samples without needing any further information. Those seem pretty well suitable for comparing the pure curve and the FFT curve as both of them are sufficiently long ($>$100 samples). The number of bins for the JS divergence is set to 50. The result seems to be rather independent of the number of bins.

Collective Measures: Point-to-point comparable patterns with fixed lengths are perfectly suitable for collective measures. Two measures are chosen: First, the rather simple point-to-point *euclidean distance*, and second, the *Pearson correlation coefficient*. Both measures depend on the fact that all dimensions have the same unit, and that changes in any dimension are weighted the same. Nevertheless, the euclidean distance is used on all possible features in order to provide an alternative to the Mahalanobis distance, though knowing that features in the global features will be weighted differently depending on

their size. The correlation coefficient needs a long enough sample for stable result, and is therefore used for the pure curve and the FFT.

All measures are transformed into an anomaly probability or distance: The higher the value, the more likely an anomaly is present. Consequently, for the scores of Pearson correlation and the Spearman rank-order correlation, high correlations are assigned to small anomaly probabilities, no correlation and anti-correlations are assigned to high anomaly scores.

All 44 combinations of feature extractions and anomaly measures can be found in Table 5.1. For each of them, an anomaly probability or distance is calculated. In case the calculation of one sample fails for some reason, it is assigned to normal (probability equals to 0).

5.4 Evaluation iNAB

In this section, the error cases are prepared once according to the NAB, and once according to the iNAB procedure. In both cases, the procedure is slightly adapted to the use case. The results are compared to ensure the reliability of the improved iNAB score.

5.4.1 Error case preparation NAB

As mentioned in Section 5.2, 15 transport errors on a daily granularity are documented. In the following, those are transformed into anomaly windows, and the weighting factors are defined.

Definition of Anomaly Windows

For the NAB score, anomaly windows are defined around every error case in order to reward early detection. In comparison to Lavin [70], the window size is not calculated by a rather arbitrary estimation, but is set to two days, in accordance to the maintenance intervals of the production line. This ensures that detected error cases can be always scheduled for the next non-productive time for examination.

As the machines are not running every day (for instance due to maintenance intervals), defining a fixed two-day-window leads to an inconsistency of the anomaly windows: During non-production, no score can be calculated, and thus, no anomalies can be detected. This inconsistency was also mentioned in Singh [102]. Here, this can be fixed easily by expanding the anomaly windows always to the last production day. Thus, the anomaly windows can span more than two days.

Feature engineering method	Noise reduction	Eucl. distance	Mahalanobis distance	Pearson correlation	JS divergence	Spearman rank-order corr
pure signal	yes/no	x		x	x	x
FFT - frequency pattern	yes/no	x		x	x	x
FFT - station frequency	no	x				
FFT - multiple station frequency	no	x				
EEMD - total energy	no	x				
EEMD - energy imfs	no	x				
EEMD - avg amplitude	no	x				
EEMD - std amplitude	no	x				
EEMD - avg frequency	no	x				
EEMD - std frequency	no	x				
EEMD - TKEO vector	no	x				
EEMD - TKEO IMF -3..3	no	x				
NMF - two comp	yes/no	x				
NMF - three comp	yes/no	x				
NMF - five comp	yes/no	x				
NMF - ten comp	yes/no	x				
Global - statistical	yes/no	x	x			
Global - tsfresh	no	x	x			
Global - catch22	yes/no	x	x			
Reference: 3x Random Generator	no	x				

Table 5.1: Overview over all 44 combinations of feature extraction (rows) and anomaly measures (columns) of this study. For completeness the three reference random generators are also mentioned.

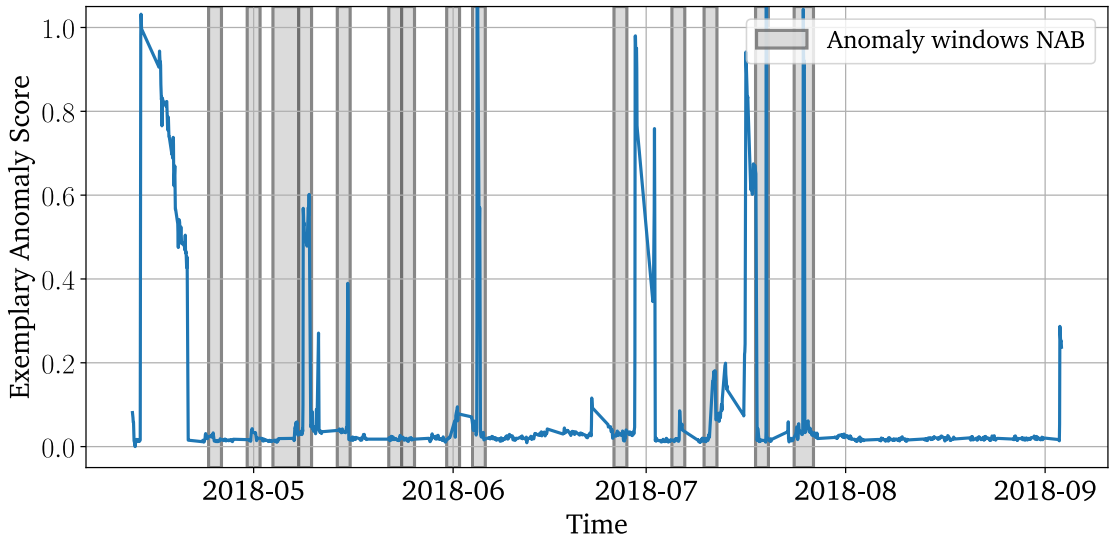


Figure 5.3: The anomaly windows expand all documented error days to the previous production day. Overlapping anomaly windows are summarized to one anomaly window.

As last step, overlapping anomaly windows are summarized to one anomaly window to avoid double-counting of the same error case. Successive error cases are not combined under the assumption that detected anomalies are immediately repaired, and thus disappear on the day of detection. In this way, anomalies which appear one or two days later are considered independent of the previous.

This procedure leads to 14 anomaly windows with lengths between two and four days (Figure 5.3).

Definition Weighting Factor

The definition of the weighting factors for True-Positive A_{TP}, False-Positive A_{FP} and False-Negative A_{FN}, is performed as proposed by Lavin: "The standard profile assigns TPs, FPs, and FNs with relative weights [..] such that random detection made 10% of the time would get a zero final score on average." [70] As $10\,\%$ marks in the paper the total length of all anomaly windows, this number is adjusted to the corresponding $15\,\%$ in this study. The standard profile is chosen to $A_{TP} = 1.0$, $A_{FP} = 0.064$ and $A_{FN} = -0.064$, which fulfills the condition described in the paper.

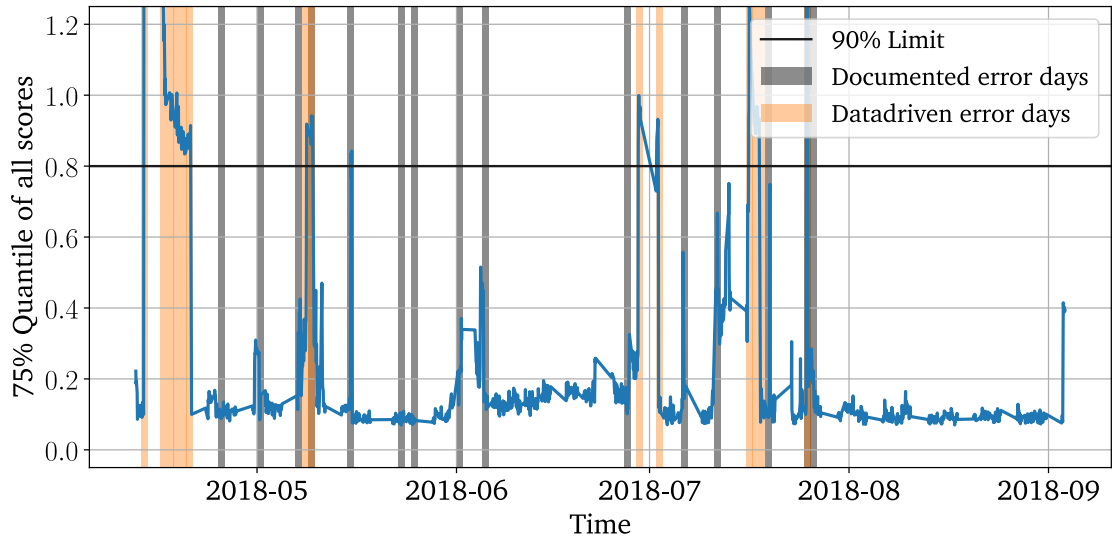


Figure 5.4: The documented error days are marked in gray. Data driven error days are added, if one quarter of the algorithms is to at least $90\,\%$ sure about the anomaly. Those days are marked in orange.

5.4.2 Error case preparation iNAB

The error case preparation of the iNAB score works equivalently to the NAB score preparation, with the additional step of adding data driven error cases.

Data Driven Error Cases

As described in Section 4.6.2, error case documentation is often not as reliable as hoped. As false positives over longer time periods are strongly punished by the NAB score, the iNAB score adds data driven error cases. In this study, anomalies are added, if at least one in four algorithms is at least 90% certain about the anomaly. In order to make all 440 calculated scores comparable (44 algorithms with each 10 reference curves), each of them is normalized by its 98%-quantile. As shown in Figure 5.4, 14 days fulfill the condition, of which two are already in the documented list, and four are ahead of documented error days. In combination with the documented error cases, the iNAB score is based on 27 error days.

Definition of Anomaly Windows

The procedure of defining the anomaly windows is performed completely equivalently to the NAB score. Though doubling the number of error days, the number

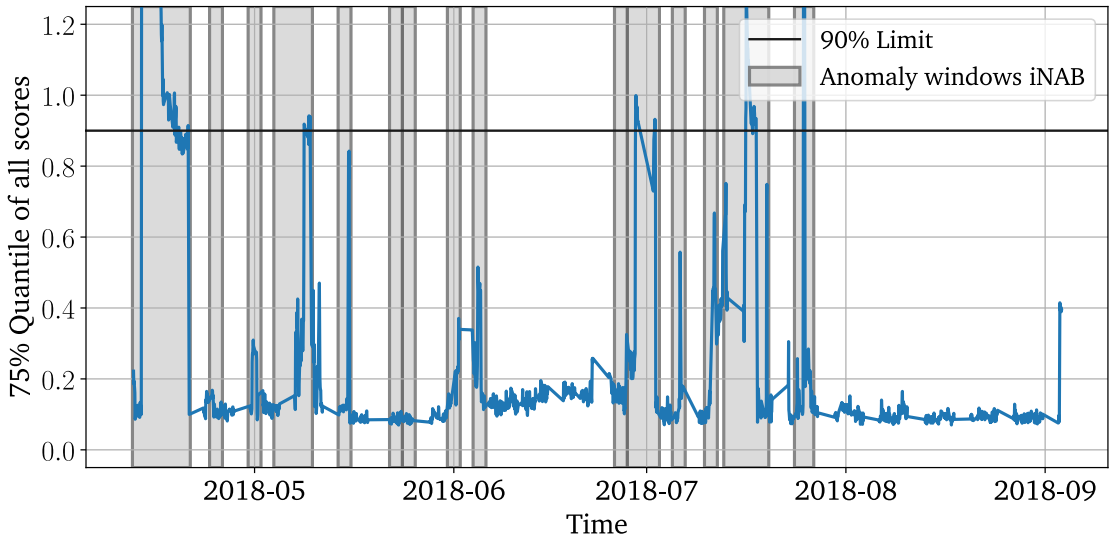


Figure 5.5: The anomaly windows expand all documented and data-driven error days to the previous production day. Overlapping anomaly windows are summarized to one anomaly window.

of anomaly windows is just increased by one to 15. As illustrated in Figure 5.5, the windows span between two and seven days.

Definition Weighting Factor

The weighting factor is also determined equivalently to the NAB score. By adding the data driven error cases, the total length of anomaly windows increases to about $25\,\%$. This leads to slight adjustment of the weighting factors to $A_{TP} = 1.0$, $A_{FP} = 0.053$ and $A_{FN} = -0.053$.

5.4.3 Comparison NAB and iNAB

To ensure the functionality of the iNAB score, the three proposed improvements and their interplay are checked in the following based on the data of this study.

For each algorithm, the anomaly probabilities of the ten reference states are combined to a group, normalized, and the NAB and iNAB scores are calculated for each anomaly probability with the above determined parameters for a variety of thresholds. The introduced parameter N for the iNAB score (length of subsequent anomalies needed for a true-positive) is set to $N = 1,\ 2,\ 4$. For each threshold, the minimum NAB/iNAB score per group is taken, in order to receive the score for the worst performing reference. In this way, if the algorithm fails for one reference, the whole algorithm is scored worse. The maximum score per group is taken. In

order to reduce complexity in the evaluation, all scores with active noise reduction are ignored for the comparison. Evaluations in the next section will show that the influence of denoising is negligible in this study.

As proposed in Section 4.6.2, three random generators - drawn from a normal, a gaussian, and an exponential distribution - are added as artificial references. For each distribution, ten samples are drawn and processed accordingly to the data-driven algorithms.

All results can be found in Table 5.6. The two additional improvements are analyzed step-by-step in the following.

Performance of original NAB score (NAB, N=1)

The original NAB score can be found in the first column in Table 5.6. Two obvious observations can be made:

1. All data-driven scores achieve NAB scores around zero. This implies that those algorithms perform in average as well as a random generator, which places $15\,\%$ anomalies at random positions. Thus, it seems, none of the algorithms is performing well.

2. The only score, which achieves a NAB score significantly above five, is the gaussian random generator. This destroys the meaningfulness of the score, as all data driven algorithms are outperformed by a random generator.

Consequently, the NAB score does not provide a useful score for comparing different algorithms.

Improvement 1: Add data-driven error cases (iNAB, N=1)

As first improvement, the documented error cases are expanded by the data-driven error cases. It is clearly visible that most data-driven scores improve their performance significantly to a score above 20. Additionally, larger differences between algorithms appear. For instance, the euclidean distance of the Fourier transform performs well with a score of 26.9, whereas the Mahalanobis distance of the tsfresh-package fails with a score of 0.

Nevertheless, the gaussian random generator still performs best with a score of 39.3. As well, the exponential random generator is in the upper half with a score of 25.1. Thus, different data-driven algorithms can be compared with this improvement. However, random generators still manage to perform similarly well.

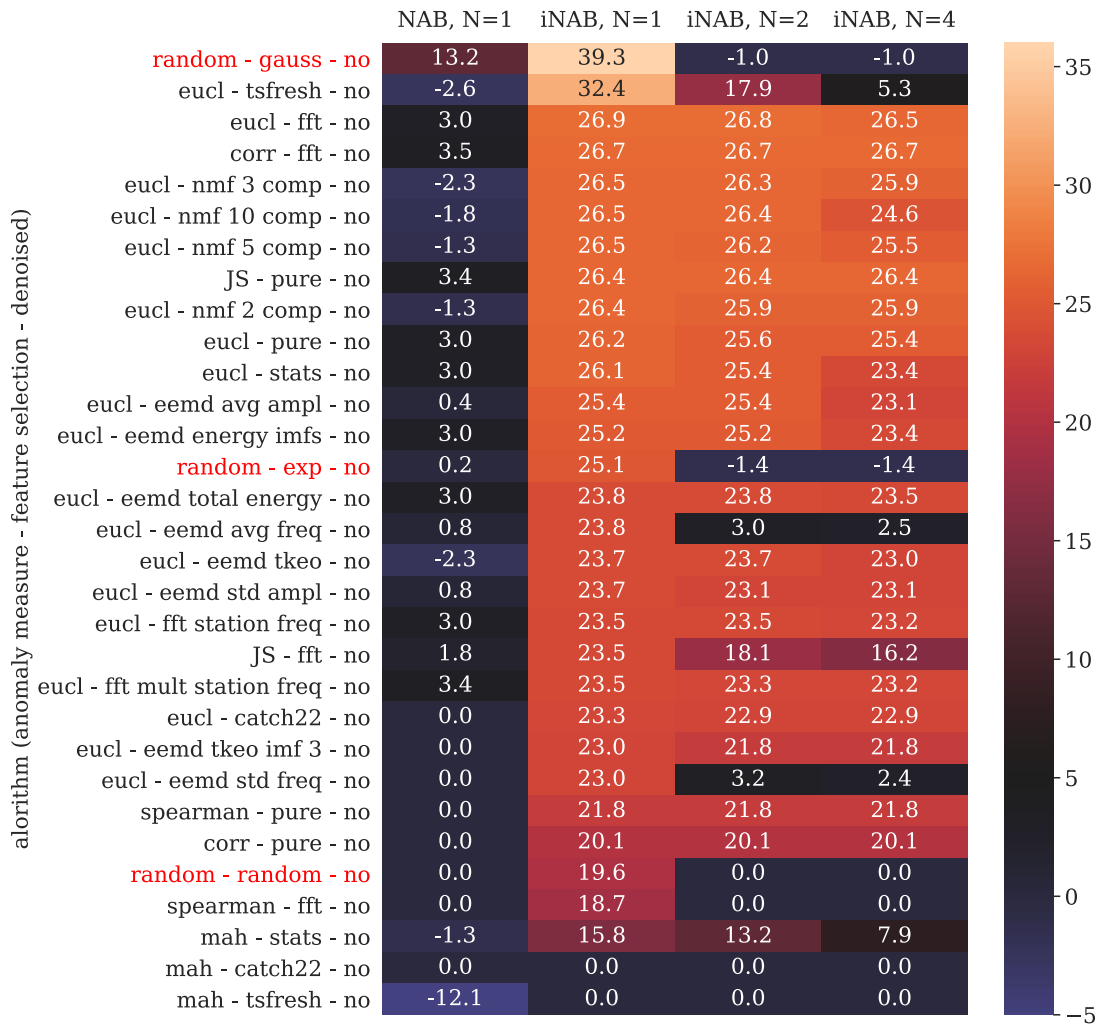


Figure 5.6: Comparison of the NAB score with the iNAB scores with $N = 1,\ 2,\ 4$. Three random generators (printed in red) are added to a set of data-driven algorithms (details see in Table 5.1). The original NAB evaluation (column 1) results in scores around zero, which implies a similar performance as an average random generator. By adding data-driven error cases (column 2), the result of all algorithms improve significantly to up to 26.8. Increasing N to $N = 2$ (column 3), the artificially added random generators can be clearly separated from the other scores. The impact of increasing to $N = 4$ (column 4) is limited.

Improvement 2: Increase N (iNAB, N=2; iNAB, N= 4)

As second improvement, detected anomalies are only considered as true-positives if N anomalies are detected in a row. The parameter N is set to $N = 2$ and $N = 4$ in the following.

The increase of N has a large impact on the random generators, as all three scores are decreased to values around zero. Additionally, other data-driven scores are also substantially reduced, for example the average of the instantaneous frequency of the EEMD from 23.8 to 3.0. Thus, it seems, more algorithms acting like random generators, can be identified. All other scores seem to be rather unchanged with the increase of N.

A further increase from $N = 2$ to $N = 4$, decreases most of the scores slightly, as it favors stable anomalies. This leads to slight changes in the order of the high-score, but with few exceptions the central interpretation stays the same.

Summary

Summarized, the combination of adding data-driven error cases and increasing N leads to a significant increase of meaningfulness of the scores. Random generators can be detected reliably, and data-driven scores show significant differences in performance. A further increase from $N = 2$ to $N = 4$ has only a limited impact.

5.5 Evaluation Anomaly Algorithms

In this section, the 44 different combinations of features and anomaly algorithms are evaluated. For that, a set of evaluation scores is defined, which focus on the stability and usability of the methods. The evaluation is performed in several rounds, only the best performing algorithms continue into the next round. The winning algorithms will be investigated in further detail and compared to a physical based score.

5.5.1 Evaluation Scores

In the original paper of the NAB score [70], the maximum NAB score is taken as the final measure for comparing different anomaly algorithms with each other. This is an accepted academical way to show that a new algorithm is performing better than already existing ones. In order to be able to use the score in the one-shot real-life applications, other considerations are at least equally important in order to receive a reliable result:

1. Insensitivity to small variations of the healthy state
   As already mentioned above, over time, slightly different states can be marked as healthy states. The anomaly measure should perform insensitive on those variations of the reference state and generalize successfully.

2. Broad boundary between normal and abnormal
   All scores return a continuous probability or distance measure as result, which first needs to be transformed into a binary normal-abnormal signal. This transformation is tricky as it has to be fixed at the beginning, when the shape and intensity of error cases is not yet known. Thus, it is essential that the chosen anomaly method gives a similar result for a wide range of transformations. The easiest transformation is by setting a threshold. Anomaly methods, which reach a high iNAB score for just a very specific threshold, cannot be transferred to a different star with achieving a reliable result.

3. Limited Resources
   The computational power is limited, as the calculation runs on an very resource limited edge device. Other jobs on the edge device are not allowed to be disturbed.

In order to fulfill all three aspects, the iNAB scores are preprocessed in a particular way, and different characteristics of it are extracted.

The first criterion is already handled by the preprocessing procedure. For each of the 44 algorithms, the anomaly probabilities for the 10 different reference states are combined to a group, and normalized to its 99.5 percentile, taking all time points into account. This enables comparability of the different groups without overestimating peaks, in which the algorithms are overly confident (see Figure 5.7 top). For each of the algorithms, the iNAB score with $N = 2$ is calculated in dependence of the binary normal-abnormal threshold. This results in 10 iNAB score curves per group. As the focus lies on the insensitivity to small variations of the reference, each group is characterized by its minimum curve. This implies, for every threshold, the smallest iNAB score is taken in order to receive a baseline accuracy. This minimum curve serves as the basis for further scores, and is visualized in Figure 5.7 bottom (dark blue).

Resource Usage The first criterion is the resource usage. A full comprehensive test on an edge device is difficult to perform, as the load and the settings of the edge device differ over time and line setup. In order to achieve a rough estimation, the evaluation time is measured on a laptop (details see Section 2.6) and averaged over at least 30 samples. As all calculations are performed on the same laptop, a comparability between the algorithms is ensured. A

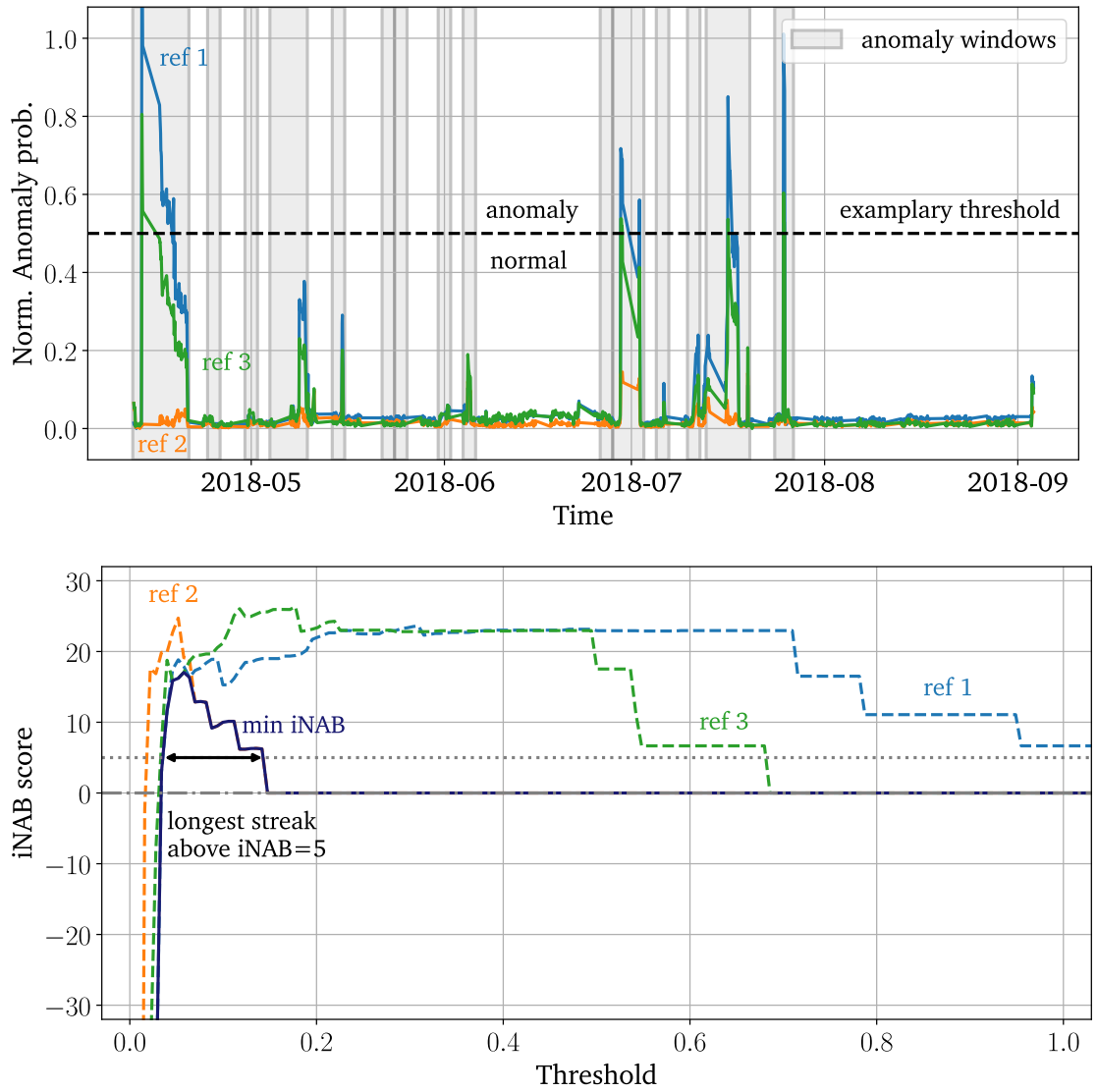


Figure 5.7: Visualization of the evaluation scores on an exemplary anomaly algorithm applied on three different reference states.
Top: For each reference state, the anomaly probability over time is calculated, and the group of the three anomaly probabilities is normalized by its 99.5 percentile. Comparing the probabilities with the documented anomaly windows, the algorithm performs decent with being trained on reference 1 (blue) and 3 (green), but fails with reference 2 (orange).
Bottom: For each threshold (separating anomaly from normal, see example top) and reference, the iNAB score is calculated. The minimum iNAB score (dark blue) is determined and serves as base-line accuracy for all further scores. In this example, reference 1 and 3 achieve iNAB scores above 20 for a wide range of thresholds. Nevertheless, due to reference 2, the minimum iNAB score does not perform very well with a 75% quantile of zero (equals random generator) and a streak above five with the length of about 0.1.

