Deutsches Zentrum
DLR für Luft- und Raumfahrt
German Aerospace Center

Hochschule Esslingen
University of Applied Sciences

Master's thesis

Anomaly detection for solar thermal parabolic trough power plants with artificial intelligence

Anomalie-Erkennung für
solarthermische Parabolrinnen-Kraftwerke
mithilfe künstlicher Intelligenz

at
University of Applied Sciences Esslingen

Author: **Josua Braun**
Study program: Applied computer science
Enrolment number: 760121

Date: 12.12.2020

First examiner: **Prof. Dr. rer. nat. Gabriele Gühring**
University of Applied Sciences Esslingen
Department of Fundamentals

Second examiner: **Prof. Dr.-Ing. Reiner Marchthaler**
University of Applied