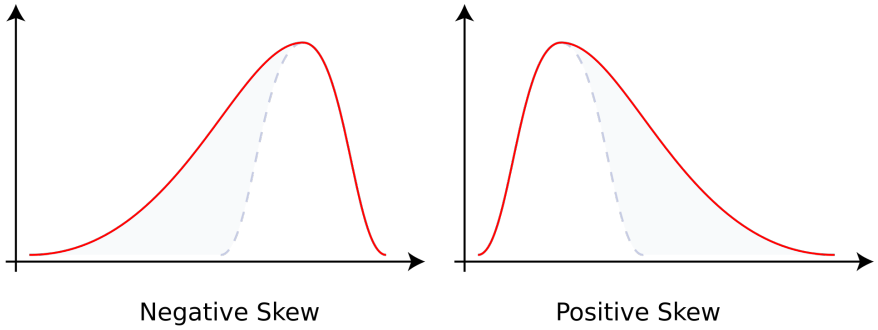


Figure 5.8: Visualization of Skewness [51]

calculation time below 10 seconds is acceptable, below one second would be favored.

Longest streak of iNAB score above five The second criterion is the longest streak with iNAB score above five. This provides a base-line criterion for the broad boundary between normal and abnormal. The number five is chosen to be slightly above zero, in order to allow distinction from random generators.

75% Quantile of iNAB score The third criterion is the 75% quantile. This measure provides a similar information as the maximum, with the difference that the broad boundary between normal and abnormal is also taken into account.

Skewness of iNAB score As last criterion, the statistical measures skewness is used in order to classify the shape of the iNAB score. The skewness is defined as the third standardized moment:

$$s = \frac{\mathrm{E}\left[(X-\mu)^3\right]}{\sigma^3} \tag{5.1}$$

with μ being the mean, σ the standard deviation and E the expectation operator. The measure is particularly suitable here as it quantifies the asymmetry of a distribution (as shown in Figure 5.8). The goal is to find an algorithm, whose iNAB score consists mainly of high scores and has thus a strongly negative skew. For the calculation, just iNAB values above five are considered, in order to ignore the tail for large thresholds.

In the following, the four measures are evaluated one after the other. After each evaluation, just the best algorithms are allowed to proceed.

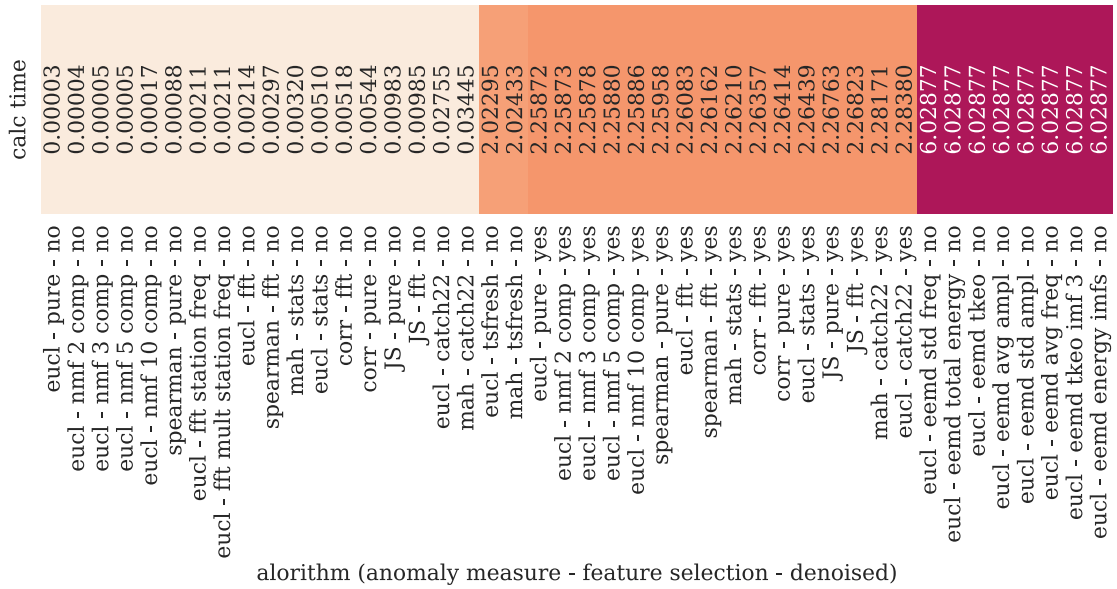


Figure 5.9: Calculation time for all different setups ordered by size (Round 1). All setups stay underneath a calculation time of 10 seconds.

5.5.2 Evaluation

Round 1: Resource Usage

The scoring time is averaged over at least 30 samples in order to receive a reliable result. As visualized in Table 5.9, all scores range from $3.0 \cdot 10^{-5}$ sec to about 6.0 sec. Thereby, all calculations based on the EEMD (including all denoised measures) or the tsfresh-package take significantly longer than the other algorithms. As all algorithms manage to finish their calculation within 10 sec, none of the algorithms is already excluded in this round. Nevertheless, in case algorithms perform similarly well in future rounds, the algorithms with shorter calculation time will be favored.

Round 2: Longest streak of iNAB score above five

The longest streak of the iNAB score above five provides several sets of information. First, a length equal to zero implies that the maximum iNAB score is below five. Thus, those algorithms can be discarded as they behave similarly to random generators. Secondly, very short streaks lead to the conclusion, that a result better than five can just be achieved for very specific thresholds. Even if the maximum was incredibly high, those algorithms are not interesting, as hitting this small window without any error data is rather impossible when training a new machine.

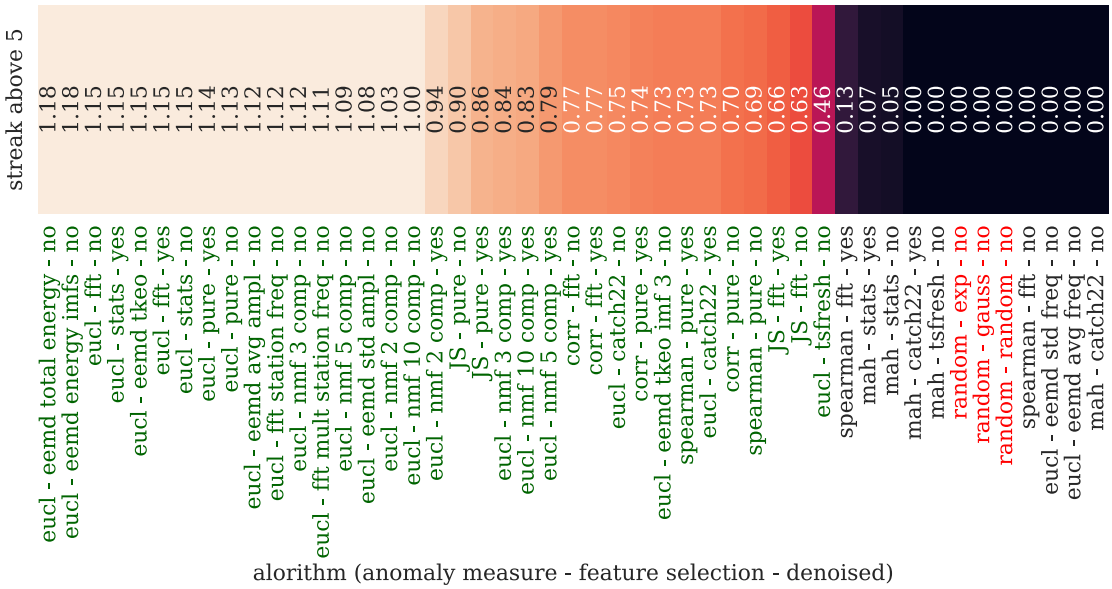


Figure 5.10: Result of the longest iNAB streak above five for all algorithms (Round 2). The three artificially added random generators are marked in red for reference. Most algorithms perform very well with streak lengths above 0.4. All scores marked with green proceed into the next round.

Thirdly, it is possible to achieve streaks even longer than one. As the groups are normalized to its 99.5 percentile, anomaly probabilities above one can appear. Here, the threshold is swept between 0 to 1.2, resulting in a maximum possible streak of 1.2. Algorithms close to the maximum streak length are extremely sure about few specific anomalies. This can be a good sign, but does not allow any conclusion for the overall performance.

The results of this round can be found in Figure 5.10. Most of the algorithms perform very well with streaks above 0.4. Examining the low-performing algorithms, specific patterns can be detected. All algorithms with the Mahalanobis distance (shortened in the Figure to “mah”) as anomaly algorithm fail independently of the preprocessing. To get a better understanding of this phenomenon, scores of the statistical features evaluated with the Mahalanobis distance are depicted in Figure 5.11 for different reference states. One can clearly see that the three scores perform very differently. Whereas reference 4 (orange) is performing very well, reference 2 (blue) fails. Thus, the Mahalanobis distance reacts too sensitive on the reference state or its surrounding, which are needed for the covariance matrix.

Evaluating the results of the EEMD, most scores perform very well with the exception of the frequency evaluation (eemd freq std, eemd freq avg). This could have been expected, as the goal of the IMF matching algorithm (Section 4.6.1) is

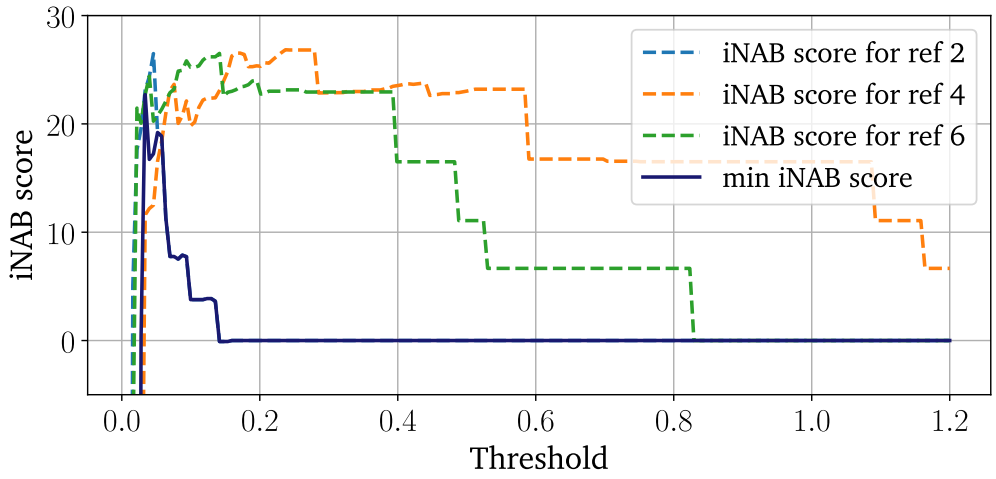


Figure 5.11: Exemplary iNAB curves of the Mahalanobis distance applied on the statistical features for three different reference curves. The different references lead to completely different results, resulting in a low minimum iNAB scores for the algorithm. Thus, the Mahalanobis distance can achieve very good results, but it is strongly influenced by the chosen reference.

to match similar frequencies into the same IMF. Thus, it is a sign that the proposed algorithm is performing as expected.

Finally, the Spearman rank-order correlation of the Fourier transform performs badly. Considering that the Fourier transform is mainly around zero for most frequencies with just a few specific peaks, it is not surprising that a rank-order score has trouble handling this kind of distribution.

At the end of Round 2, nine algorithms are excluded, with leaving 35 algorithms for Round 3.

Round 3: 75% Quantile of iNAB score

In Round 3, the 75% quantile of the iNAB score is examined. It provides information about the performance of the algorithms for the TOP 25% iNAB scores, which corresponds with a threshold range of 0.3 (as the maximum threshold is set to 1.2).

The results in Figure 5.12 show that a lot of algorithms perform similarly well, with one clear winner (Pearson correlation of the Fourier transform) and one clear looser (euclidean distance of tsfresh features). Denoising the data has a negligible influence on the result. As consequence of Round 1, all setups with noise reduction are ignored in the following, in order to keep the calculation time low.

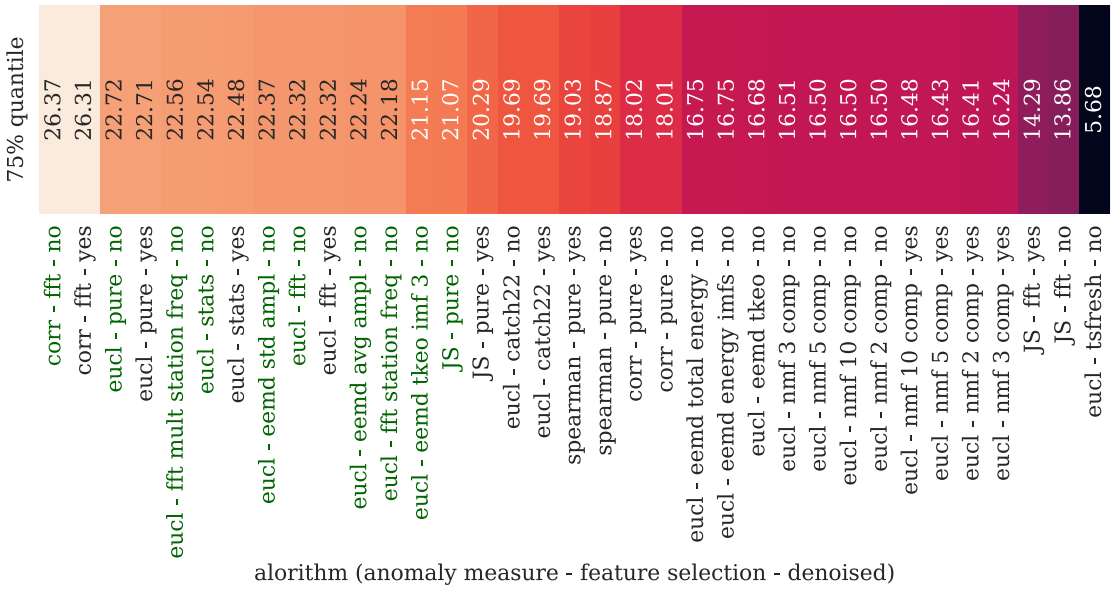


Figure 5.12: Result of the 75% quantile for all algorithms in Round 3. The TOP 10 algorithms without noise reduction (marked in green) will continue into Round 4.

In terms of global features, the tsfresh package performs worst with 5.68, catch22 reaches the center field and the statistical measures reaches the TOP10. Here, a final conclusion about the packages must be taken with caution. As all of them are evaluated with the euclidean norm, some features may be over- or underrepresented.

In the next round, the TOP 10 algorithms (ignoring the ones with noise reduction) will continue (marked in green). Surprisingly, those consist of a broad mixture of different algorithms. Frequency based algorithms are as prominent as algorithms based on the unprocessed curve, or algorithms based on the statistical features. Thus, it seems like the error patterns show very distinct characteristics in a variety of extracted features. The only algorithm that drops out collectively is the NMF. Though being in the TOP 5 algorithms of maximum iNAB scores (see Table 5.7 column 3), all setups based on the NMF can be found in the lower half for the 75% quantile, independent of the chosen number of components. Thus, the NMF detects anomalies successfully, but performs worse in separating the normal from the abnormal state.

Round 4: Skewness of iNAB score

The last round is based on the skewness, which measures the asymmetry of the iNAB scores. The lower the skewness, the more values can be found in the range

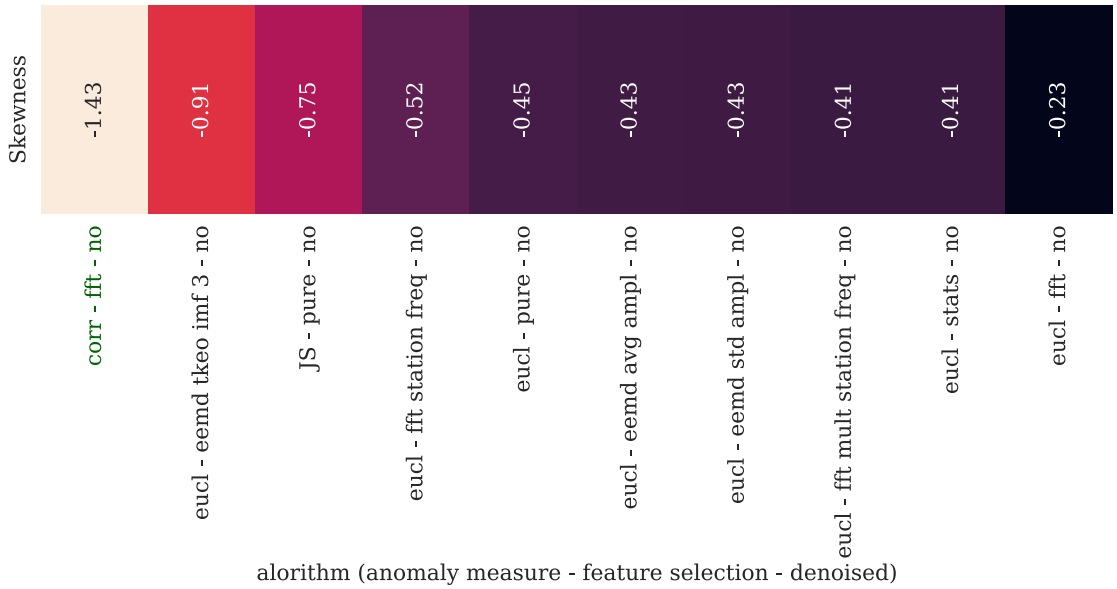


Figure 5.13: Result of the skewness for all algorithms in Round 4. All algorithm yield a negative result with the Pearson correlation of the Fourier transform as the winner.

of high iNAB scores. It is evaluated for the iNAB scores above five, in order to ignore the low-iNAB score tails for large thresholds.

Figure 5.13 shows the results of the TOP10 algorithms of Round 4. All of them yield negative values, leading to the wanted behavior. The winning algorithm is the same as in Round 3: the Pearson correlation of the Fourier transform. In order to gain a better understanding and validate the result, the TOP 3 iNAB scores are plotted in dependence to the threshold (see Figure 5.14). One can clearly see that the Pearson correlation of the Fourier transform (blue) reaches the highest iNAB score, and keeps it for a large range of thresholds. In contrast, the Jensen-Shannon divergence (green dash-dotted) just peaks shortly to a high iNAB score, though keeps the slightly lower score for a broader range of thresholds. Consequently, the winning method does not win every round of this study, but finds a balances between the different requirements.

Conclusion

In this study, a large variety of algorithms were compared to each other by different stability measures. Surprisingly, with a maximum iNAB score of 26.8, none of the algorithms could detect all or even most of the documented error cases reliably (which would yield to a iNAB score of 100). It seems like some error cases are not represented in the data, either they were detected and fixed right after

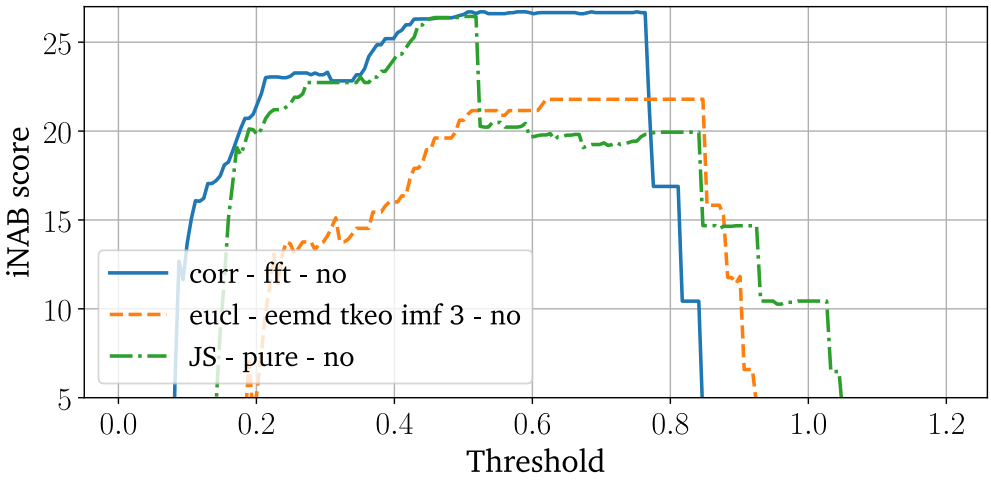


Figure 5.14: iNAB score in dependence of the threshold for the 3 winning algorithms of Round 4.

emerging, or they had an origin that does not lead to any characteristics in the data. The study yields a clear winner with the Pearson correlation of the Fourier transform. Interestingly, the physically motivated score with taking the station frequency in the Fourier spectrum as anomaly measure manages a good spot in the TOP4 algorithms.

In the next section, the winning score and the physically motivated score will be investigated in further detail. Afterwards, the winning score is applied to data of different bottlers in order to evaluate the idea of transfer learning.

5.6 Compare Winning Method with Physical Method

The physical score and the winning score of the study are based on very similar features, both relying on the frequency spectrum, with the physical score only examining the station frequency, and the Pearson correlation taking the whole spectrum into account. Due to the good performance of the physical score, it seems that most error cases can be detected by just considering the station frequency. Nevertheless, the Pearson correlation manages to extract additional information resulting in a better score.

In order to investigate the two scores, the anomaly probabilities over time averaged over all reference states are depicted in Figure 5.15. Comparing the anomaly probabilities with the documented and data driven anomaly windows, the win-

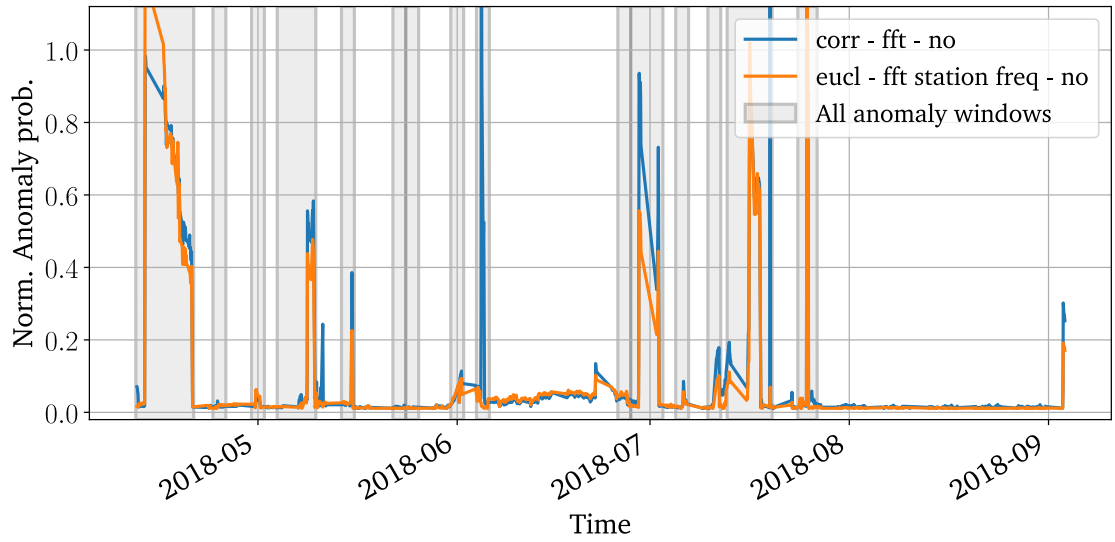


Figure 5.15: Comparison of the winner algorithm (Pearson correlation of Fourier transform) with a physical motivated algorithm (Station frequency of Fourier transform). The two scores behave in a similar manner with exceptions on June 4th and July 19th.

ning algorithm detects six of the 15 error cases reliably (probability above 0.5), one with a probability of 0.4, and one with about 0.2, leading to a detection rate of 46.7%. Thereby, the algorithm detects four error cases significantly before spotted by the operator, some even several days ahead. Even if not all error cases are detected, it is a good sign that error cases can be detected ahead of the on-site people, and thus the approach provides useful information in the field of "Predictive Maintenance". The physical algorithm detects most error case at exactly the same times with slightly different probabilities: It recognizes four error cases reliably, one with 0.2, and one with 0.1. Major differences between the scores can be found on June 4th and July 19th. Those error cases are detected by the Pearson correlation with a very high anomaly probability, but are missed completely by the physical approach.

For examining this observation, in Figure 5.16, one anomalous angle pattern is chosen that was detected by both algorithms (April 17th, 07:00), and one that was just detected by the Pearson correlation (June 4th, 18:00). The pattern, which was detected by both algorithms (top left), shows the characteristics of a synchronization error, with a strongly enhanced amplitude of the station frequency in the frequency spectrum (top right). The frequency here is measured in the unconventional unit "peaks per round" as it allows an easy translation to the mechanical properties of the machine. 26 peaks/round matches with number of stations and the theory of Chapter 3. The second error case on June 4th shows a completely

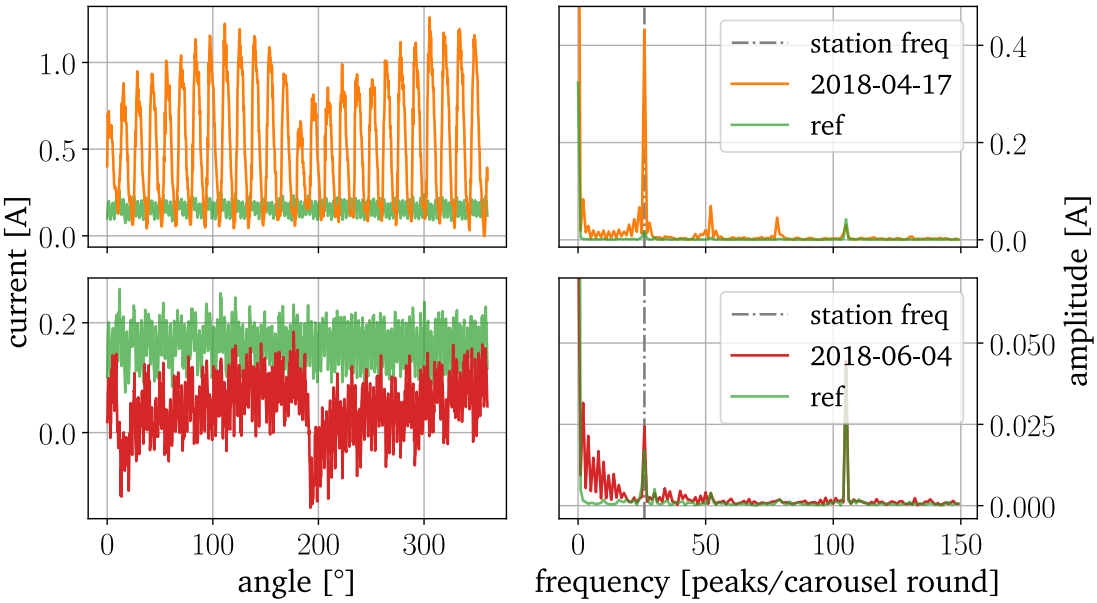


Figure 5.16: Two error cases detected by the algorithms, each as angle pattern (left) and Fourier spectrum (right). Top: Characteristic synchronization error with strongly enhanced station frequency. Bottom: Error pattern with saw-tooth shape, changing the frequency pattern with leaving the station frequency almost unchanged.

different pattern (bottom left). The pattern has a saw-tooth shape with a by factor five smaller amplitude in comparison to the first error case. In the Fourier spectrum (bottom right) a broad band of frequencies are slightly enhanced, with keeping the station frequency roughly the same. Thus, the winning method outperforms the physical measure for this error case, as it detects changes in the pattern in a sensitive way, which are not represented in the physical model.

As a consequence, the physical model based on insights of Chapter 3 performs very well for synchronization errors, but fails for error cases, which produce a different shape. In contrast, the winning algorithm of this study is much more sensitive to other shapes and generalizes very effectively. Thus, both approaches, the physical simulation and the generalized anomaly approach, show their advantages. The physical score provides a reliable result for the simulated error case, and allows an instantaneous specification of the error case, but fails to detect any other anomalies. The generalized anomaly approach detects all different kinds of anomalies, but doesn't allow any interpretation. Consequently, the combination of anomaly detection with specialized models enables the detection of all kinds of anomalies with providing additional information to some of them.

5.7 Transfer Learning to other Bottlers

As final step of this study, the generalization of the winning algorithm to further stars and bottlers is checked. It is very essential that the one-shot semi-supervised anomaly detection approach can be easily transferred to a new star, with just one healthy sample in the training phase, and no available error cases of that new star. Thereby, the challenge of the transfer consists of two parts: First, does the algorithm in general detect anomalies reliably, and second, how to set the limit to differentiate normal from abnormal.

Here, the first part can be only evaluated in a sample-based qualitative way. Though having data of other stars and production sites available for several months, no or not sufficient error documentation for the time windows exist. Thus, a random set of detected anomalies can be examined by an expert, but the number of missed error cases cannot be quantified as the ground truth is missing.

For the second part, the Pearson correlation coefficient has the advantage that it already scales the result between -1 and 1, and thus limits of the study could be transferred directly to different stars. Analyzing the comparative study, normal and abnormal can be easily separated: Correlation coefficients above 0.98 yield the classification “normal”, coefficients below 0.965 “anomaly”. In the group of anomalies, most anomalies can be found between 0.90 and 0.965, with three exceptional anomalies with correlation values between 0.73 and 0.85. These limits serve as orientation for the transfer to further stars.

Three stars of two additional breweries B and C are used for the transfer learning test. Different locations in the machine are chosen in order to generalize the test in a further dimension. Star 1 of bottler B is an infeed wheel (equivalently to the star in the study), star 2 of bottler B is positioned after the filling carousel, and star 3 of bottler C after the crowner. For each star, a reference curve is chosen by an expert. As depicted in Figure 5.17, these reference states differ from each other by frequency, amplitude, mean and general pattern, as various machine settings influence the reference curve. This reinforces the concept of transfer learning, as a reference state of one star cannot be used for a different star. For testing the behavior, the correlation coefficient of the Fourier transform is calculated for each star for an interval of more than 7 months of data. In order to investigate the behavior of the stars, for each star three samples are drawn randomly, one in each of the above correlation windows. Additional - if existent - an additional curve with a correlation coefficient below 0.73 is chosen.

In Figure 5.18, every column compares the chosen anomaly curves with its reference. In the first row, curves with correlation coefficients above 0.98 are

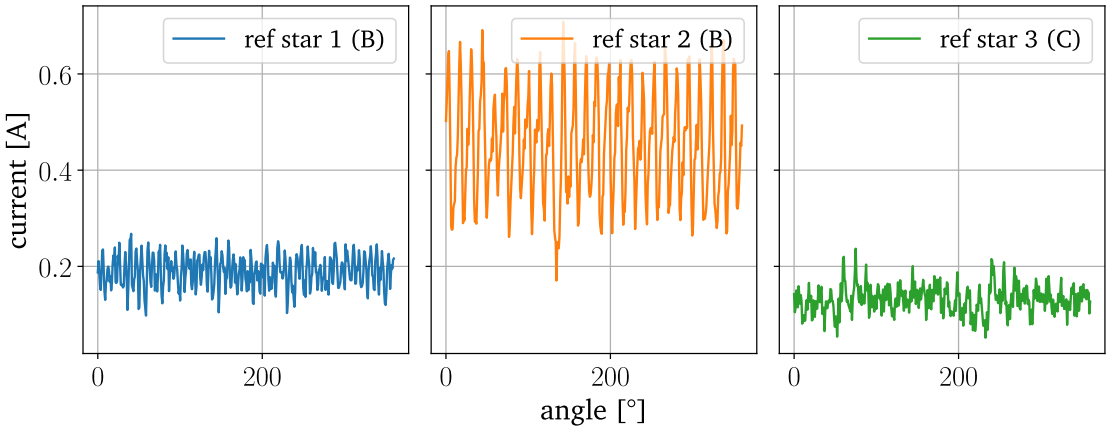


Figure 5.17: Reference states for three stars of two breweries B and C.

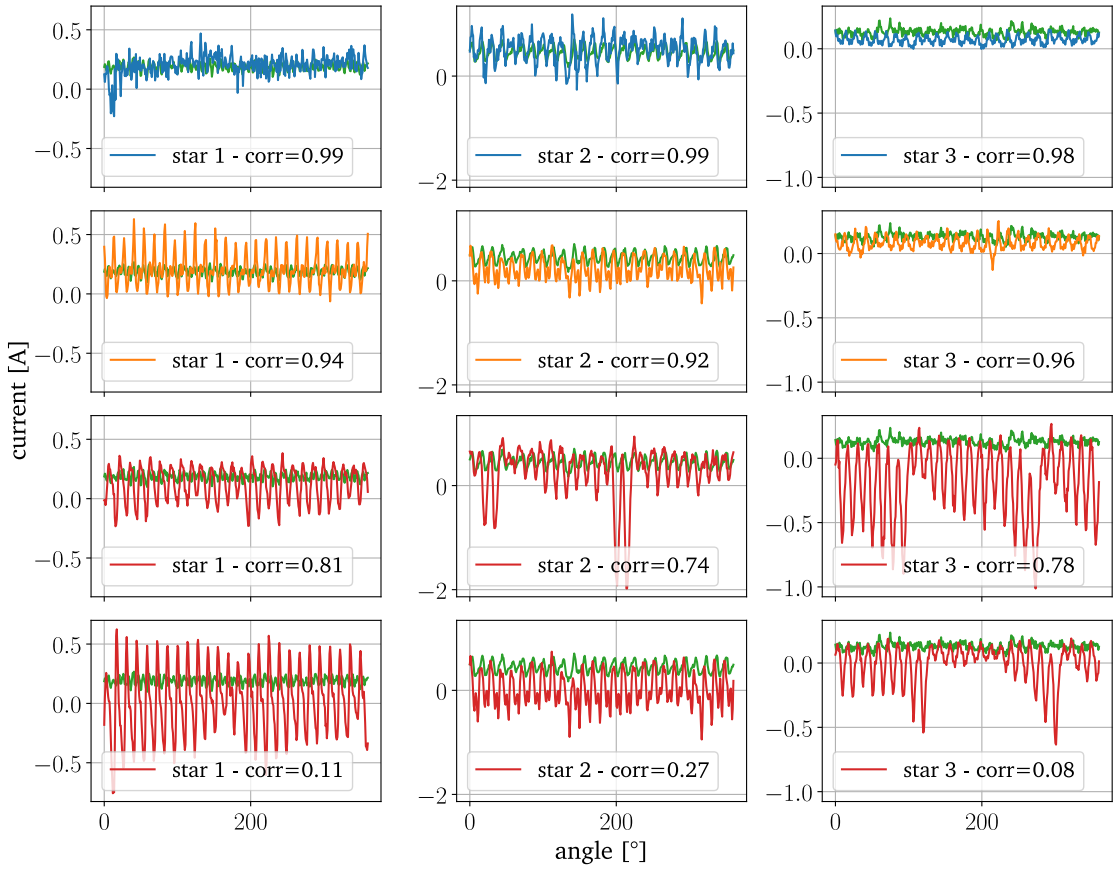


Figure 5.18: Four exemplary curves for each star with different correlation coefficients, compared to the reference state (green).

depicted. According to the comparative study all of those curves should qualify as 'normal'. Looking into detail, each of the curves deviates very slightly from its reference state. Discussion with an expert, those small deviations are normal and should not be detected as anomaly. Thus, the first check is successful for all three stars.

In the second row, the curves with a correlation coefficient between 0.90 and 0.97 should represent anomalous curves. Analyzing the first column, the classification as anomaly is justified, as the amplitude is strongly increased. In the second column, the amplitude is also increased, but not as strong as in the first column (in comparison to the reference). In the third column, the curve just shows small deviations, which are - according to the expert - not necessarily strong enough for the anomaly classification. Though, further knowledge about the machine would be needed for a clear assessment, it seems that the limit is set too sensitive for this star.

The third and forth row show patterns with correlation coefficients below 0.81. All of those patterns can be clearly classified as anomaly by eye. In each case, the pattern deviates strongly from the reference. Interestingly, for all three stars, curves with correlation coefficients far below 0.73 exist.

Summarized, the qualitative test for transfer learning was successful. The detected anomalies of the winning algorithm could be confirmed as anomalies by eye and expert. However, the limit between normal and abnormal can vary from star to star, as the anomaly limit of 0.965 seems to be too strong for star 3. In addition, for all three stars, curves with correlation coefficients far below 0.73 were found. Thus, at this stage, a star-based fine adjustment of the limits seem inevitable. In order to improve the limit setting, the comparative study would need to be extended to further fillers in other production lines. At the moment, the proposed limits of the comparative study provide a good starting point with already high accuracy for most stars. Depending on the feedback of the people on-site, adjustments of the limits may be necessary.

5.8 Discussion and Outlook

In this chapter, a large comparative anomaly study was performed on the data of angle-current curves. Each anomaly setup consisted of a feature extraction method and an anomaly algorithm. The number of possible setups was strongly reduced by the restriction of just one healthy sample in the training data. Most anomaly algorithms need a larger number of healthy samples. Here, algorithms

for the minimalistic one-shot semi-supervised anomaly detection approach were searched.

In order to compare the result of the different setups, the iNAB score (developed in the previous chapter) was used. Before using it as score, its functionality was tested and compared to the NAB score. The suggested improvements increased the robustness and reliability of the score significantly.

Based on the iNAB score, different empirical scores were developed, which allowed a multilayered evaluation of the algorithms. Due to the one-shot approach, characteristics like stability over different reference states, or large separation between normal and abnormal states were considered, and rated. The 44 different anomaly setups yielded a clear winner: the Pearson correlation of the Fourier transform. In the frequency space, the anomalies were separated best from the normal state, and a broad spectrum of different anomalous patterns could be detected. Seven of the 15 documented error cases could be detected ahead of the people on-site, some of them even days ahead. Interestingly, a physical score based on Chapter 3 also managed to be one of the four best performing algorithms. The main disadvantage of the physical score was the specialization on one specific error case, and thus, missing two error cases with different characteristics. Hence, the physical model has the strong disadvantage that all possible error cases have to be modeled, whereas an anomaly algorithm is able to detect a large variety of (even unknown) error cases. Nevertheless, the anomaly detection algorithm and the physical model can complement each other well. Whereas the anomaly detection algorithm is able to detect different kinds of anomalies, the physical model can classify some of the anomalies and provide a specific error description.

As a last step, the concept of transfer learning was tested on the winning algorithm by applying it to three other stars of other breweries. Due to missing labels, it was performed on a sample-based way. The algorithm was proven successful in detecting a variety of different anomalies, which could be confirmed by an expert. In setting the limits between normal and anomaly, the different stars yielded slightly different results. Thus, a fine adjustment routine is needed following the training phase.

As a next step, it would be worth considering multi-algorithm approaches, in which several algorithm “vote” for anomalies. This approach could lead to a solution with no need of star-specific fine adjustments. This approach was so far not considered in order to keep the scoring time low. With using a variety of algorithms with very short scoring times or similar preprocessing, this approach should be still doable.

Additionally, using further information could improve the detection rate of 47%. As some error cases originate in electrical or network faults, information

of error messages or response times could be taken into account. This could lead to the detection of anomalies with a larger variety of origins.

This study also yields a research topic in a completely different field of science: the human-machine (or here human-algorithm) interaction in Predictive Maintenance use cases. Though not explicitly described in this study, several limitations appeared in the communication with the machine operators. The fine adjustment and further improvement of the algorithms is only possible in an automated communication process, in which the operator provides the necessary, and even more important, correct information to the algorithm. Thereby, the main challenge poses that anomalies appear on an irregular basis (sometimes once every couple months) with no explicit instruction as the origin is unknown. Additionally, they appear at times at which the machine is still able to produce at high-speed, even if the production at those times reduce the lifetime strongly, and a sudden break-down is very likely. An intuitive self-explaining feedback-loop completely integrated in the daily schedule of the operator would be a research topic on its own.

6 Physics- and Expert-Driven Error Sketch Recognition

One of the main reasons for the lack of almost any machine learning approaches in the bottling industry is the missing labels of error cases. Even if an error case happened, it is usually not documented in a standardized way and the knowledge about the event often gets lost on the way to the Data Scientist. Additionally, in case error cases are documented, the same error cases happen rather rarely, and in order to develop a generalized model, error data from several production lines are needed. This drastically limits the possibilities for supervised machine learning in this industry.

In order to tackle the challenge of missing labels and error data, several approaches have been tried. Next to anomaly detection, a common approach is to simulate the behavior of the machine. The simulation, e.g. finite element based, models the machine and different error cases, with taking all available physical knowledge into account. This approach seems promising in some cases, but also comes with some drawbacks. First, the simulations are often complex, and thus need a lot of programming and calculation time. Second, in order to gain reliable error patterns, simulations must to be tuned on the explicit machine data and also error data. This implies that every model has to be adapted to the specifics of that machines and need to be rerun. Thus, simulation approaches can give a jump start, but the portability from one bottler to the next comes with high cost of resources.

In this chapter, we aim to establish a completely new approach, called Physics- and Expert-driven Error Sketch Recognition (PEESR):

1. We use the knowledge of experts in order to create error data. Not with time-intensive simulations, but by sketching their gut feeling and pre-knowledge on what error cases usually look like.

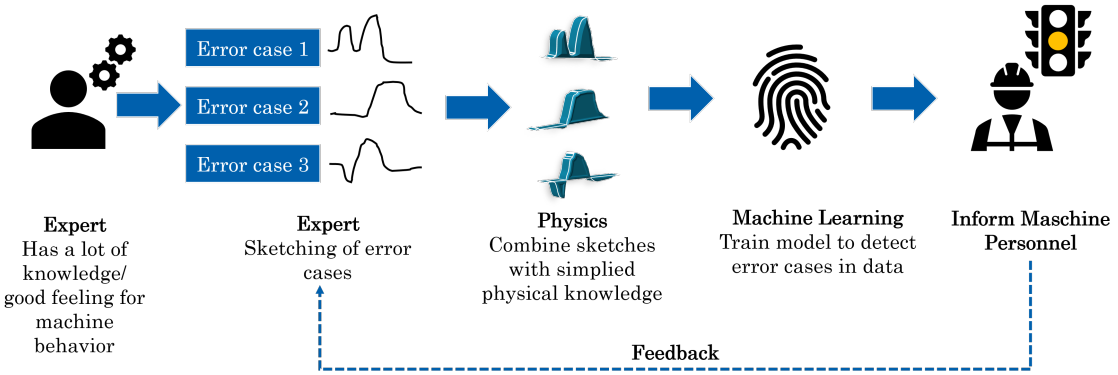


Figure 6.1: High-level principle of physics- and expert-driven error sketch recognition.

2. These sketches are combined with simplified physical models (like from Chapter 3) and transferred into computational curves.

3. The sketches are the basis for detecting and recognizing the error cases in real data.

This approach allows to detect and classify error cases, even if no error data is available. Additionally, sketches support transfer learning naturally. Different sizes of the machine or different positions of handovers can be adjusted easily by adjusting the sketch. As sketches simplify reality, they cannot be overfitted to one machine, but have the potential to generalize over different production lines and sites naturally. However, it is important to note that the success of this approach strongly depends on the quality of the sketches. It cannot be ruled out that error cases show different behavior than expected, potentially leading to a misinterpretation. Thus, the feedback of the machine operator is very essential in order to improve the sketches. This also has the special effect that the experts can learn more about their machines.

In the following chapter, we will test this new approach on the detected anomalies of Chapter 5 while taking the physical knowledge of Chapter 3 into account.

6.1 Literature Research

The following chapter provides a brief overview of PEESR in literature. First, research fields with similar challenges are identified and introduced. Second, different algorithmic strategies for solving such a challenge are presented. Third, a promising algorithm is introduced, which has proven successful in similar challenges.

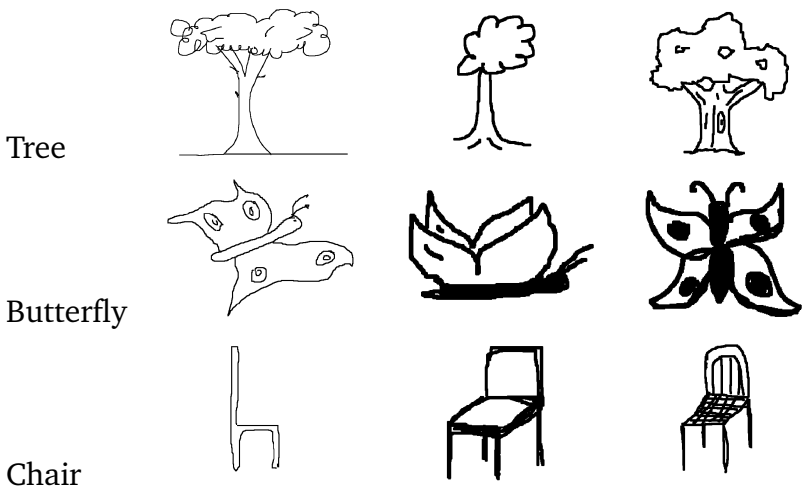


Figure 6.2: Three people were asked to sketch the same three word. One can clearly see the huge variation in between the freehand sketches.

6.1.1 Assigning to Research Field

The approach of recognizing error cases by physics and expert based sketches cannot be assigned to one specific field of research, but rather to the combination of several. It shares some specifics with the hot research topics of one-shot learning, sketch recognition and physics-guided machine learning. All of those topics are still extensively growing research fields.

- Sketch or pattern recognition:
  Computerized sketch recognition is widely spread in daily life with ever increasing importance. Freehand letters and symbols drawn on computer screens are directly transformed into text, geometric patterns or even circuit diagrams [131, 129, 52]. Cameras recognize hand gestures [20] and suspects can be identified from forensic facial sketches [89].
  All these topics can handle simplifications of reality or huge variations in the patterns, as depicted in Figure 6.2. In order to meet those requirements, most models are trained on extensive data sets of either the patterns which should be recognized (e.g. MNIST data set [71]), or on directly comparable data which allows easy transfer (e.g. for facial recognition).

- One- or Few-Shot-Learning:
  The Field of One- or Few-Shot-Learning tries to bridge the gap between Artificial Intelligence and human learning. Similar to human learning, the image or object should be recognized after very few samples. This research field is especially active in topics like image classification or object tracking. There

are also applications in other fields, like e.g. gene research in biology [61]. In terms of solving this challenging task, there is a variety of elegant approaches, as elaborated by Wang [119], some of which overlap with sketch and pattern recognition. The very basic principle is that all available knowledge should be used. The next section will introduce three basic principles, how knowledge can be integrated into a model.

- Physics-guided machine learning:
  Physics-guided machine learning combines the field of machine learning with physical models, experience and experimental data.
  One way to achieve this is by incorporating physically meaningful features to improve feature extraction. Besides the improved features, the big advantage of this approach is an explainable model which can be understood and interpreted by experts [11]. Another approach uses physical models to jumpstart machine learning models. Effects not considered in the physical model can be learned by the machine learning model and, for example, forecasts of physical properties can be improved [11] and better physical understanding of the system can be developed.

This approach for combining the three fields has already successfully practiced in several fields outside of physics. For financial time series, several approaches [117, 113] are based on the 53 chart patterns of Bulkowski [13], in which the unique characteristics and relationships of price movement are examined in detail. In the field of sketch based query system on time series, the expert can sketch simple patterns, which are then recognized within time series data [25, 82]. In the field of network attacks, there are approaches for detecting DDoS attacks [118] or attacks on software switches [77] more efficiently with a combination of anomaly detection and sketches.

In the field of predictive maintenance and machine failures, to our knowledge, this idea is as yet barely touched, and no comparable literature could be found.

6.1.2 Algorithmic Taxonomies

In this section, we introduce different strategies, which allow One- or Few-Shot-Learning approaches to generalize the very limited amount of available information. This knowledge will be very valuable for choosing suitable approaches in the next section.

The following structure and syntax is taken from the recent survey by Wang [119]. All ideas are visualized in Figure 6.3. We start by introducing the syntax:

- $h_I \in \mathcal{H}$ represents the function of the empirical risk. It describes the loss over the training set I, with $\mathcal{H}$ being the hypothesis space of all possible solutions.
- $h^* \in \mathcal{H}$ represents the function of the expected risk. It describes the loss over the testing set with the estimation error ε_{est}.
- $\hat{h}$ represents the function of the expected risk in the scoring process with the approximation error ε_{app}.

In machine learning cases with sufficient training data, a lot of information can be extracted by the training data. Thus, h_{I} and h^* are very close in $\mathcal{H}$ and ε_{est} is accordingly very small (see Figure 6.3 top left). In the setup of Few-Shot-Learning (top right), the information in the training set is very limited, and generalizations have to be learned, which strongly increases ε_{est}. Wang [119] describes three different strategies how to tackle this challenge:

Data Prior knowledge is used to augment the training data I, and to generate extensively increased training data $\tilde{I}$. In this way, the Few-Shot-Learning challenge is transformed into a "normal" machine learning challenge and standard algorithms can be used.
Data Augmentation includes methods like translation, flipping or rotation of a picture. It can also make use of similar or unlabeled data sets.

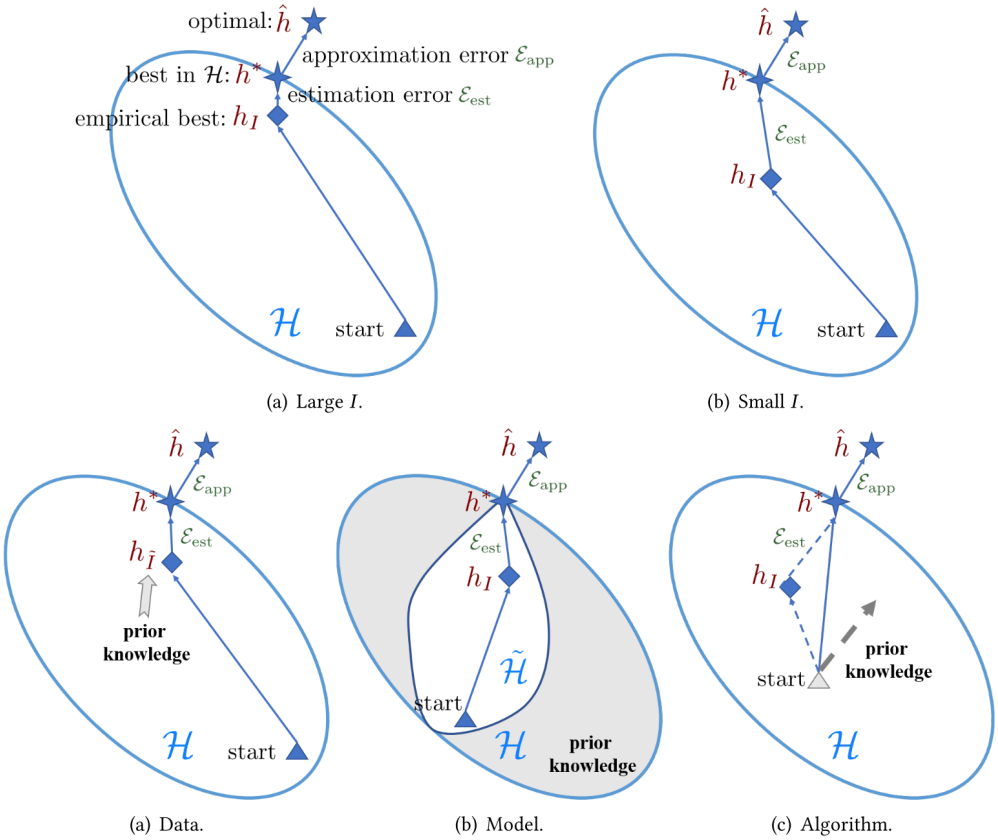
Model Prior knowledge is used to constrain the complexity of $\mathcal{H}$, which results in a much smaller hypothesis space $\tilde{\mathcal{H}}$. Within this smaller hypothesis space, a stable solution can be achieved with the existing training data more easily.
Simultaneous multitask learning of similar challenges and embedding the samples into a lower-dimensional space make use of this concept.

Algorithm Prior knowledge is used to optimize the search strategy by providing a good initialization or by guiding the search steps.
For example, knowledge of similar challenges can be used in order to pre-train most of the models parameters, or a suitable simplified model can be chosen.

In most cases, a combination of the three methods is used in order to find a suitable solution.

6.1.3 Algorithm: Dynamic Time Warping

In the following, we want to focus on one specific algorithm: the Dynamic Time Warping (DTW). It has proven successful in a broad variety of fields and sketch-based applications [46, 64, 82, 87, 88]. Rakthanmanon even states that "after an



(a) Large I. (b) Small I.

(a) Data. (b) Model. (c) Algorithm.

Figure 6.3: Concepts for the solution of Few-Shot-Learning challenges. Top left: Normal machine learning with a lot of data. Top right: Few training samples which lead to substantially increased estimation error ε_{est}.
Bottom: Illustration of three different concepts to decrease ε_{est} by using prior knowledge. Left: Increase of the training data I to $\tilde{I}$ by data augmentation. Middle: Constraining the hypothesis space of all possible solutions $\mathcal{H}$ to a smaller subset $\tilde{\mathcal{H}}$ by the choice of model and model parameters . Right: Optimized search strategy by providing good initialization or by guiding the search steps. [119]

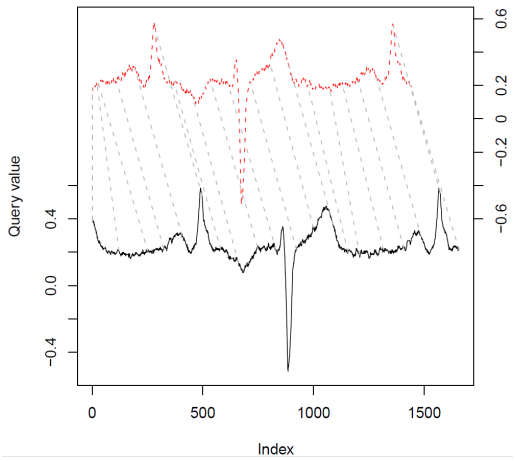


Figure 6.4: Visualization of DTW matching of two time series. [40]

exhaustive literature search of more than 800 papers, we are not aware of any distance measure that has been shown to outperform DTW by a statistically significant amount on reproducible experiments" [95]. In the following, we introduce the basic principle of the algorithm, and an extension for higher dimensions.

Basic Algorithm

Dynamic Time Warping (DTW) determines the similarity of two temporal sequences, which may be distorted in time. For example, it can be applied to recognize people via speech or walking patterns, despite variations in speaking and walking pace.

Mathematically, the DTW tries to minimize the distance between two time series X and Y, by stretching or compressing them locally [40]. Thereby, the test time series $X = ((t_1, x_1), ..., (t_N, x_N))$ can have a different length than the reference time series $Y = ((\tau_1, y_1), ..., (\tau_2, y_M))$, with the time indices t_i and τ_j, and values x_i and y_j respectively. A pairwise distance measure $d(x_i, y_j) \geq 0$ acts as a similarity measure of the two time series. The warping curve $\phi(k) = \big(\phi_x(k), \phi_y(k)\big) = \big(t_i, \tau_j\big)$ maps the time indices of the two time series onto each other. DTW optimized this mapping in order to minimize the accumulated distance measure. Figure 6.4 shows such a mapping, all minima and maxima of the two curves are mapped onto each other in order to minimize the distance.

The definition of the distance measure is not fixed and can be chosen in respect to the nature of the data. Hence, DTW can act on any kind of data as long as a well-defined distance can be calculated. In most cases, the Euclidean distance is used:

$$d_\phi(X,Y) = \sqrt{\sum_{k=1}^{K} \Big(X(\phi_x(k)) - Y(\phi_y(k))\Big)^2} \tag{6.1}$$

For the warping curve, every test time index t_i, $i \in \{1..N\}$, has to be matched with a reference time index τ_j, $j \in \{1..M\}$, and vice versa. The start point (t_1, τ_1) and the end point (t_N, τ_M) are normally mapped to each other. In order to impose time ordering and to avoid unnecessary loops, the mapping must be monotonic in both the time indices t_i and τ_j.

Depending on the application, further local and global restrictions may be introduced. Locally, so-called "step patterns" can ban specific mappings or introduce weightings. For example, they can restrict the consecutive number of elements $t_i..t_{i+n}$, which can be mapped on the same τ_j (and vice versa). Globally, a-priori knowledge about maximum time distortion can be used. For example, the popular Sakoe-Chiba band [98] imposes a maximum time deviation T_0 between the two matched indices:

$$\forall \phi_k = (t_i, \tau_j) : |t_i - \tau_j| \leq T_0 \tag{6.2}$$

Interestingly, the DTW distance measure does not classify as a metric, as the triangle inequality is not ensured in all cases.

In terms of time complexity, the DTW computes in the order of $O(N \cdot M)$. Lower orders could be only achieved for special cases [41].

Multidimensional DTW

The DTW method can be easily extended to K-dimensional time series with a suitable choice of the distance measure. Two popular implementations are used in literature (although it is often not specified which one is used [87]). Here, the test time series $\vec{X} = ((t_1, \vec{x}_1), ..., (t_N, \vec{x}_N))$ and the reference time series $\vec{Y} = ((\tau_1, \vec{y}_1), ..., (\tau_M, \vec{y}_M))$ consist of k-dimensional vectors $\vec{x}_n = [x_{n1}, ..., x_{nK}]$ and $\vec{y}_m = [y_{m1}, ..., y_{mK}]$ for each time point t_i and τ_j.

1. DTW with independent warping (DTW$_i$)
   The DTW measure is calculated separately for each dimension k and the results are added.

$$\text{DTW}_i = \sum_{k=0}^{K} \text{DTW}(X_k, Y_k) \tag{6.3}$$

2. DTW with dependent warping (DTW$_d$)
   The n features are taken as n-dimensional vector and a single DTW measure is calculated with the n-dimensional definition of the Euclidean distance.

$$\text{DTW}_d = \text{DTW}(\vec{X}, \vec{Y}) \quad \text{, with } d(\vec{x}_i, \vec{y}_j) = \sqrt{\sum_{k=0}^{K} (x_{ik} - y_{jk})^2} \tag{6.4}$$

The most essential difference between DTW$_i$ and DTW$_d$ lies in the warping curve. For DTW$_d$, all features share the same warping curve, whereas in the case of DTW$_i$, every feature has its own warping curve, allowing different features to have different matching curves. Depending on the use case, one or the other definition may be more suitable.

6.2 Sketch Preparation

After having a brief look into literature and a possible algorithm, this section will focus on the first part of PEESR: The "physics- and expert driven sketches". As a first step, different error cases will be discussed with experts, and first expert-driven sketches will be created. In a second step, those sketches will be enriched by the physical findings of Chapter 3.

6.2.1 Sketching Error Cases

The first goal is to obtain sketches of error cases for the field of small transfer stars (see Figure 6.5). Thereby, two questions are asked: Which error cases can appear during operation of a transfer star? And which behavior in the current is expected for each error case, especially in the angle-based or high-resolution curves? If possible, sketch a characteristic pattern?

In order to collect different points of view and knowledge, two discussion rounds are happening, once an one day workshop with Krones mechanical engineers, and once a two hour call with an experienced maintenance manager of a brewery.

The answers of the two workshops are summarized in the following. In brackets is the short work name, which will be used in the following. Some exemplary sketches can be found in Figure 6.6.

Synchronization Error (Sync)

As already described in Section 2.2, the synchronization error is one of the most frequent error cases in production. It describes the state in which the two stars

Figure 6.5: Example of two transfer stars (white) transporting the bottle to and away from the big filler carousel in the middle.[67]

don't line up during an handover - which results into small crashes at every handover. As already discussed, according to Claim 1 this error should not happen, as the initial fixation is permanent. Nevertheless, the forces of a crash can shift the fixation. Additionally, the fixation has to be corrected with every product change, as parts of the stars are swapped, to take the new bottle shape into account. An imprecise fixation at that point leads to a constant misalignment for the next hours of production.

As already discussed in Section 3.4, a small crash during an handover leads to a slowing down and then speeding up behavior of the machine. This results in a variation of the current. For the synchronization error, every handover leads to this variation of the current. Thus, the sketch looks similar to a sinus wave.

Error cases concerning One Station (One station)

The next sketch summarizes all error cases concerning one station. A frequent error case is that one (or more) crown caps are lying in the carousel. This happens especially at the neighbor stars of the crowner. The cap decreases the space in the station for the bottle which can influence the handovers. Another error case is a mechanical defect of one station like a broken corner.

Those error cases influence the handovers of one station, leaving all other handovers unchanged. For these error cases, we expect to differentiate two cases. For star having two handovers to other stars, there should be two handovers which show crash-like behavior. If a star transports the bottle from a star to a conveyor belt (or vice versa), just the star handover should lead to that faulty behavior. The handover to the conveyor belt comes with a higher degree of freedom and thus should not lead to an increase in current.

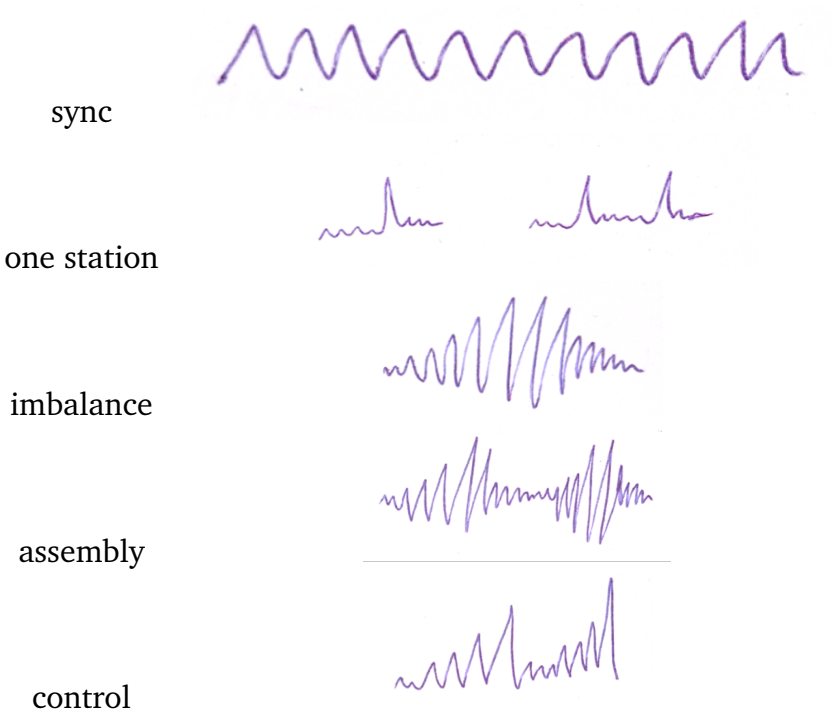


Figure 6.6: Hand-drawn sketches, created in workshops with experts.

Transfer Plate Error

For the error case called transfer plate error, the transport height of the bottle bottoms does not fit. Normally, the bottles are always pushed slightly downhill from one star to the next. If the previous star is slightly lower than the next, the bottles have to be pushed slightly uphill with every handover. This causes a small crash similar to the synchronization error, just concerning with a different part of the bottle and carousel.

As this error case influences all handovers, we end up with the same sketch as the synchronization error. On the sketch-level, we don't expect that the two error cases can be differentiated. On an occurrence-level, we expect this error case mainly after an overhaul.

Carousel Imbalance (Imbalance)

Imbalance has a very similar origin like the transfer plate error. The carousel is not built up completely parallel to the floor, but is slightly tilted, leading to different heights at different angle. For some bottles, the handover is normal, but for some others the bottle has to be pushed slightly uphill.

Sketching this error case would result into an oscillation in current, some handovers go easy, others have a faulty characteristics.

Incorrect Assembly (Assembly)

The product specific part of the star consists of two parts in order to enable an easier product change. If those two parts are not assembled correctly, each half experiences the same behavior as the transfer plate error. For some bottles in each half, the handovers are okay, others experience some resistance.

It is difficult to predict, how the incorrect assembly exactly influences the current, except that it should repeat once per half. Due to the similarity to the carousel imbalance, we sketch an oscillation with twice the frequency.

Friction at Railing

During the transport in the star, the railing keeps the bottles in place. In case the railing is positioned too close to the star, the bottles experience higher friction with the railing. This should result in an average higher current with keeping the patterns of handovers unchanged.

This leads to the conclusion, that this error does not change the pattern of the current and thus it is not possible to sketch a characteristic pattern.

Motor and Gear Failures

Independent of the direct bottle transfer, different defects in the motor or gears can be imagined. For example, wear increases the play of the gear, or bearings in the motor break.

As the gear and the motor rotate in a higher frequency than the star, those error cases will not appear as characteristic patterns in the current averaged over the carousel angle. In a high resolution signal of the current, those frequency could be detected. As this kind of data is not available for this study, this class of error cases will not be considered further.

Bearing Failures

The carousel is balanced on bearings which enable movement with little friction. Bearings failures are known to show very characteristic fault frequencies [43]. Some of them are multiples of the rotational frequency and thus should show up in the current pattern. Those formulas are based on specifics of the bearings, like number of balls or ball diameter.

After some research and contacting the producer of the bearings, it was surprisingly not possible to acquire those parameters for the stars, which will be considered in the following. The bearings are part of the motor-gear-block and the specifics are handled as valuable knowledge of the producers. This limits the sketch to an unspecific high-frequency pattern. As the rough order of frequency is unknown, this error case will not be considered for the moment.

Control Error (Control)

The last group of error cases are caused by the control. Depending on the control parameter, the reaction parameter can be set too tight or too loose, resulting in an overreaction or a delayed reaction. As already discussed in Section 3.4.4, the field of control errors is very complicated, especially as different kind of controls are in use.

This part of the machine is so far ignored in the physical model and predictions require a more detailed knowledge of the used control. Nevertheless, we want to add one sketch of a control error, which was reported by the maintenance manager. In a typical overreaction of the control, the current is building up over the course of half a round in an exponential manner and then break down, just to start over again.

6.2.2 Physical Improvement of Sketches

The discussion with the experts resulted into five error cases which could be detected in the current with in total seven sketches. In a next step, we combine those sketches with the gained knowledge of the Physics-Chapter 3 and transfer the hand-drawn sketches to computer-based sketches.

In Chapter 3, we created a simplified physical model of a faulty handover and compared it to an anomalous handover. As the two curves showed big similarities, we use the data of the anomalous handover as basis for the computer-based sketches. The physics shows a direct relationship between level of misalignment and amplitude of the current. With this knowledge, we take the anomalous handover and just modify it in terms of amplitude for simulating healthy and non-healthy handover.

For the computer-based sketches, two main parameters were identified which can vary from star to star: The number of stations n_{s}, and the number of the stations between the handover from one star to the next n_{delta}. The two parameters are implemented as input for the computer-based sketches. This allows a fast adaptation of the sketches for different machines and stars.

The final sketches are depicted in Figure 6.8.

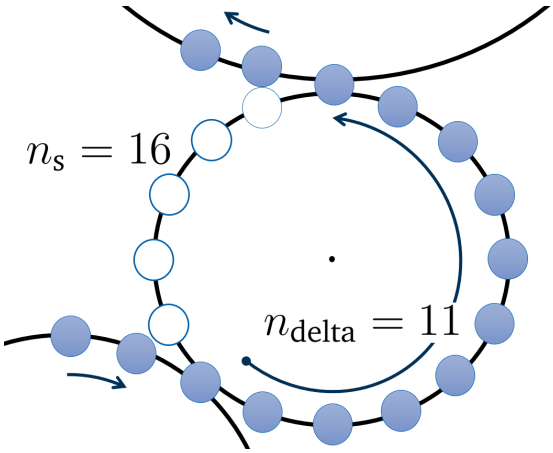


Figure 6.7: Example of a star with 16 stations ($n_s = 16$) and 11 stations between the two handover to the neighbor stars ($n_{delta} = 11$).

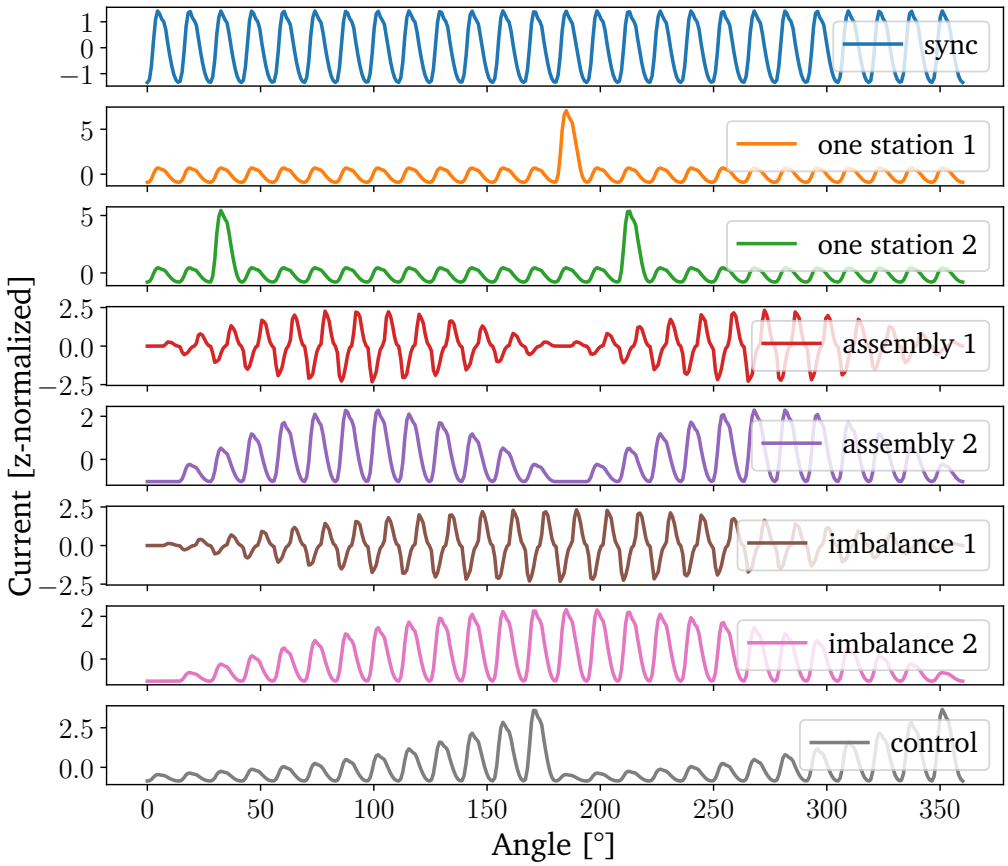


Figure 6.8: Sketches of different error cases for $n_s = 26$ and $n_{delta} = 13$.

6.3 Study Setup

6.3.1 Scoring Data

The study is performed on data of two different bottlers, with each three different stars. The two bottler - in the following called bottler B and bottler C - are the same as used for the qualitative Transfer Learning test in Section 5.7. Applying the algorithm to in total six different stars allows testing the portability of the algorithms and sketches in the study. Both bottlers are breweries, one situated in Germany, the other in Poland.

For bottler B, there are 7 months of data available, for bottler C 8 months, both of the year 2020. The same preprocessing as in Section 2.4 is performed. As an additional step, a slight Gaussian smoothing is used to reduce noise. As a last step, all curves are z-normalized to ensure comparability.

Summed over the three stars, the preprocessing results in 4.063 curves for bottler B and 6.529 curves for bottler C. The difference in number results from more production days for bottler C in the available time period.

In order to receive labeled error cases, three steps are performed on the curves:

1. Anomaly Detection
   For each star, a reference curve is chosen and the most promising anomaly detection method from Chapter 5 is performed on the data. The anomaly criterion is chosen weaker than in the previous study in order to not miss any interesting patterns. In total 2.606 anomalous patterns are detected.

2. Remove Duplicates
   As a next step, some patterns reoccur over a longer period of time and are thus over-represented in the study. All curves, which correlate to at least $97\,\%$ to an earlier curve of that star, are ignored. This results in 305 distinctly different curves with which we will continue working in the following.

3. Manual Labeling
   The last step is a manual classification of the curves. The classification person is provided with the sketches and asked to classify the data into those categories as shown in Figure 6.9. As a lot of curves don't represent one error case very clearly, maximum two categories can be chosen. Additionally, there is a category for new distinct patterns - which are missing in the sketches - and one for unclear error cases, mainly for overreactions of the anomaly detection due to the weaker criterion. Examples of the labeled data can be found in Figure 6.10.
   Being able to choose up to two error cases also has a physical background.

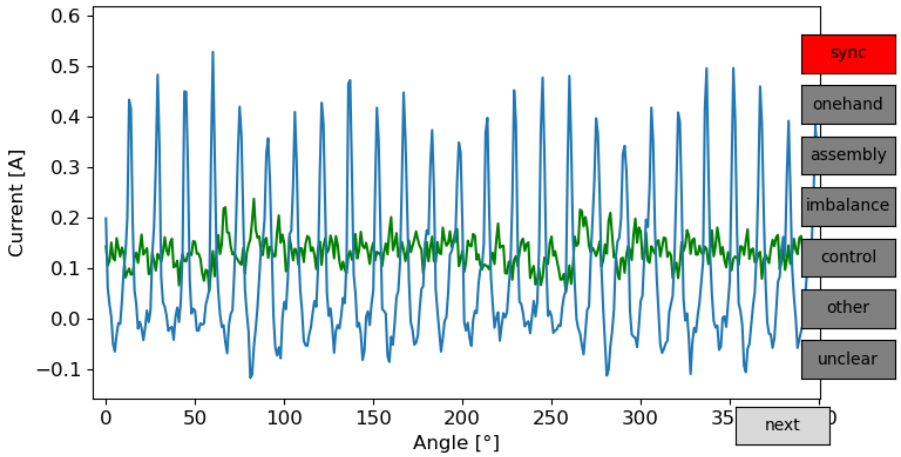


Figure 6.9: Screenshot of tool for manual labeling. The graph shows the curve to be classified (blue) in comparison to the reference curve (green). In this case the user has chosen the error case "sync". Up to two patterns are allowed to be marked. With the button next, the next pattern is shown and all buttons are reset.

Most of the error cases can happen independently from each other (e.g. assembly and control), or one error case can enhance the other error case (e.g. assembly and sync).
In 148 curves one or two of the sketched patterns can be clearly identified. For the other curves, the classification person gives a guess to which category it could belong (when possible). Interestingly, three new shapes are also discovered in the data set, which were not predicted by the sketches. We will use this knowledge in order to test the algorithm, if they can be found.

6.3.2 Algorithmic Setup

As algorithm, we will use the widely suggested Dynamic Time Warping (DTW), as described in Section 6.1.3. In the following, we define different data augmentation and DTW setups, which will be tested, and a classification method.

Data Augmentation and DTW settings

Inspired by the introduced algorithmic taxonomies of one-shot-learning in Section 6.1.2, we want to implement some of the ideas and evaluate their effect. Thereby, we choose in total four techniques, each two from the classes "data" and "model". The techniques are depicted in Figure 6.11.

1. Data Augmentation 1: Mirroring & Translation (mandatory)
   We increase the amount of our training data set by horizontally mirroring,

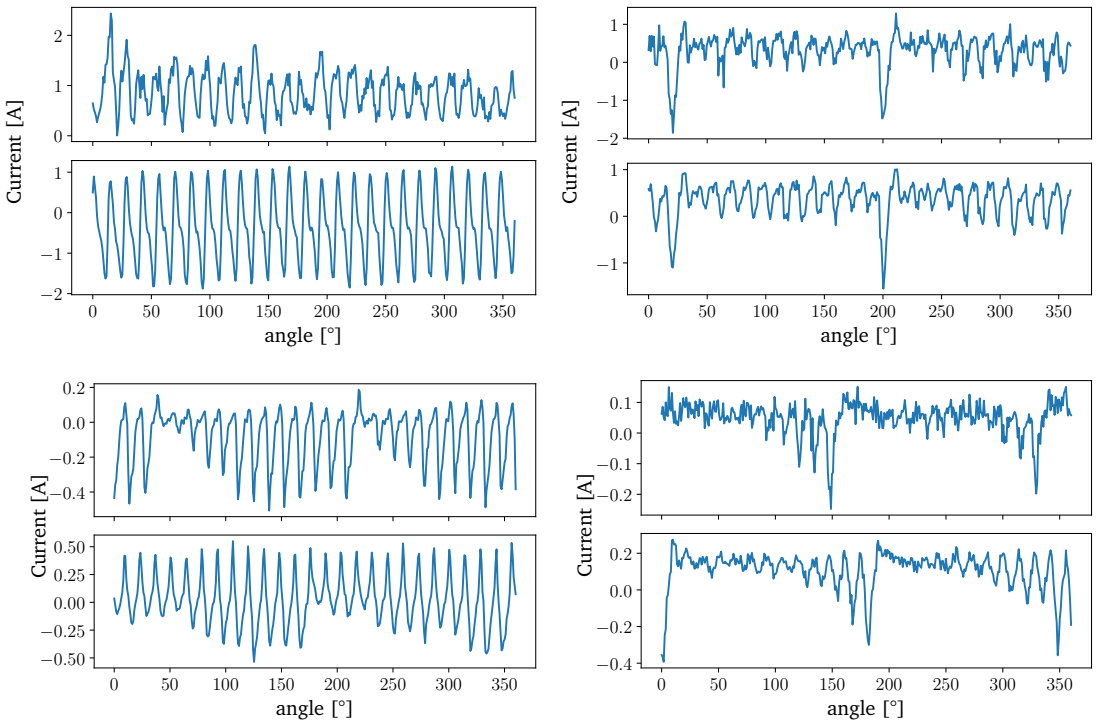


Figure 6.10: Examples of labeled "real-life"-patterns: Synchronization Error (top left), One Station Error (top right), Assembly error (bottom left) and Control Error (bottom right).

and by vertically translating each pattern. Both augmentation methods have a physical background: The error can occur at any position of the star, and a low-mass star can be pushed or pulled by the high-mass star, resulting into a positive or negative current. All setups will use this enriched data set as a basis to enhance the probability to recognize the patterns.

2. Data Augmentation 2: Randomizing
   So far the sketches resemble the perfect situation that all handover show the same exact behavior. In reality, there is always a variation between handovers. We will use that prior knowledge and create for every sketch 10 additional sketches, for which the heights of the different handovers randomly vary of $\pm 10\%$. We will evaluate the effect of this data augmentation method (in the evaluation: random = "yes" / "no").

3. Model simplification 1: Restrict maximum distortion in DTW
   In terms of reducing the complexity of the DTW model, we restrict the hypothesis space of all possible solutions by using the Sakoe-Chiba band in DTW. The Sakoe-Chiba band [98] limits time distortion by restricting a maximum time deviation between the two matched indices of testing and reference data. In our case, time / angle distortions should not exceed more than two handovers as otherwise some sketches are not distinguishable anymore. We will set the maximum distortion to a variety of values: 0, 0.25, 0.5, 1 and 2 handover. The option "0 handover" of DTW represent the normal euclidean distance.

4. Model simplification 2: Feature selection
   In order to reduce the complexity in the data, a manual feature selection is performed. Examining the sketches, the information of each handover can be summarized by calculating the amplitude, minimum or maximum value. We will use two different setups, once just with amplitude, and once with all three features. In this way, the curve of 400 points is transformed into a one- (1d) or three-dimensional (3d) feature space with a length of 26 points. This reduces the complexity by 94% or 80%, respective.

The data augmentation 1 is always performed, for the other enrichment methods all different combinations are considered. In terms of translation step size (Data Augmentation 1), a step size equal to five is used for the normal 400-dim pattern, and equal to one for all feature-engineered data. As DTW can handle slight shifts, the step size of five is used in order to reduce computational time.

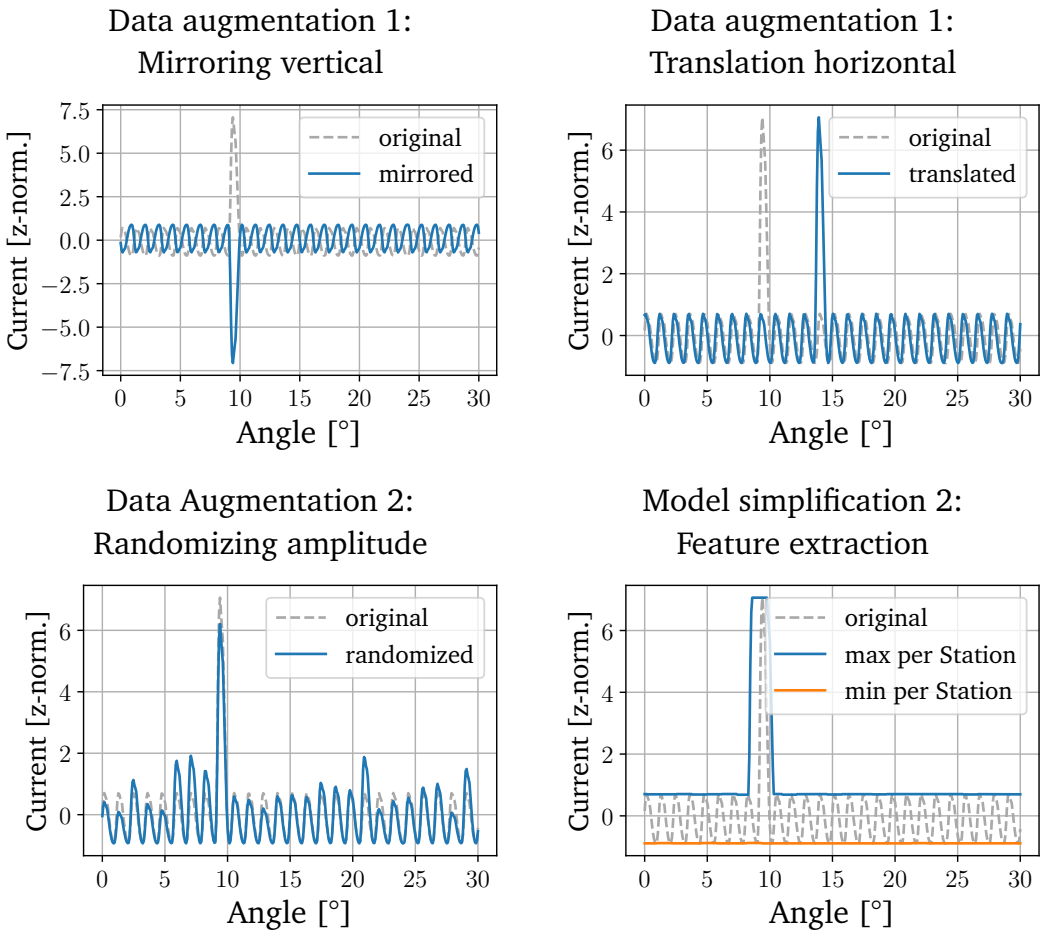


Figure 6.11: Exemplary visualization of different data augmentation and model simplification methods.

Classification Process

The DTW is performed for every sketch, and the winning sketch is chosen by the smallest DTW measure. A second winner is added if its DTW measure differs by less than a percentage δ_{max} from the first winner. This allows illustrating uncertainties of the algorithm, similar to the uncertainties of the classification person. In this study, δ_{max} is set once to $\delta_{\text{max}} = 5\%$ and once to $\delta_{\text{max}} = 10\%$.

6.3.3 Measures

The different setups will be compared by the following three measures:

1. Accuracy:
   The accuracy determines how well the predicted labels fit together with the manual labels. We will use two different definitions:

   (a) Weak Accuracy: As soon as one of the (maximum two) classifications of the algorithm fit together with at least one of the (maximum two) manual classifications, it will enter the score as "correctly classified". The number of "correctly classified" samples divided by the total number of samples give the accuracy. The accuracy depends on the chosen δ_{max}, which was introduced above.

   (b) Strong Accuracy: In comparison to the Weak Accuracy, the Strong Accuracy just scores complete hits as "correctly classified". The classification of the algorithm with one or two classes has to fit exactly the one or two manual classifications. Any deviation is classified as negative. Equivalent to the Weak Accuracy, the Strong Accuracy depends on the choice of δ_{max}.

2. Resource Usage:
   As already pointed out in Chapter 5, the code has to run on very resource-limited edge device. For that reason, we introduce the following two measures:

   (a) Scoring Time: The Scoring Time is the calculation time per sample on a standard laptop (see details 2.6) averaged over at least 10 samples. It will differ from computer to computer but will give an order of magnitude which is expected on the edge device. Additionally, it allows to compare the computational complexity of the different setups as all calculations were performed on the same computer under the same conditions.

Setups which need more than one minute to score one sample will be stopped immediately, calculations longer than 30 sec will be also discarded except they show extraordinary accuracy.

(b) Model Size: Memory space is limited on the edge device, thus the measure model size will specify the size of the model which is needed for the scoring process. Setups with model sizes larger than 20 MB will be excluded from the evaluation.

3. Possibility to express uncertainty and detect new patterns:
Sketch-detection in real data always comes with some uncertainty. The approach should offer the possibility to measure this uncertainty. In this way, wrong classifications can be reduced and new patterns which are not represented by the sketches can be detected.

6.4 Results

In the evaluation of the different setups, the difficulty for the algorithm will be incrementally increased. We start with the scoring of curves, which are uniquely assigned to one training class. In the next step, we also consider the curves which resemble a combination of two classes. In the last step, all curves are taken into account, also the ones with could not be assigned to any pattern. In each step, the best setups continue in the competition.

The comparison measures will stay the same over the course of the chapter and are based on the Section 6.3.3.

As implementation of the DTW, the Python dtaidistance package [85] is used.

6.4.1 Step 1: Curves with similar patterns to sketches

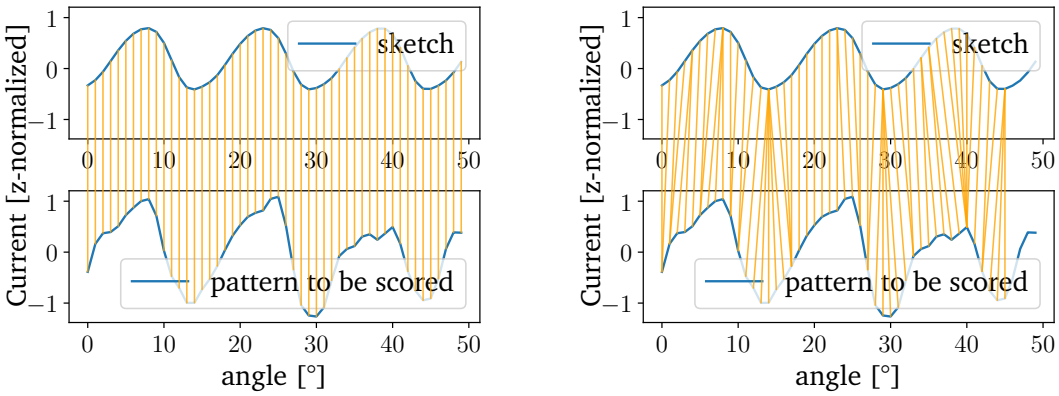
The first test contains all curves, which can be unambiguously mapped to one of the sketches. According to the manual classification process, 70 patterns fulfill that criterion. They distribute on the four sketch groups: Sync (42), One Station (3), Assembly (9) and Control (16). The sketch group Imbalance is not represented.

The scoring process was performed on all setups described in Section 6.3.2. The results are summarized in Table 6.1. The evaluation concentrates on setups, which perform worse than the others. They are marked yellow or red.

We start with examining the resource usage. All setups easily pass the limit for the model size of 20 MB. The biggest model barely exceeds 0.5 MB. In terms of scoring time, there is huge variation between the setups. Some of the setups without feature extraction exceed the limit of 60 sec and are thus aborted during

Feature Selection	Random	Max Distortion [handover]	Scoring Time [sec]	Model Size [kB]	Weak Acc. $\delta_{\max} = 5\%$	Strong Acc. $\delta_{\max} = 5\%$	Weak Acc. $\delta_{\max} = 10\%$	Strong Acc. $\delta_{\max} = 10\%$
none	no	eucl.	14.2 sec	60 kB	98.6%	90.0%	98.6%	85.7%
		0.25	10.5 sec	60 kB	97.1%	97.1%	98.6%	94.3%
		0.5	19.5 sec	60 kB	97.1%	95.7%	98.6%	94.3%
		1.0	35.4 sec	60 kB	98.6%	95.7%	98.6%	94.3%
		2.0	>60.0 sec	60 kB				
	yes	all	>60.0 sec	532 kB				
amplitude (1d)	no	eucl.	0.2 sec	14 kB	90.0%	84.3%	94.3%	81.4%
		1.0	0.2 sec	14 kB	95.7%	94.3%	97.1%	91.4%
		2.0	0.4 sec	14 kB	95.7%	92.9%	97.1%	88.6%
	yes	eucl.	2.4 sec	117 kB	90.0%	87.1%	95.7%	81.4%
		1.0	2.8 sec	117 kB	95.7%	92.9%	95.7%	90.0%
		2.0	3.3 sec	117 kB	94.3%	94.3%	94.3%	92.9%
amplitude, min, max; (3d) dep. warping	no	eucl.	0.2 sec	42 kB	90.0%	84.3%	92.9%	78.6%
		1.0	0.4 sec	42 kB	95.7%	94.3%	97.1%	92.9%
		2.0	0.6 sec	42 kB	95.7%	94.3%	97.1%	88.6%
	yes	eucl.	1.4 sec	351 kB	91.4%	84.3%	92.9%	77.1%
		1.0	3.3 sec	351 kB	94.3%	92.9%	97.1%	90.0%
		2.0	4.7 sec	351 kB	95.7%	92.9%	95.7%	90.0%
amplitude, min, max (3d) indep. warping	no	eucl.	0.7 sec	42 kB	87.1%	82.9%	92.9%	80.0%
		1.0	1.0 sec	42 kB	95.7%	92.9%	95.7%	90.0%
		2.0	1.3 sec	42 kB	97.1%	92.9%	98.6%	88.6%
	yes	eucl.	6.5 sec	351 kB	90.0%	81.4%	91.4%	77.1%
		1.0	7.8 sec	351 kB	95.7%	92.9%	95.7%	91.4%
		2.0	9.2 sec	351 kB	97.1%	92.9%	97.1%	88.6%

Table 6.1: In the first step, 70 unambiguously classified curves are scored in different setups and evaluated in terms of two accuracy measures and two resource measures.



(a) Point-to-point matching for calculating the Euclidean distance.

(b) Point-to-point matching for calculating the DTW measure with a maximum distortion of 0.25 handovers.

	Eucl. Distance	DTW measure Max. Dist: 0.25 handover
Assembly	12.02	9.09
Imbalance	12.02	9.15
Sync	12.06	8.07
One Station	16.62	12.70
Control	17.70	14.80

(c) Distances for one scoring sample for Euclidean Distance and DTW measure.

Figure 6.12: Comparison of Euclidean Distance to DTW measure.

the calculation. In general, there is the clear correlation between the calculation time and the number of dimensions, amount of training data (e.g. via randomization) and maximum distortion. The only irregularity happens between row one (no feature extraction, eucl.) and row 2 (no feature extraction, max dist 0.25), as the translation step size is decreased from five to one (as mentioned in Section 6.3.2).

As next step, we evaluate the accuracies. The weak accuracy lies for all setups between 87.1% and 98.6%, the strong accuracy between 77.1% and 97.1%. This is a rather encouraging result as all setups perform in an acceptable range, some in an extraordinary range. Looking into details, the setups without feature extraction show slightly better and more stable results than the ones with feature selection, and that even for the small maximum distortion of 0.25 handovers.

In general, most setups with euclidean norm are outperformed by a neighbor DTW setup. The only exception is row one, the euclidean distance without any feature extraction. This setup achieves the best result in the weak accuracy of all, but doesn't perform that excellent in the strong accuracy. We want to have a closer look on that phenomenon in Figure 6.12. The top shows the point-to-point matching for a real-life example for the euclidean norm (a) and DTW (b). One can clearly see, how DTW manages to compensate variations in the curve. The bottom table shows the minimum distance for one sample for each sketch. One notices, that the euclidean distance finds the TOP 3 sketches, but has difficulties to determine a winner. This behavior leads to a high weak accuracy, but a low strong accuracy. DTW detects the same TOP 3, but is able to differentiate them and declare Sync as clear winner. This gives the DTW the crucial advantage.

For the step 2, we will reduce the number of setups by excluding all which exceed a calculation time of 30 sec, or show an accuracy lower than 80% (marked in red in Table 6.1). This leaves us with in total 18 setups.

6.4.2 Step 2: Curves with mixtures of up to two sketches

In the second step, the data set is extended by patterns, which were manually classified as one or a combination of two of the defined sketches, leading to a total of 148 curves. This also includes patterns for which the classification person was unsure, but guessed one sketch. They group into Sync (103), One Station (3), Assembly (9), Control (28), and the combination of Sync and Assembly (5). As the resource usage per scoring event is unchanged from Step 1, it will not be discussed again. The results are shown in Table 6.2.

With the exception of the setups based on the euclidean norm, all setups exceed a Weak Accuracy of 88.5% and a Strong Accuracy of 77.0%, some of them even reaching accuracies of $98.0\,\%$. This result is again extremely promising, especially taking into account that in about half of the cases the classification person was not $100\,\%$ sure. The only significantly worse performing setup is the setup, which does not make use of any enrichment methods of Section 6.3.2 (row 1). Otherwise, interestingly, there are differences between the different setups, but no strict trends can be detected. In most cases, the DTW-setting with a maximum distortion of one handover seems to overrule the other settings.

This broad successful behavior is very encouraging as it is a sign for the stability of the algorithmic choices.

In order to develop a better feeling for the different setups, we compare two confusion matrices of similarly well-performing setups. The two chosen setups are marked with green in Table 6.2. The left confusion matrix of Figure 6.13, orig-

Feature Selection	Random	Max Distortion [handover]	Weak Acc. $\delta = 5\%$	Strong Acc. $\delta = 5\%$	Weak Acc. $\delta = 10\%$	Strong Acc. $\delta = 10\%$
none	no	eucl.	90.5%	69.6%	93.2%	65.5%
		0.25	95.9%	88.5%	98.0%	82.4%
		0.5	93.9%	83.8%	94.6%	79.1%
amplitude (1d)	no	eucl.	90.5%	80.4%	93.2%	79.1%
		1.0	93.2%	81.8%	95.3%	79.1%
		2.0	89.2%	83.1%	91.2%	77.0%
	yes	eucl.	90.5%	82.4%	93.9%	78.4%
		1.0	91.9%	84.5%	94.6%	81.1%
		2.0	91.2%	80.4%	93.2%	79.1%
amplitude, min, max (3d) dep. warping	no	1.0	91.2%	83.8%	95.3%	81.1%
		2.0	90.5%	81.8%	93.9%	78.4%
	yes	1.0	91.9%	86.5%	95.3%	83.1%
		2.0	93.9%	81.8%	95.3%	79.1%
amplitude, min, max (3d) indep. warping	no	eucl.	88.5%	79.7%	91.9%	77.7%
		1.0	91.2%	83.8%	93.9%	79.1%
		2.0	91.2%	81.8%	95.3%	79.1%
	yes	1.0	92.6%	85.8%	92.6%	81.8%
		2.0	93.2%	82.4%	93.9%	79.1%

Table 6.2: In the second step, 148 curves (which are classified as a mixture of up to two sketches) are scored in different setups and evaluated in terms of two accuracy measures. $\delta_{\max}$ was here shortened to δ for space reasons.

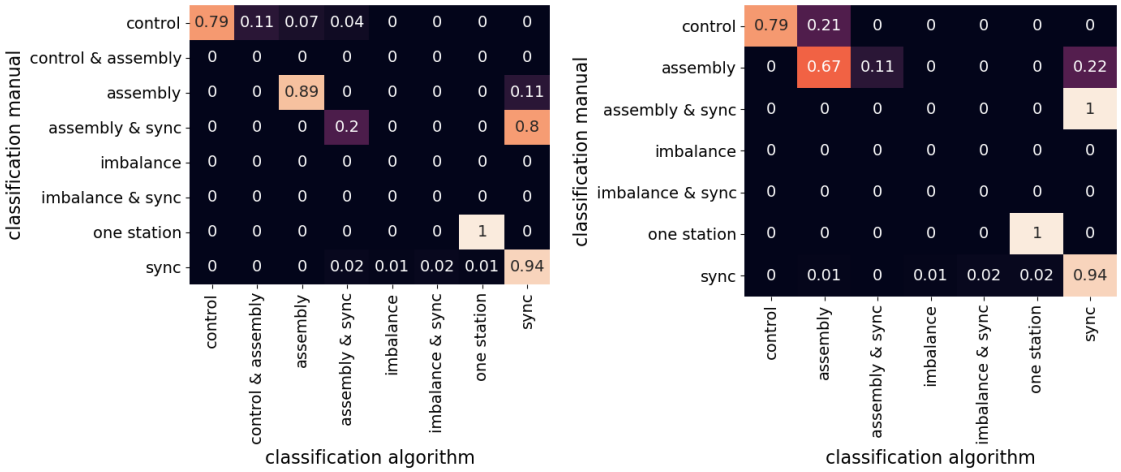


Figure 6.13: Two similarly well performing confusion matrix normalized by the classification of expert. Left: Confusion matrix of the original curve without randomization, maximum distortion of 0.25 handovers and $\delta_{\max} = 5\%$. Right: Confusion matrix of the 3-dimensional feature extraction with dependent warping including randomization, maximum distortion of 1 handovers and $\delta_{\max} = 5\%$. The two confusion matrices are very similar with the left one being slightly stronger in recognizing the "Assembly" error cases.

inates from the original curve without randomization, a maximum distortion of 0.25 handovers and a $\delta_{\max} = 5\%$. The right one is the confusion matrix of the 3-dimensional feature extraction with dependent warping including randomization, a maximum distortion of 1 handover and also $\delta_{\max} = 5\%$. Both matrices are normalized by the manual classification (rows). Interestingly, the two matrices show high similarities. The left side shows slightly stronger recognition of "Assembly" error cases. Both setups perform very reliable for the error cases of "Synchronization" and "One Station". This implies that the models learn very similar characteristics of the curves.

As the different setups show very similar results, almost all setups would qualify for further consideration. With the knowledge that they show similar confusion matrices, we can reduce nevertheless the number of setups without loosing information. We continue with the five setups, which have a Weak Accuracy above $90\,\%$ and a Strong Accuracy above $80\,\%$. All accuracies failing these two criteria are marked in orange in the Table 6.2.

6.4.3 Step 3: Realistic scenario

The preceding two steps showed that the algorithm DTW is capable in recognizing patterns reliably. The third step is the final stress test: All available anomaly curves

are added to the scoring data set, though they are not represented in the sketches. Additionally, the usability for the machine personnel will be evaluated.

Specifics of Test Data

In the manual classification process, in total 157 patterns could not be assigned to any sketch. In 47 occurrences a distinctly different pattern was found, the other 110 patterns were assigned to the category "unclear". We try to reduce this number in two steps:

1. In order to receive all patterns and stress the algorithm, the anomaly detection in Section 6.3.1 was chosen rather generously. Thus, some of those patterns are simply an overreaction of the Anomaly Algorithm. Choosing the criterion slightly stronger to a more realistic level, 20 curves of the category "unclear" can be ignored for the following.

2. Additionally, the remaining patterns were reevaluated if they can be matched to any of the existing sketches. Those patterns were marked with the combination of the sketch name and "unclear". In total, 19 curves were assigned to Synchronization, six to Control and three to Imbalance Error.

Summarized, 106 curves are not represented in the sketches, and 28 curves can be mapped with some uncertainty. This is a special stress test for the algorithm, as almost $50\,\%$ of the 285 curves are not represented in the sketches.

Evaluation Perspective

For this evaluation, we change the perspective from the algorithmic side to the machine personnel side. For the machine personnel, it is most important that he can trust the error description proposed by the algorithm. He prefers to not receive any detailed error description than a false one. This has a big influence on how we set up the classification and the evaluation. We propose following procedure: For $\Delta\,\%$ of the curves, the machine personnel will receive a detailed error description. The rest $(1-\Delta)\,\%$ of the curves are being marked as "unclear" by the algorithm, and the machine personnel will just receive the message "anomaly" without any error description. As the machine personnel is interested in the correctness of the received error descriptions, the accuracy measure will just take these classifications into account.

This view leads to two interesting questions. The first concerns the definition due to which patterns are classified as "anomaly" or not. For that, we propose to take the value of the DTW measure into account. The smaller the distance to the

most similar sketch, the more likely the pattern does resemble that sketch. Thus, classifications with small DTW measures should receive an error description, ones with big should be classified as "anomaly". The $\Delta\,\%$- quantile can be directly taken as the Δ-threshold.

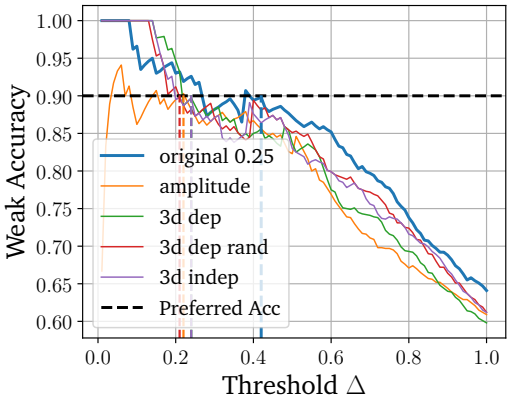
The second question deals with the explicit value of the threshold Δ. We consider a two step approach. In the first step, the correctness of the given error descriptions should exceed a specific value. In the following, we set the preferred Weak Accuracy to roughly $90\,\%$. In the second step, after fixing a value of Δ, the number of unclear patterns which receive an error description is analyzed, and checked if it can be minimized with a small loss in accuracy. In the best case scenario, all unclear patterns trigger the message "anomaly", and all error descriptions are correct.

Results

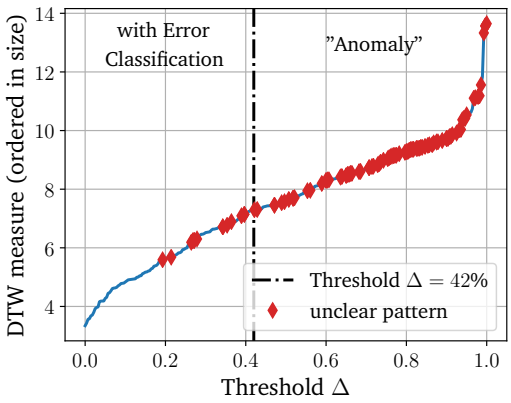
In the evaluation, we use the five winner setups of Step 2 of this study. In order to determine Δ, we plot the Weak Accuracy of all five setups in dependence of the chosen threshold Δ, as shown in Figure 6.14 a. Surprisingly the accuracy of most setups shows an almost linear dependency to the threshold. Evaluating the preferred Weak Accuracy of about $90\,\%$, the clear winner is DTW without feature extraction and a maximum distortion of 0.25 handovers. The preferred Weak Accuracy of about $90\,\%$ can be reached in this setup by thresholds up to about $\Delta = 42\,\%$. Marking this threshold in an plot of sorted DTW measures (Figure 6.14 b), one can clearly see that just very few unclear or new patterns (marked in orange and gray) lie underneath the chosen threshold. In more detail, just 11 of the unclear patterns receive mistakenly an error description.

Achieving an accuracy of $89.38\,\%$ with $\Delta = 42\,\%$ is a pretty extraordinary result, considering that almost $50\,\%$ of the patterns were categorized as "unclear". Thus, the algorithm reproduces the manual classification almost perfectly. It is also noteworthy, that the Strong Accuracy with $86.4\,\%$ is almost as high as the Weak Accuracy. The Δ-threshold has the additional effect, the cases in which the algorithm could not decide between two error cases are filtered out now.

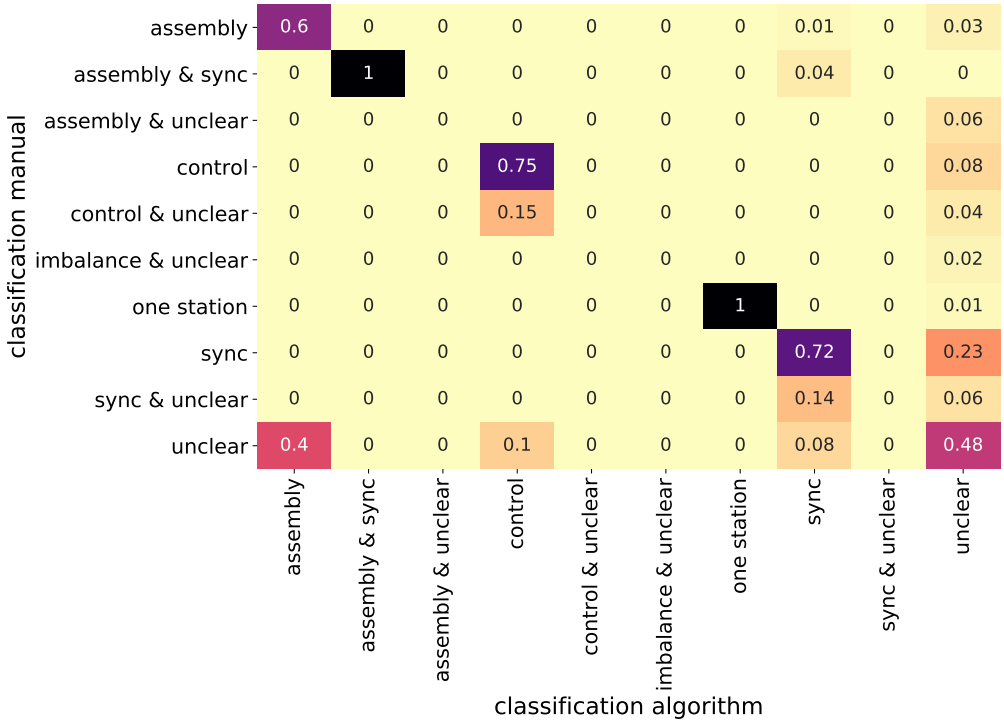
In order to get more into detail, we want to emphasize different aspects of the confusion matrix (Figure 6.14 c). It is normalized to the predictions (columns), in order to represent the view of the machine personnel. The category "unclear" summarizes all patterns, which are not represented in the sketches. One can see at first glance that all predictions about the error cases "One Station" and "Assembly & Sync" were correct and can be trusted fully. In the error case "Control" and "Sync" even the patterns, which were added with the reevaluation (e.g. control & unclear) are categorized correctly. The 11 unclear patterns, which landed un-



(a) The Weak Accuracy for the five DTW setups in dependency of the chosen threshold Δ. The preferred accuracy of about $90\,\%$ can be reached for the setup without feature extraction and a max distortion of 0.25 for thresholds up to $\Delta = 42\,\%$.



(b) All patterns ordered in size of their DTW measure for the setup. The patterns not represented in the sketches are marked in red. With choosing $\Delta = 42\,\%$, 11 non-represented curves are underneath the threshold.



(c) Confusion matrix for DTW with maximum distortion of 0.25 handovers, $\delta_{\max} = 5\%$, and $\Delta = 42\,\%$. The results are normalized by the predicted values. With the exception of "Assembly", the classifications of the algorithms reach a accuracy of more than $90\,\%$ for every fault category.

Figure 6.14: Classification results for the realistic setup with about $50\,\%$ patterns, which could not be matched to any sketch.

derneath the Δ-limit, are split up over the three classes of "Control", "Assembly" and "Sync". Thus, all three classes show some unreliability of the prediction. For "Control" and "Sync" this unreliability is in the order of $10\,\%$. In the class of "Assembly", the order is quite a bit higher with $40\,\%$, the machine personnel cannot trust this classification fully. Additional to the unclear patterns, there is just one misclassification between the sketches: One "Assembly" error case was classified as "Sync".

Summarizing, the stress test reached an extraordinary accuracy of $89.38\,\%$ for a threshold of $\Delta = 42\,\%$. Considering the high ratio of unclear data, and the training data being sketches instead of error data, the DTW setup lead to surprisingly satisfying results.

6.5 Feedback and Retraining

The special charm of the Expert- and Physics driven error sketch recognition is the fast feedback loop to the experts, and the fast retraining and improvement of the model. In the following Section, the newly found patterns will undergo this cycle.

6.5.1 Discussing new patterns with experts

As already briefly noted in the last Section, the classification person already detected new patterns in the data. Further analysis of those and the unclear patterns reveal six new patterns. Examples are shown in Figure 6.15. We give them rather descriptive names (from top right to bottom left): "Two Station", "Three Station", "Once Linear", "Twice Linear", "Linear & Belly" and "High Frequency".

Discussing the physical origin of those six patterns with the experts reveal very interesting findings:

1. The patterns "Twice linear", "Linear & Belly" and "One Linear" show a very clear split into two halves. This is an indication that they are caused by the two halves of the star. Thus, these patterns are simply other appearances of the assembly error, and can be all combined to the error case "Assembly error".

2. The "High Frequency" pattern allows a match to several of the possible error cases discussed in Section 6.2.1. Motor-, Gear- or Bearing-Failures are expected to show a high frequency patterns in the current. Without a detailed analysis, the specific origin cannot be determined. The fact that high frequency patterns appear is very promising as it could allow to detect those error cases in advance.

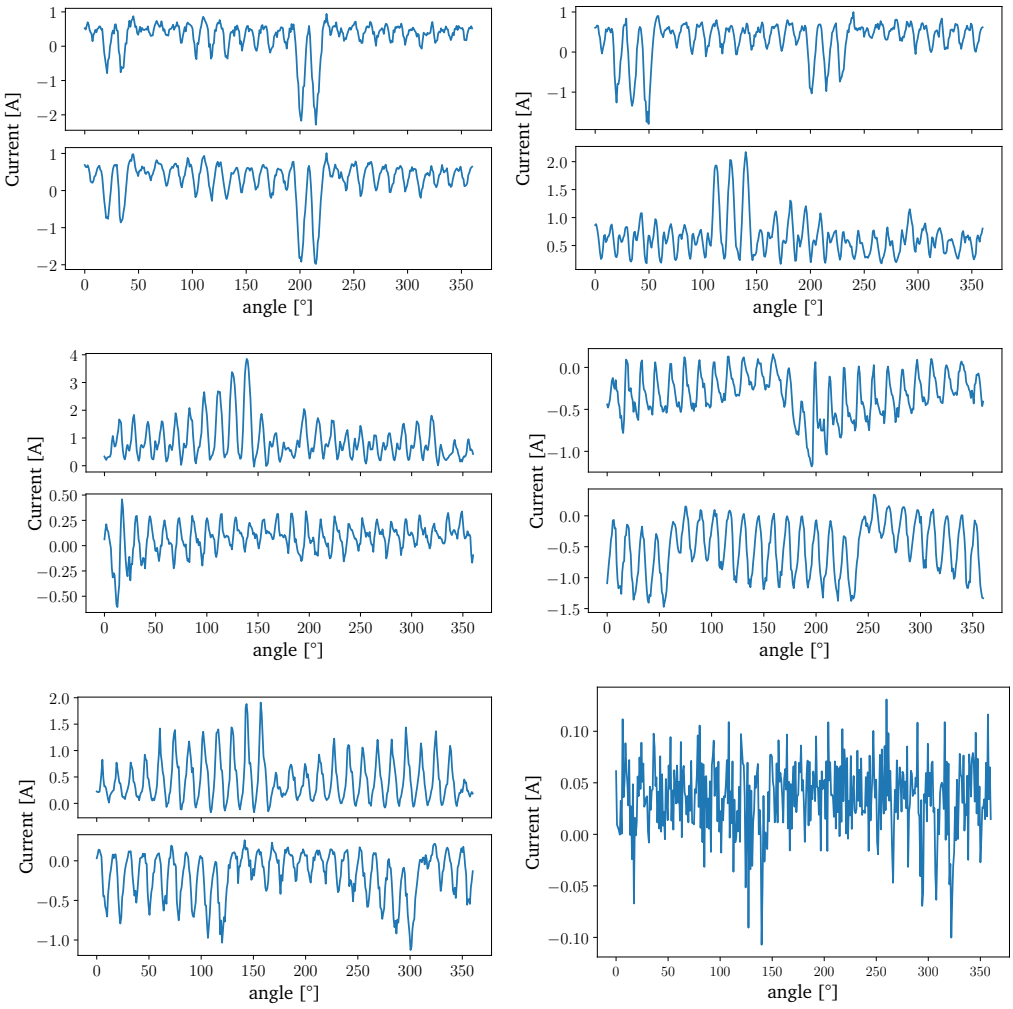


Figure 6.15: Examples of the newly detected patterns (from top left, to bottom right): “Two Station”, “Three Station”, “Once Linear”, “Twice Linear”, “Linear & Belly” and “High Frequency”.

3. The discussion about the patterns “Two Station” and “Three Station” was most intense, as no physical origin of those patterns could be identified. The origin has to be connected with handovers at specific stations as the distance between the groups of peaks is the distance between the two handover (in & out). But this particular behavior of the machine is not expected by the known physics of the machine. The origin of the patterns will be examined further together with a bottler.

In summary, the feedback-loop to the experts is very fruitful as further error patterns for existing error cases were identified. Additionally, the deployment of the algorithm in a production line allows the experts to learn more about further error cases.

6.5.2 Transform patterns into new sketches

The first sketches in Section 6.2.1 were created by using the freehand sketches of the experts, in combination with the physical knowledge. This time, real error patterns are available. These new patterns can be added in two different ways:

1. The error patterns are added directly to the collection of the patterns. This allows a completely automated self-learning process without any manual interaction.

2. The characteristics of the error patterns are extracted, and are joined with the physical knowledge in order to receive a computer-generated sketch. The manual effort has several advantages. At first, simplified patterns are more likely to generalize over several bottlers. Secondly, different variations of the patterns (e.g. randomized) can be created, which allows a faster learning curve. Thirdly, a script allows an easy transfer of the gained knowledge to different star sizes and setups.

Here, we will choose the Option 2, as the generalization of the model to different production lines is most important. The pattern “High Freq” will not be considered, as it is difficult to create an expressive sketch for a high frequency pattern with randomized amplitudes. Figure 6.16 show examples of the created sketches for the five patterns. All created patterns go through the data augmentation procedure described in Section 6.3.2.

Additionally, analyzing the real error patterns for “Control” allows to add another sketch in order to represent different variations of this error case. The additional sketch is also depicted in Figure 6.16.

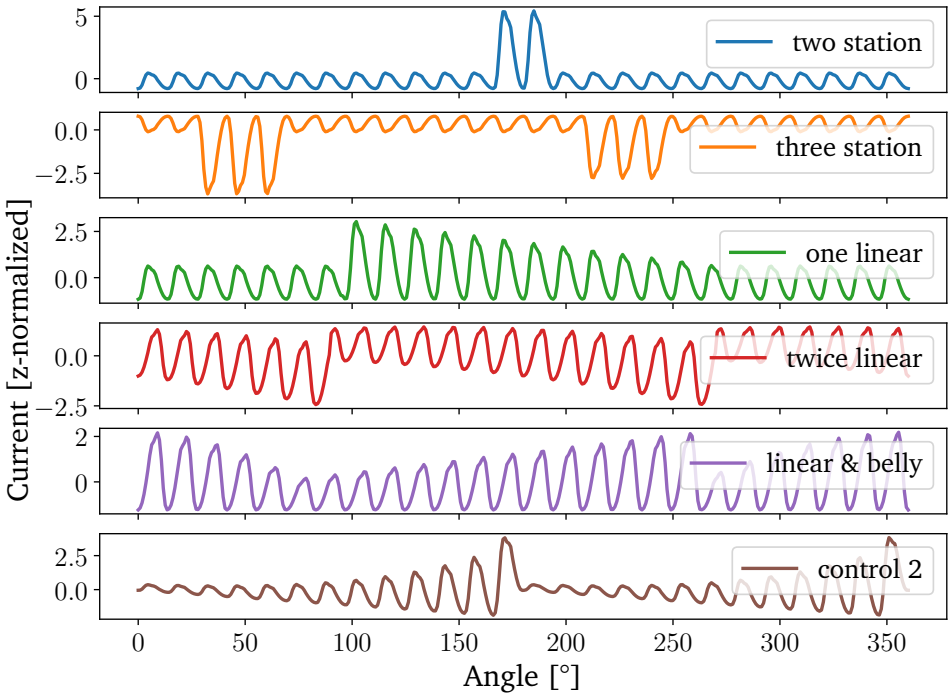


Figure 6.16: Examples of sketches for the new error patterns.

6.5.3 Study Setup

For this study, all available sketches are used. In total 33 sketches are included as representation of 10 error patterns for 7 error descriptions. As the previous manual labeling process just considered 5 error sketches, the labeling process is repeated. The anomaly detection is slightly modified in order to avoid non-anomalous curves in the data, following the leanings of Section 6.4.3. As further data was acquired in the mean time, in total 377 distinctly different anomalous curves are detected.

Equivalent to the first part, the classification person can choose up to two error descriptions in case the pattern is not clearly assignable to one pattern. Both classifications are treated equally. The amount of chosen combinations are shown in Figure 6.17. The number of unclear patterns is reduced substantially to 31 - in comparison to 106 in the last section. Nevertheless, in 51 cases the classification person was unsure if the pattern resembles one of the sketches or not. Summing up, even after adding further error sketches, about one in five patterns cannot be assigned clearly to one sketch. In general, the error cases “Synchronization Error” and “Assembly Error” are happening by far more often than the other error cases. This should be taken into consideration when evaluating the accuracy.

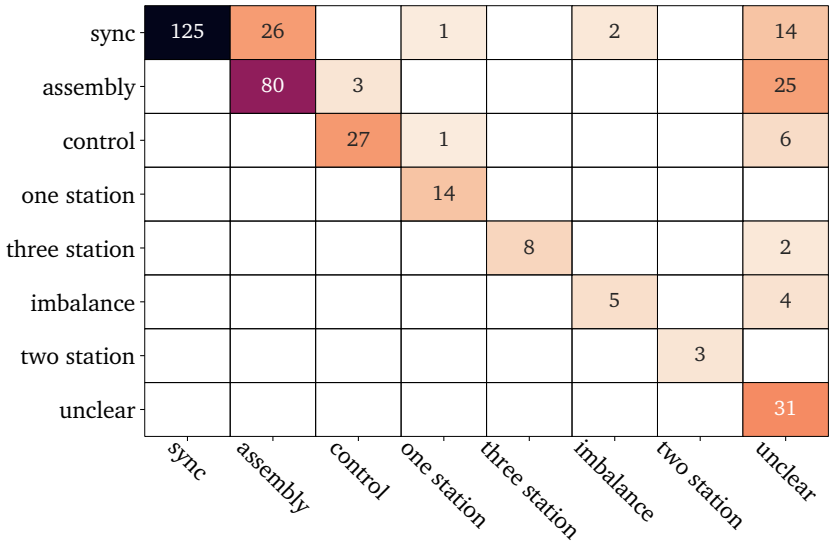


Figure 6.17: Number of patterns, which are classified into the eight error descriptions and the combinations of them. On the diagonal are the number patterns that have been assigned to only one error. On the off-diagonal are the number of pattern, which have been assigned to two patterns.

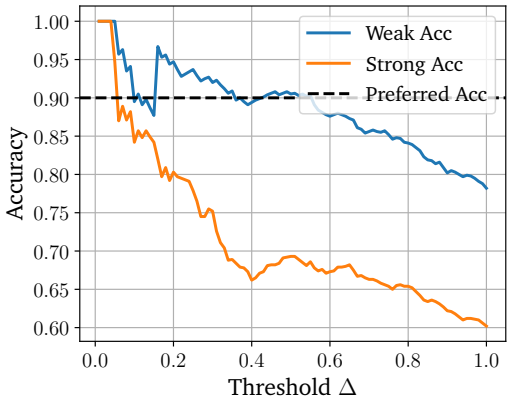
As algorithmic setup, the DTW with maximum distortion of 0.25 handovers and $\delta_{\text{max}} = 5\%$ is used, which was the winning algorithm of the first part.

6.5.4 Results

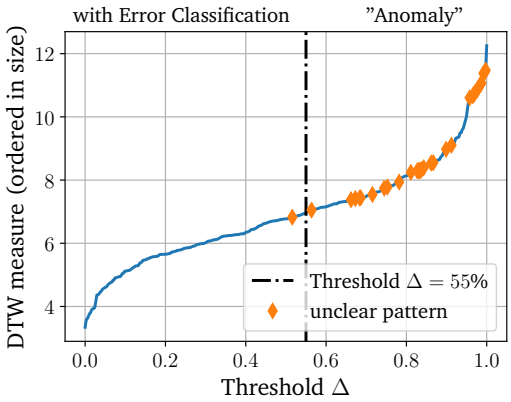
The evaluation in this part is happening completely equivalent to the evaluation of the five sketches in Section 6.4.3, which was centered around the machine personnel. Equivalent to above, only $\Delta\,\%$ of the curves will receive an error description.

In order to determine the threshold Δ, we evaluate the accuracies in dependence of Δ (Figure 6.18 a). The goal of $90\,\%$ Weak Accuracy can be achieved for thresholds below $\Delta \leq 55\,\%$. Just one curve with the classification "unclear" can be found underneath that threshold (Figure 6.18 b), thus the threshold can be fixed at $\Delta = 55\,\%$. This implies that the machine personnel receives about $30\,\%$ more error descriptions than before the retraining process with keeping the accuracy constant.

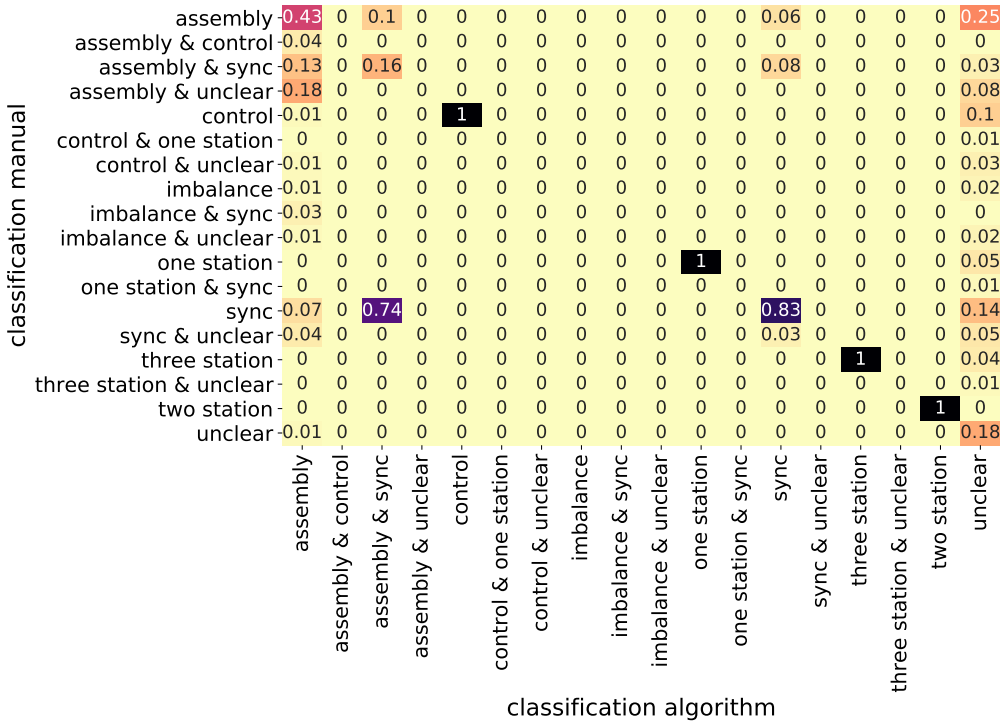
Going into detail in the confusion matrix (Figure 6.18 c), four of the seven classes reach an accuracy of $100\,\%$, namely "Control", "One Station", "Two Station" and "Three Station". The class "Imbalance" was never detected by the algorithm. The Weak Accuracy of $94\,\%$ has improved in comparison to the last Section. The only class which causes big problems is the class "Assembly" - about one in



(a) The Weak and the Strong Accuracy depend on the chosen threshold Δ. The preferred accuracy of about $90\,\%$ can be reached up to $\Delta = 55\,\%$.



(b) All patterns ordered in size of their DTW measure with the patterns not represented in the sketches marked in orange. Just one unrepresented pattern is falling underneath the threshold of $\Delta = 55\,\%$



(c) Confusion matrix for $\Delta = 55\,\%$ normalized by the predicted values. With the exception of "Assembly", all classes reach an accuracy above $94\,\%$.

Figure 6.18: Classification results for the retrained setup. About $20\,\%$ of the patterns could not be matched to any sketch.

four classifications is completely incorrect. Considering the sketches, this class is the trickiest one, as there are a lot of variations which do not show such an unambiguous pattern like for instance “One station”.

Summing up, with seven error classes an accuracy above $90\,\%$ could be easily achieved with choosing $\Delta = 55\,\%$. Except for the class “Assembly”, all classes result in extraordinary accuracies. With ignoring this class, Δ could be easily set to a higher value. Regarding $20\,\%$ of the patterns being assigned to “unclear”, $\Delta = 55\,\%$, is a pretty good result.

In terms of calculation time, adding the additional sketches increases the calculation time significantly to about 27.1 sec. This is in an acceptable range in regard to the fixed limit of 60 sec, nevertheless a shorter calculation time would be preferred.

6.6 Optimize Scoring Time

In a final step, we evaluate if the scoring time of the algorithm (in the last run about 27.1 sec) can be optimized with keeping the accuracy roughly constant. As the calculation is happening as one of many on a resource-limited edge device, any reduction in calculation time is preferred in order to guarantee the overall stability of the edge device.

We propose three different kinds of optimizations. The summary of all approaches can be found in Table 6.3.

6.6.1 DTW with ψ-Relaxation

A huge increase in calculation time originates in the high number of translations of each sketch. In this study, every sketch is being translated 80 times. An optimization could be achieved, if the DTW does not need to match every pattern to all translations, but to just one sketch, which contains all translations. This long sketch can be easily achieved by repeating the sketch twice behind each other, as depicted in Figure 6.19.

This approach cannot be handled by the standard DTW algorithm, as that one has the restriction that start and end points have to be matched together. A modification called ψ-relaxation weakens this constraint [101]. Up to ψ start and end points of a sequence can be ignored if this leads to a lower measure. In this case, we set ψ to the original length of the sketch.

As shown in Table 6.3, the DTW with ψ-Relaxation lead to a slight decrease in both Weak Accuracy and Strong Accuracy. More important, the calculation time with 35.6 sec, though, does not show any improvement to the previous approach.

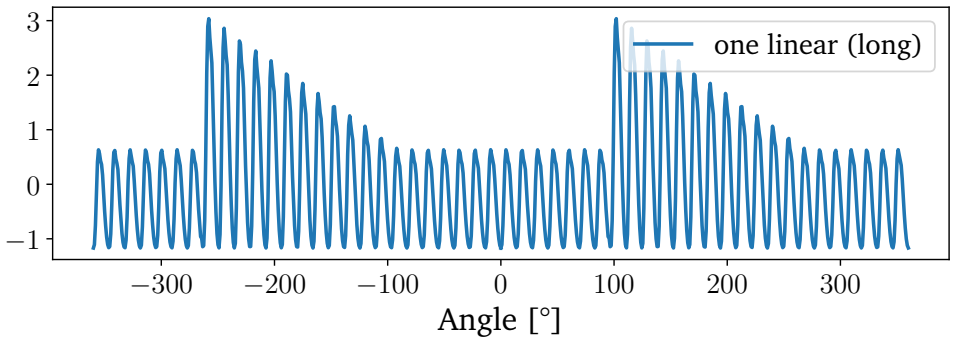


Figure 6.19: Example of a long sketch, which repeats the sketch twice. By doing this, all possible translations are contained in the sketch.

Due to scaling of the order $O(n^2)$ the increased length destroys the advantage of reducing the number of translations.

6.6.2 Pre-Selection via Euclidean distance

The second approach tries to reduce the calculation time by introducing a two-step system. In a first step, a resource-efficient measure presorts the sketches and translations, and selects the most promising ones. In a second step, the more resource-intense DTW measure is performing a fine-tuning of the preselected sketches and chooses the final winner(s).

As already seen in Figure 6.12 in Section 6.4.1, the resource-efficient Euclidean distance is able to detect basic similarities but fails in considering the details. Therefore, it meets all necessary requirements for the first step. The Euclidean measure evaluates all 2.640 sketches (33 sketches with each 80 translations) and marks the TOP n sketches. Those are reconsidered by the DTW algorithm.

The results in Table 6.3 look very promising. Taking the TOP 200 (= removing $92.4\,\%$ of sketches in the first step), the calculation time reduces substantially by $63.5\,\%$ with just a small loss of Weak Accuracy of $1.4\,\%$. Interestingly, the Strong Accuracy even improves by $1.4\,\%$. With $n = 50$ and $n = 100$, the calculation time can be reduced further by additional $5.5\,\%$ with a marginal loss in accuracy of about $1.0\,\%$.

6.6.3 Remove Duplicates in Sketches

The third approach removes duplicates, which are caused by the translation of the sketches. In sketches like “Control” or “Twice Linear” a pattern is repeated twice.

Setting	TOP n	Weak Acc $\delta_{\text{max}} = 5\%$	Strong Acc $\delta_{\text{max}} = 5\%$	Calc Time [sec]
normal	-	90.3%	68.6%	27.1 sec
ψ-Relaxation	-	86.0%	65.2%	35.6 sec
Pre-Selection: Eucl.	50	87.9%	69.1%	8.4 sec
	100	88.4%	69.1%	9.1 sec
	200	88.9%	70.0%	9.9 sec
Pre-Selection: Eucl & Remove Duplicates	50	87.9%	69.1%	7.4 sec
	100	88.4%	69.1%	8.1 sec
	200	88.9%	69.6%	9.0 sec

Table 6.3: Results for optimizing the scoring time. The combination of preselecting via euclidean distance and removing duplicate sketches, reduces the time by $66.8\,\%$ with keeping the accuracy constant. The maximum distortion is kept constant at 0.25 handovers.

For those sketches just half of the translations are needed to represent all possible translations.

We remove all translations, which correlate at least to $98.5\,\%$ to another translation. The number of sketches decreases by $15.2\,\%$ to a total of 2.240 sketches. Combining this improvement with the previous optimization leaves the accuracy roughly unchanged, but decreases the calculation time by about $10 - 12\,\%$. Thus, in the end, we are able to reduce the calculation time by a very substantial percentage of $66.8\,\%$ with keeping the accuracies in average constant. Three calculations can be now performed in the time of previously one.

6.7 Discussion and Outlook

Summarizing, the premiere of sketch-based error recognition in Predictive Maintenance was overly successful. For the first time, expert error sketches enriched by physical knowledge were used as training data in this field. The full concept including feedback-loop was performed for the field of error cases in small stars in the bottling industry.

Giving an overview, the experts were capable to identify five different sources of errors, and sketch them in an accurate way. The physics knowledge helped to transform them into more realistic sketches. The DTW algorithm proved itself as a very reliable method, which can recognize known sketches, and identify unknown patterns. As the percentage of unknown patterns (at the beginning $\sim 50\,\%$, after

retraining $\sim 20\,\%$) was especially high, the algorithm had to perform a real stress test. Classifying the TOP $42\,\%$ and $55\,\%$ (before and after retraining), the algorithm achieved extraordinary accuracies of about $90\,\%$. Performing some optimizations the scoring time could be reduced to less than 9.0 seconds, which is substantially lower than the goal of 30.0 seconds. Thus, it fulfills the basic requirement to be run on a resource-limited edge device.

The study confirms the premise that sketches support transfer learning naturally as the study was performed with in total six different stars from two breweries. Additionally, the study showed very illustrative that this approach also enables the experts to learn more about their machines, as for two error cases no mechanical explanation could be found yet.

As next steps, the algorithms and the feedback-loop needs testing in the real-time environment. Especially interesting will be the feedback of the machine personnel to the error cases “Two station” and “Three station”. A continuous learning process will be possible with the feedback of the breweries.

For a final proof of effective transfer learning, the algorithm should be also tested on non-beer producing bottlers.

Considering other use cases, DTW could come to its limits. In literature, the first pre-trained deep convolutional neuronal networks are used for time series classification, e.g. in [62]. This approach could lead to a more generalized solution, which is transferable over several use cases. The big question, if that approach can also handle sketches, opens a completely new exiting research topic.

7 | Conclusion and Outlook

This thesis proposes a new path for detecting and classifying machine errors with only a very small training data set. The approach was tested and developed on the field of transport error cases in filling machines, which are one of the most common error cases in bottling lines. For monitoring the machines, the electrical current of the driving motors was used. Due to the low temporal resolution of the signal, it was combined with angular information of the transport star, in order to obtain statistical high-resolution patterns for star rotations. Those patterns were used throughout the studies.

The main challenge is posed by the very small training data set. Here, "very small" is understood in the smallest way possible: The training data consists of just one healthy rotation pattern. All additional information about error cases is introduced by simplified physical models and expert knowledge. The approach consists of three steps:

1. Physical understanding:
   A basic physical understanding of the machine and its error cases is needed. An analytical physical model (as in Chapter 3) can help with that.

2. One-shot semi-supervised anomaly detection:
   Error cases in the data are detected by anomaly detection. Due to the minimal training set, semi-supervised methods are chosen, which can learn in a one-shot manner (see Chapter 5). Due to the large number of possible preprocessing steps and algorithms (as introduced in Chapter 4), physical intuition helps to narrow down suitable algorithms.

3. Physics- and expert-driven error sketch recognition:
   For classifying the detected error cases, domain experts are asked to sketch characteristics of different error cases. Those are enriched with physical knowledge in order to enhance the similarity with real-world error cases. Comparing a detected anomaly with those sketches allows a direct interpretation of the anomaly (see Chapter 6).

This novel approach was performed on the transport error cases in beer fillers with great success. For the anomaly detection and sketch recognition, different algorithmic setups were compared. Thereby, the following overly advantageous characteristics could be shown:

Successful Transfer Learning Anomaly detection and sketch recognition support transfer learning to new stars and machines naturally. Only one healthy pattern is needed for the retraining. Additionally, overfitting is not a concern for this approach.

Uncertainty of algorithm The anomaly detection and the sketch recognition were both implemented in a manner allowing them to express a degree of certainty, which is essential for the overall approach.

Detection of new error cases Due to the combination of machine learning with physical and expert knowledge, it is possible to detect new error cases and integrate them smoothly into the algorithmic pipeline. This is not possible with neither purely physical simulations, nor supervised machine learning algorithms, as both are restricted to already known error cases. This enables a new manner for the machine designers and the machine operators to learn more about their machines.

This study paves the way for further studies and use cases, which otherwise suffer from very little training data. In terms of bottling lines, the procedure could be applied to further types of fillers or other machines and processes. But it is not restricted to the bottling industry, as other fields involving custom machine construction suffer the same effects, e.g. constructors of vacuum furnaces or paper production machines.

To take the topic a step further, automated selection of algorithms would be preferable. In this study, different algorithms were tested on manually-labeled data sets in order to show their functionality. For semi-supervised anomaly detection, extensive literature research could suggest suitable algorithms based on the training data and physical properties of the machine. A similar approach was already performed by Fulcher [37] for the case of labeled time series data sets. For sketch recognition, it is conceivable that smart evaluation of the sketches could already restrict the algorithmic setups. For instance, algorithms should be able to distinguish sketches reliably, including after adding noise to the sketches.

Concluding, combining physics with machine learning can unlock the challenging field of very small data sets, and shows a way towards Predictive Maintenance in the field of special machinery.

Appendix

Solve Lagrange formalism - Part 2

As addition to Section 3.4.3, we solve the Lagrangian for the second time window $t \in [\frac{\varepsilon_1 + \beta}{v} + \Delta t_{\text{delay}}, \frac{\varepsilon_1 + \varepsilon_2}{v}]$, which represents the angle area of $\alpha \in [\beta, \varepsilon_2]$. For a better overview, we rewrite at first the two constraints g_2 and g_4 from Eq. 3.8 and Eq. 3.10:

$$g_2(\alpha, t) = (\alpha - \varepsilon_2) \cdot \delta(t - \frac{\varepsilon_2 + \varepsilon_1}{v}) \tag{7.1}$$

$$g_4(\alpha, t) = (\alpha - \beta) \cdot \delta(t - \frac{\varepsilon_1 + \beta}{v} - \Delta t_{\text{delay}}). \tag{7.2}$$

Setting up the Lagrangian formalism with those two constraints leads to

$$m\,\ddot{\alpha} = \lambda_2\,\delta(t - \frac{\varepsilon_2 + \varepsilon_1}{v}) + \lambda_4\,\delta(t - \frac{\varepsilon_1 + \beta}{v} - \Delta t_{\text{delay}}). \tag{7.3}$$

The equation of motion is gained by integrating twice over time:

$$\dot{\alpha}(t) = \lambda_2\,\theta(t - \frac{\varepsilon_2 + \varepsilon_1}{v}) + \lambda_4\,\theta(t - \frac{\varepsilon_1 + \beta}{v} - \Delta t_{\text{delay}}) + b_1 \tag{7.4}$$

$$\begin{aligned}\alpha(t) = {} & \lambda_2\,(t - \frac{\varepsilon_2 + \varepsilon_1}{v})\,\theta(t - \frac{\varepsilon_2 + \varepsilon_1}{v}) \\ & + \lambda_4\,(t - \frac{\varepsilon_1 + \beta}{v} - \Delta t_{\text{delay}})\,\theta(t - \frac{\varepsilon_1 + \beta}{v} - \Delta t_{\text{delay}}) \\ & + b_1 t + b_2\end{aligned} \tag{7.5}$$

with θ being the Heaviside step function and using $\int dt\,\delta(t-a) = \theta(t-a)$ and $\int dt\,\theta(t-a) = (t-a)\,\theta(t-a)$.

The equation of motion can be solved by inserting the Eq. 7.5 into the constraint 4:

$$\begin{aligned}&\Big(\lambda_2\,(t-\frac{\varepsilon_2+\varepsilon_1}{v})\,\theta(t-\frac{\varepsilon_2+\varepsilon_1}{v})\\ +\lambda_4\,(t-\frac{\varepsilon_1+\beta}{v}-\Delta t_{\text{delay}})\,\theta(t-\frac{\varepsilon_1+\beta}{v}-\Delta t_{\text{delay}})&\\ &+b_1t+b_2-\beta\Big)\cdot\delta(t-\frac{\varepsilon_1+\beta}{v}-\Delta t_{\text{delay}})=0.\end{aligned}\tag{7.6}$$

This equation is fulfilled for all $t\neq\frac{\varepsilon_1+\beta}{v}+\Delta t_{\text{delay}}$ and leads for $t=\frac{\varepsilon_1+\beta}{v}+\Delta t_{\text{delay}}$ to:

$$\begin{aligned}b_1(\frac{\varepsilon_1+\beta}{v}+\Delta t_{\text{delay}})+b_2-\beta=0&\\ \Rightarrow b_2=\beta-b_1\,(\frac{\varepsilon_1+\beta}{v}+\Delta t_{\text{delay}}).&\end{aligned}\tag{7.7}$$

Inserting the equation of motion into constraint 2 yields to

$$\begin{aligned}&\Big(\lambda_2\,(t-\frac{\varepsilon_2+\varepsilon_1}{v})\,\theta(t-\frac{\varepsilon_2+\varepsilon_1}{v})\\ +\lambda_4\,(t-\frac{\varepsilon_1+\beta}{v}-\Delta t_{\text{delay}})\,\theta(t-\frac{\varepsilon_1+\beta}{v}-\Delta t_{\text{delay}})&\\ &+b_1t+b_2-\varepsilon_2\Big)\cdot\delta(t-\frac{\varepsilon_2+\varepsilon_1}{v})=0.\end{aligned}\tag{7.8}$$

This is again fulfilled for all $t\neq\frac{\varepsilon_2+\varepsilon_1}{v}$. For $t=\frac{\varepsilon_2+\varepsilon_1}{v}$ and Eq. 7.7 we receive a definition for λ_4:

$$\begin{aligned}\lambda_4\,(\frac{\varepsilon_2+\varepsilon_1}{v}-\frac{\varepsilon_1+\beta}{v}-\Delta t_{\text{delay}})+b_1\frac{\varepsilon_2+\varepsilon_1}{v}+\beta-b_1\,(\frac{\varepsilon_1+\beta}{v}+\Delta t_{\text{delay}})-\varepsilon_2=0&\\ \Rightarrow\lambda_4=\frac{\varepsilon_2-\beta}{\frac{\varepsilon_2-\beta}{v}-\Delta t_{\text{delay}}}-b_1.&\end{aligned}\tag{7.9}$$

Replacing λ_4 and b_2 according to Eq. 7.9 and Eq. 7.7 into 7.5, leads with a few simplifying steps directly to the final equation of motions for the time frame of $t\in[\frac{\varepsilon_1+\beta}{v}+\Delta t_{\text{delay}},\frac{\varepsilon_1+\varepsilon_2}{v}]$:

$$\alpha(t)=\frac{\varepsilon_2-\beta}{\frac{\varepsilon_2-\beta}{v}-\Delta t_{\text{delay}}}\,(t-\frac{\varepsilon_1+\beta}{v}-\Delta t_{\text{delay}})+\beta\tag{7.10}$$

$$\dot\alpha(t)=\frac{\varepsilon_2-\beta}{\frac{\varepsilon_2-\beta}{v}-\Delta t_{\text{delay}}}.\tag{7.11}$$

Glossar

bottling line: Production line with a number of different machines in order to produce filled and labeled bottles on a pallet. For instance, this can include machines for cleaning, filling, labeling, packing and palletizing of bottles. The line setup depends a lot on the filled product.

carousel: Machine part, which rotates around its axis, and thereby transports bottles.

healthy data: Times in which the machine is operating normally without any defects in the machine.

error data: Times in which the machine is operating with a defect in the machine.

failure data: see error data

filler: Machine, which fills and closes bottles.

reference data: see healthy data

semi-supervised anomaly detection: Anomaly detection, which is based solely on healthy data.

star: see carousel

station: Part of the carousel, which holds the bottle during the transport process.

Words of gratitude

This work would have been not possible without the support of various people:

- I want to send a huge thank you to Prof. Elmar Lang, who gave me full freedom in the choice of topic, always supported my ideas, and always made time whenever I needed support. Thanks for all the good discussions and suggestions about different algorithms and strategies. Also, thanks to the whole work group, who inspired me and provided insights into a lot of different topics.

- Second, a very big thanks to my colleagues at Syskron GmbH. Thanks to Tobias Amann and the entire management team, who gave me the possibility to start the PhD part-time and gave me complete freedom to choose the topic. Thanks to Markus Zölfl for the support in the second half of the PhD, and especially for proof-reading my thesis. Big thanks also to my whole Data Science Team! Prof. Markus Goldhacker, who initiated the collaboration with Elmar Lang, and shared a lot of machine learning techniques with me. Marinus Bommer, Dominik Ramsauer and Franz Diebold, who were always good discussion partners and showed me a lot Python tricks. Kathrin Meindl and Florian Raff, who helped me with a lot of AWS infrastructure issues. Last, but not least, Markus Löffler, who always questioned my results from the user perspective, and thus helped me to develop algorithms that are not one valuable academically, but also applicable to industry.

- Third, a big thanks to all my best colleagues in Krones AG. First, the LCS use case team, who shared their machine experience with me, and answered every single question. Thanks for brainstorming about possible error cases and helping me to sketch and interpret them. Also thanks to Sebastian Langwieser, who provided me with a lot of insights into the control system of the motor and helped me to interpret the data correctly.

- Thanks to all three breweries for allowing me to use and evaluate their data. A special thanks to Rudi, for your good input on the error cases and sketches.

- Last, but not least, a very warm thank you to my parents, my sister and especially my partner Geoff. I always had full support of all of you! Thanks Geoff for always providing mental support, and for your continuous understanding, when I worked on my thesis during our free time (sometimes even on vacation)!

I look forward to invite each of you for a beer (or a different drink).

I want to close this thesis with a self-written haiku poem with a personal opinion about Machine Learning and Physics:

Territory war
Machine Learning and Physics?
Both flowering math!

Bibliography

[1] ADAM S.P., ALEXANDROPOULOS S.A.N., PARDALOS P.M., and VRAHATIS M.N., *No free lunch theorem: A review*, Springer Optimization and Its Applications **145**, 57 (2019). 37

[2] AHMAD S., LAVIN A., PURDY S., and AGHA Z., *Unsupervised real-time anomaly detection for streaming data*, Neurocomputing **262**, 134 (2017). 51, 61

[3] AHMED M., NASER MAHMOOD A., and HU J., *A survey of network anomaly detection techniques*, Journal of Network and Computer Applications **60**, 19 (2016). 38

[4] AN J. and CHO S., *Variational Autoencoder based Anomaly Detection using Reconstruction Probability*, Special Lecture on IE **2** (2015). 57

[5] ANKERST M., BREUNIG M.M., KRIEGEL H.P., and SANDER J., *OPTICS: Ordering Points to Identify the Clustering Structure*, ACM SIGMOD Record **28**, 49 (1999). 55

[6] BAIR E., *Semi-supervised clustering methods*, Wiley Interdiscip Rev Comput Stat. **5**, 349 (2013). 51

[7] BARBARA D., WU N., and JAJODIA S., *Detecting Novel Network Intrusions Using Bayes Estimators*, in *First SIAM International Conference on Data Mining*, 1–17 (2001). 53

[8] BARRETT R.T., *Fastener Design Manual*, NASA Reference Publication **1228**, 16 (1990). 18

[9] BBS AUTOMATION BLAICHACH GMBH, *Table Sliding Friction*, `https://www.schweizer-fn.de/stoff/reibwerte/reibwerte_gleitreibung.php` (Accessed: 2020-07-29). 18

[10] BENGIO Y. and LECUN Y., *Scaling Learning Algorithms toward AI*, Large-Scale Kernel Machines **34**, 1 (2007). 54

[11] BIKMUKHAMETOV T. and JÄSCHKE J., *Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models*, Computers and Chemical Engineering **138** (2020). 106

[12] BREUNIG M.M., KRIEGEL H.P., NG R.T., and SANDER J., *LOF: Identifying Density-Based Local Outliers*, in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 93–104, Association for Computing Machinery, New York, NY, USA (2000). 56

[13] BULKOWSKI T.N., *Encyclopedia of chart patterns*, John Wiley & Sons, 2nd ed. (2005). 106

[14] CASAS P., D'ALCONZO A., SETTANNI G., FIADINO P., and SKOPIK F., *POSTER: (Semi)-Supervised Machine Learning Approaches for Network Security in High-Dimensional Network Data*, in *2016 ACM SIGSAC Conference*, 1805–1807 (2016). 52

[15] CHALAPATHY R., MENON A.K., and CHAWLA S., *Anomaly Detection using One-Class Neural Networks*, arXiv:1802.06360v2 (2019). 54

[16] CHANDOLA V., BANERJEE A., and KUMAR V., *Anomaly Detection: A Survey*, ACM Computing Surveys **41**, 1 (2009). 38, 50, 51, 52, 53, 55, 56, 57, 58, 68

[17] CHAPMAN P., CLINTON J., KERBER R., KHABAZA T., REINARTZ T., SPSS C.S., and WIRTH R., *Step-by-step data mining guide*, SPSS inc **78**, 1 (2000). 40

[18] CHAUDHARY A., SZALAY A.S., and MOORE A.W., *Very Fast Outlier Detection in Large Multidimensional Data Sets*, in *Proceedings of ACM SIGMOD Workshop in Research Issues in Data Mining and Knowledge Discovery DMKD*, ACM Press (2002). 56

[19] CHEN X., GU L., LI S.Z., and ZHANG H.J., *Learning representative local features for face detection*, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, IEEE (2001). 49

[20] CHENG H., YANG L., and LIU Z., *Survey on 3D Hand Gesture Recognition*, IEEE Transactions on Circuits and Systems for Video Technology **26**, 1659 (2016). 105

[21] CHRIST M., BRAUN N., NEUFFER J., and KEMPA-LIEHR A.W., *Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)*, Neurocomputing **307**, 72 (2018). 50, 77

[22] CIVERA M., FILOSI C.M., PUGNO N.M., SILVESTRINI M., SURACE C., and WORDEN K., *Assessment of vocal cord nodules: A case study in speech processing by using Hilbert-Huang Transform*, in *Journal of Physics: Conference Series*, vol. 842 (2017). 46

[23] ČÍŽEK V., *Discrete Hilbert Transform*, IEEE Transactions on Audio and Electroacoustics **18**, 340 (1970). 46

[24] COOLEY J.W. and TUKEY J.W., *An Algorithm for the Machine Calculation of Complex Fourier Series*, Mathematics of Computation **19**, 297 (1965). 44

[25] CORRELL M. and GLEICHER M., *The semantics of sketch: Flexibility in visual query systems for time series data*, 2016 IEEE Conference on Visual Analytics Science and Technology, VAST 2016 - Proceedings 131–140 (2017). 106

[26] DANFOSS, *Orbital Motors: Type WG* (2018), `https://assets.danfoss.com/documents/56391/BC270080703001en-000101.pdf` (Accessed: 2020-11-30). 18

[27] DE STEFANO C., SANSONE C., and VENTO M., *To reject or not to reject: that is the question - an answer in case of neural classifiers*, IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews **30**, 84 (2000). 53

[28] ERFANI S.M., RAJASEGARAR S., KARUNASEKERA S., and LECKIE C., *High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning*, Pattern Recognition **58**, 121 (2016). 51, 54

[29] ESLING P. and AGON C., *Time-series data mining*, ACM Computing Surveys, Association for Computing Machinery **45**, 12 (2012). 38

[30] ESTER M., KRIEGEL H.P., SANDER J., and XU X., *A density-based algorithm for dis- covering clusters in large spatial databases with noise*, KDD-96 Proceedings 226–231 (1996). 55

[31] ETHERINGTON T.R., *Mahalanobis distances and ecological niche modelling: Correcting a chi-squared probability error*, PeerJ **2019**, 1 (2019). 52

[32] FALTERMEIER R., ZEILER A., TOMÉ A.M., BRAWANSKI A., and LANG E.W., *Weighted Sliding Empirical Mode Decomposition*, Advances in Adaptive Data Analysis **3**, 509 (2011). 62

[33] FENG Z., LIANG M., and CHU F., *Recent advances in time-frequency analysis methods for machinery fault diagnosis: A review with application examples*, Mechanical Systems and Signal Processing **38**, 165 (2013). 3

[34] FULCHER B.D., *Feature-based time-series analysis*, in G. Dong and H. Liu, eds., *Feature Engineering for Machine Learning and Data Analytics*, March, chap. 4, Taylor & Francis Group, Boca Raton, 1 ed. (2018). 49

[35] FULCHER B.D. and JONES N.S., *Highly comparative feature-based time-series classification*, IEEE Transactions on Knowledge and Data Engineering **26**, 3026 (2014). 49

[36] FULCHER B.D., LITTLE M.A., and JONES N.S., *Highly comparative time-series analysis: The empirical structure of time series and their methods*, Journal of the Royal Society Interface **10** (2013). 49

[37] FULCHER B.D., LUBBA C.H., SETHI S.S., and JONES N.S., *A self-organizing, living library of time-series data*, Scientific Data **7**, 1 (2020). 49, 144

[38] GAO H., LIANG L., CHEN X., and XU G., *Feature extraction and recognition for rolling element bearing fault utilizing short-time Fourier transform and non-negative matrix factorization*, Chinese Journal of Mechanical Engineering **28**, 96 (2015). 48

[39] GARRIDO R. and CONCHA A., *Inertia and friction estimation of a velocity-controlled servo using position measurements*, IEEE Transactions on Industrial Electronics **61**, 4759 (2014). 18

[40] GIORGINO T., *Computing and visualizing dynamic time warping alignments in R: The dtw package*, Journal of Statistical Software **31**, 1 (2009). 109

[41] GOLD O. and SHARIR M., *Dynamic time warping and geometric edit distance: Breaking the quadratic barrier*, ACM Transactions on Algorithms **14** (2018). 110

[42] GOLDSTEIN M. and UCHIDA S., *A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data*, PLoS ONE **11** (2016). 38, 39, 50

[43] GRANEY B.P. and STARRY K., *Rolling element bearing analysis*, Materials Evaluation **70**, 78 (2012). 44, 114

[44] GRUBBS F.E., *Procedures for detecting outlying observations in samples*, Technometrics **11**, 1 (1969). 52

[45] GUPTA J., *One-class data description TAX* (2019), `https://en.wikipedia.org/wiki/File:One-class_data_description_TAX.png` (Accessed: 2021-04-19). 54

[46] HAN T., PENG Q., ZHU Z., SHEN Y., HUANG H., and ABID N.N., *A pattern representation of stock time series based on DTW*, Physica A: Statistical Mechanics and its Applications **550**, 124161 (2020). 107

[47] HARDY G., LITTLEWOOD J., and PÓLYA G., *Inequalities*, Cambridge University Press, Cambridge, UK (1952). 46

[48] HARRIS C.R. ET AL., *Array programming with NumPy*, Nature **585**, 357 (2020). 13

[49] HAWKINS D.M., *Identification of Outliers (Monographs on Applied Probability and Statistics)*, Springer Netherlands, 1 ed. (1980). 38

[50] HE Z., XU X., and DENG S., *Discovering cluster-based local outliers*, Pattern Recognition Letters **24**, 1641 (2003). 56

[51] HERMANS R., *Negative and positive skew diagrams* (2008), `https://commons.wikimedia.org/wiki/File:Negative_and_positive_skew_diagrams_(English).svg` (Accessed: 2021-04-03). 89

[52] HOU X., *A sketch recognition algorithm based on Bayesian network and convolution neural network*, Journal of Advanced Computational Intelligence and Intelligent Informatics **23**, 261 (2019). 105

[53] HUANG N.E., SHEN Z., LONG S.R., WU M.C., SNIN H.H., ZHENG Q., YEN N.C., TUNG C.C., and LIU H.H., *The empirical mode decomposition and the Hubert spectrum for nonlinear and non-stationary time series analysis*, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **454**, 903 (1998). 44

[54] HUI L., HAIPING W., and CHENMING Y., *A hybrid model for appliance classification based on time series features*, Energy and Buildings **196**, 112 (2019). 50

[55] HUNTER J.D., *Matplotlib: A 2D Graphics Environment*, Computing in Science & Engineering **9**, 90 (2007). 13

[56] HYNDMAN R.J., WANG E., and LAPTEV N., *Large-scale unusual time series detection*, in *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 1616–1619, IEEE (2016). 50

[57] IBM, *IBM SPSS Modeler CRISP-DM Guide*, IBM Corporation 1–50 (2016). 40

[58] JENSEN K., *CRISP-DM Process Diagram* (2012), `https://commons.wikimedia.org/wiki/File:CRISP-DM_Process_Diagram.png` (Accessed: 2020-12-13). 40, 72

[59] JIANG F., ZHU Z., LI W., ZHOU G., and CHEN G., *Fault identification of rotor-bearing system based on ensemble empirical mode decomposition and self-zero space projection analysis*, Journal of Sound and Vibration **333**, 3321 (2014). 3

[60] KANDANAARACHCHI S., MUÑOZ M.A., HYNDMAN R.J., and SMITH-MILES K., *On normalization and algorithm selection for unsupervised outlier detection*, Data Mining and Knowledge Discovery **34**, 309 (2019). 50

[61] KANG X., HAJEK B., WU F., and HANZAWA Y., *Time series experimental design under one-shot sampling: The importance of condition diversity*, PLoS ONE **14**, 1 (2019). 106

[62] KASHIPAREKH K., NARWARIYA J., MALHOTRA P., VIG L., and SHROFF G., *ConvTimeNet: A Pre-trained Deep Convolutional Neural Network for Time Series Classification*, in *Proceedings of the International Joint Conference on Neural Networks*, 1–8, IEEE (2019). 141

[63] KHATUN S., MORSHED B.I., and BIDELMAN G.M., *A Single-Channel EEG-Based Approach to Detect Mild Cognitive Impairment via Speech-Evoked Brain Responses*, IEEE Transactions on Neural Systems and Rehabilitation Engineering **27**, 1063 (2019). 50

[64] KIM S.H., LEE H.S., KO H.J., JEONG S.H., BYUN H.W., and OH K.J., *Pattern matching trading system based on the dynamic time warping algorithm*, Sustainability (Switzerland) **10**, 1 (2018). 107

[65] KOHONEN T., *Self-organized formation of topologically correct feature maps*, Biological Cybernetics **43**, 59 (1982). 55

[66] KOKOSKA S. and ZWILLINGER D., *CRC Standard Probability and Statistics Tables and Formulae, Student Edition* (2000). 58

[67] KRONES AG, *Krones Fuellsysteme fuer Bier*, `https://www.krones.com/media/downloads/fuellsysteme_bier_de.pdf` (Accessed: 2020-11-13). 8, 112

[68] KRONES AG, *Krones Website*, `www.krones.com` (Accessed: 2020-11-20). 4, 8, 20

[69] LASZUK D., *Python implementation of Empirical Mode Decomposition algorithm* (2017), `https://github.com/laszukdawid/PyEMD` (Accessed: 2020-07-09). 76, 77

[70] LAVIN A. and AHMAD S., *Evaluating real-time anomaly detection algorithms - The numenta anomaly benchmark*, in *14th International Conference on Machine Learning and Applications (IEEE ICMLA'15)*, 38–44 (2015). 60, 61, 66, 68, 79, 81, 86

[71] LECUN Y., BOTTOU L., BENGIO Y., and HAFFNER P., *Gradient-based learning applied to document recognition*, in *Proceedings of the IEEE*, 2278–2324 (1998). 105

[72] LEE D.D. and SEUNG H.S., *Learning the parts of objects by nonnegative matrix factorization*, Nature **401**, 788 (1999). 49

[73] LEE D.D. and SEUNG H.S., *Algorithms for Non-negative Matrix Factorization*, in T.K. Leen, T.G. Dietterich, and V. Tresp, eds., *Advances in Neural Information Processing Systems 13*, 1, 556–562, MIT Press (2001). 48

[74] LEI Y., LIN J., HE Z., and ZUO M.J., *A review on empirical mode decomposition in fault diagnosis of rotating machinery*, Mechanical Systems and Signal Processing **35**, 108 (2013). 46

[75] LI M. and VITÁNYI P., *An Introduction to Kolmogorov Complexity and Its Applications*, Springer-Verlag New York, 3 ed. (2008). 58

[76] LIANG L., LIU F., LI M., HE K., and XU G., *Feature selection for machine fault diagnosis using clustering of non-negation matrix factorization*, Measurement: Journal of the International Measurement Confederation **94**, 295 (2016). 49

[77] LIU Z., BEN-BASAT R., EINZIGER G., KASSNER Y., BRAVERMAN V., FRIEDMAN R., and SEKAR V., *NitroSketch: Robust and general sketch-based monitoring in software switches*, in *Proceedings of the ACM Special Interest Group on Data Communication*, 334–350, Association for Computing Machinery, New York, NY, USA (2019). 106

[78] LOUTRIDIS S.J., *Damage detection in gear systems using empirical mode decomposition*, Engineering Structures **26**, 1833 (2004). 46

[79] LUBBA C.H., SETHI S.S., KNAUTE P., SCHULTZ S.R., FULCHER B.D., and JONES N.S., *catch22: CAnonical Time-series CHaracteristics*, Data Mining and Knowledge Discovery **33**, 1821 (2019). 49, 50, 77

[80] LV Y., YUAN R., and SONG G., *Multivariate empirical mode decomposition and its application to fault diagnosis of rolling bearing*, Mechanical Systems and Signal Processing **81**, 219 (2016). 46

[81] MACKAY D.J., *Information Theory, Inference, and Learning Algorithms*, vol. 13, Cambridge University Press (2003). 58

[82] MANNINO M. and ABOUZIED A., *Expressive Time Series Querying with Hand-Drawn Scale-Free Sketches*, in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, 1–13, Association for Computing Machinery, Montreal QC, Canada (2018). 106, 107

[83] MARAGOS P., KAISER J., and QUATIERI T., *On amplitude and frequency demodulation using energy operators*, IEEE Transactions on Signal Processing **41**, 1532 (1993). 46

[84] MCKINNEY W., *Data Structures for Statistical Computing in Python*, in S. van der Walt and J. Millman, eds., *Proceedings of the 9th Python in Science Conference*, 56–61 (2010). 13

[85] MEERT W., HENDRICKX K., and CRAENENDONCK T.V., *wannesm/dtaidistance*, `https://github.com/wannesm/dtaidistance`. 123

[86] MILLER C., NAGY Z., and SCHLUETER A., *A review of unsupervised statistical learning and visual analytics techniques applied to performance analysis of non-residential buildings*, Renewable and Sustainable Energy Reviews **81**, 1365 (2018). 50

[87] OKAWA M., *Template Matching Using Time-Series Averaging and DTW with Dependent Warping for Online Signature Verification*, IEEE Access **7**, 81010 (2019). 107, 110

[88] OKAWA M., *Online signature verification using single-template matching with time-series averaging and gradient boosting*, Pattern Recognition **102**, 107227 (2020). 107

[89] OUYANG S., HOSPEDALES T., SONG Y.Z., LI X., LOY C.C., and WANG X., *A survey on heterogeneous face recognition: Sketch, infra-red, 3D and low-resolution*, Image and Vision Computing **56**, 28 (2016). 105

[90] PAPAGEORGIOU D., BLANKE M., HENRIK NIEMANN H., and RICHTER J.H., *Online friction parameter estimation for machine tools*, Advanced Control for Applications: Engineering and Industrial Systems **2** (2020). 18

[91] PEDREGOSA F. ET AL., *Scikit-learn: Machine Learning in Python*, Journal of Machine Learning Research **12**, 2825 (2011). 13, 76

[92] PENROSE R., *A generalized inverse for matrices*, Mathematical Proceedings of the Cambridge Philosophical Society **51**, 406 (1955). 48

[93] QWERTYUS, *Illustration of approximate non-negative matrix factorization (NMF)* (2013), `https://commons.wikimedia.org/wiki/File:NMF.png` (Accessed: 2020-12-14). 48

[94] RAI A. and UPADHYAY S., *Bearing performance degradation assessment based on a combination of empirical mode decomposition and k-medoids clustering*, Mechanical Systems and Signal Processing **93**, 16 (2017). 3, 46

[95] RAKTHANMANON T., CAMPANA B., MUEEN A., BATISTA G., WESTOVER B., ZHU Q., ZAKARIA J., and KEOGH E., *Searching and mining trillions of time series subsequences under dynamic time warping*, Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining 262–270 (2012). 109

[96] ROUSSEEUW P. and DRIESSEN K., *A Fast Algorithm for the Minimum Covariance*, Technometrics **41**, 212 (1999). 53

[97] RUFF L., VANDERMEULEN R.A., GÖRNITZ N., DEECKE L., SIDDIQUI S.A., BINDER A., MÜLLER E., and KLOFT M., *Deep one-class classification*, 35th International Conference on Machine Learning, ICML 2018 **10**, 6981 (2018). 54

[98] SAKOE H. and CHIBA S., *Dynamic Programming Algorithm Optimization for Spoken Word Recognition*, IEEE Transactions on Acoustics, Speech, and Signal Processing **26**, 43 (1978). 110, 120

[99] SCHÖLKOPF B. and SMOLA A.J., *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA; .*, MIT Press, Cambridge, MA (2002). 38

[100] SHYU M.L., CHEN S.C., SARINNAPAKORN K., and CHANG L., *A novel anomaly detection scheme based on principal component classifier*, in *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining*, 171–179 (2003). 57

[101] SILVA D.F., BATISTA G.E., and KEOGH E., *Prefix and suffix invariant dynamic time warping*, Proceedings - IEEE International Conference on Data Mining, ICDM **16**, 2374 (2016). 138

[102] SINGH N. and OLINSKY C., *Demystifying Numenta anomaly benchmark*, in *2017 International Joint Conference on Neural Networks (IJCNN)*, 1570–1577 (2017). 61, 79

[103] SOHAIB M., KIM C.H., and KIM J.M., *A hybrid feature model and deep-learning-based bearing fault diagnosis*, Sensors (Switzerland) **17** (2017). 54

[104] SOUZA V.M.A., CHERMAN E.A., ROSSI R.G., and SOUZA R.A., *Towards Automatic Evaluation of Asphalt Irregularity Using Smartphone's Sensors*, in *International Symposium on Intelligent Data Analysis*, 322–333 (2017). 50

[105] SURACE C. and WORDEN K., *A novelty detection method to diagnose damage in structures: an application to an offshore platform*, In Proceedings of Eighth International Conference of Off-shore and Polar Engineering **4**, 64 (1998). 52

[106] SURACE C., WORDEN K., and TOMLINSON G., *A novelty detection approach to diagnose damage in a cracked beam*, Proceedings of the International Modal Analysis Conference - IMAC **3089**, 947 (1997). 52

[107] TABRIZI A., GARIBALDI L., FASANA A., and MARCCHESIELLO S., *A novel feature extraction for anomaly detection of roller bearings based on performance improved ensemble empirical mode decomposition and teager-kaiser energy operator*, International Journal of Prognostics and Health Management **6** (2015). 46, 76

[108] TABRIZI A., GARIBALDI L., FASANA A., and MARCHESIELLO S., *Early damage detection of roller bearings using wavelet packet decomposition, ensemble empirical mode decomposition and support vector machine*, Meccanica **50**, 865 (2015). 46

[109] TANG J., CHEN Z., FU A., and CHEUNG D., *Enhancing Effectiveness of Outlier Detections for Low Density Patterns*, in M. Chen, P. Yu, and B. Liu, eds., *Advances in Knowledge Discovery and Data Mining. PAKDD 2002. Lecture Notes in Computer Science, vol 2336*, 535–548, Springer, Berlin, Heidelberg (2002). 56

[110] TAX D.M. and DUIN R.P., *Support Vector Data Description*, Machine learning **54**, 45 (2004). 53

[111] PANDAS DEVELOPMENT TEAM T., *pandas-dev/pandas: Pandas* (2020), https://doi.org/10.5281/zenodo.3509134. 13

[112] TOBAR F.A., YACHER L., PAREDES R., and ORCHARD M.E., *Anomaly detection in power generation plants using similarity-based modeling and multivariate analysis*, Proc. of the American Control Conference (ACC) 1940–1945 (2011). 3

[113] TSINASLANIDIS P.E. and ZAPRANIS A.D., *Technical analysis for algorithmic pattern recognition* (2016). 106

[114] TUKEY J.W., *Exploratory Data Analysis*, Addison-Wesley (1977). 52

[115] VIRTANEN P. ET AL., *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nature Methods **17**, 261 (2020). 13

[116] W-TECH, *Regelverfahren für elektrische Antriebe*, http://www.servotechnik.de/fachwissen/regelung/f_beitr_00_602.htm (Accessed: 2021-01-19). 33

[117] WAN Y. and SI Y.W., *A hidden semi-Markov model for chart pattern matching in financial time series*, Soft Computing **22**, 6525 (2018). 106

[118] WANG C., MIU T.T., LUO X., and WANG J., *SkyShield: A sketch-based defense system against application layer DDoS attacks*, IEEE Transactions on Information Forensics and Security **13**, 559 (2018). 106

[119] WANG Y., YAO Q., KWOK J.T., and NI L.M., *Generalizing from a Few Examples: A Survey on Few-shot Learning*, ACM Computing Surveys **53**, 1 (2020). 106, 107, 108

[120] WASKOM M.L., *seaborn: statistical data visualization*, Journal of Open Source Software **6**, 3021 (2021). 13

[121] WEISSTEIN E.W., *Statistical Correlation*, `https://mathworld.wolfram.com/StatisticalCorrelation.html` (Accessed: 2021-06-14). 59

[122] WIRTH R. and HIPP J., *CRISP-DM : Towards a Standard Process Model for Data Mining*, in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, 29–39, Springer-Verlag London, UK (2000). 40

[123] WODECKI J., KRUCZEK P., BARTKOWIAK A., ZIMROZ R., and WYŁOMAŃSKA A., *Novel method of informative frequency band selection for vibration signal using Nonnegative Matrix Factorization of spectrogram matrix*, Mechanical Systems and Signal Processing **130**, 585 (2019). 48

[124] WÖSTMANN R., SCHLUNDER P., TEMME F., KLINKENBERG R., KIMBERGER J., SPICHTINGER A., GOLDHACKER M., and DEUSE J., *Conception of a Reference Architecture for Machine Learning in the Process Industry*, in *2020 IEEE International Conference on Big Data (Big Data)*, 1726–1735 (2020). 12

[125] WU Z. and HUANG N.E., *Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method*, Advances in Adaptive Data Analysis **1**, 1 (2009). 45

[126] XU F., SONG X., TSUI K.L., YANG F., and HUANG Z., *Bearing Performance Degradation Assessment Based on Ensemble Empirical Mode Decomposition and Affinity Propagation Clustering*, IEEE Access **7**, 54623 (2019). 46

[127] XU X., LIU H., and YAO M., *Recent Progress of Anomaly Detection*, Complexity **2019** (2019). 50, 51

[128] YE N. and CHEN Q., *An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems*, Quality and Reliability Engineering International **17**, 105 (2001). 52

[129] YESILBEK K.T. and SEZGIN T.M., *Sketch recognition with few examples*, Computers and Graphics (Pergamon) **69**, 80 (2017). 105

[130] YU D., SHEIKHOLESLAMI G., and ZHANG A., *FindOut: Finding Outliers in Very Large Datasets*, Knowledge And Information Systems 4 387–412 (2002). 55

[131] ZHANG X., LI X., LIU Y., and FENG F., *A survey on freehand sketch recognition and retrieval*, Image and Vision Computing **89**, 67 (2019). 105

[132] ZHAO X., PATEL T.H., and ZUO M.J., *Multivariate EMD and full spectrum based condition monitoring for rotating machinery*, Mechanical Systems and Signal Processing **27**, 712 (2012). 62

[133] ZOPE K., SINGH K., NISTALA S.H., BASAK A., RATHORE P., and RUNKANA V., *Anomaly Detection and Diagnosis in Manufacturing Systems : A Comparative Study of Statistical , Machine Learning and Deep Learning Techniques*, Proceedings of the Annual Conference of the PHM Society **11** (2019). 51, 52

Eidesstattliche Erklärung

1. Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; die aus anderen Quellen direkt oder indirekt übernommenen Daten und Konzepte sind unter Angabe des Literaturzitats gekennzeichnet.

2. Bei der Auswahl und Auswertung folgenden Materials haben mir die nachstehend aufgeführten Personen in der jeweils beschriebenen Weise entgeltlich/ unentgeltlich geholfen: -

3. Weitere Personen waren an der inhaltlich-materiellen Herstellung der vorliegenden Arbeit nicht beteiligt. Insbesondere habe ich hierfür nicht die entgeltliche Hilfe eines Promotionsberaters oder anderer Personen in Anspruch genommen. Niemand hat von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

4. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt.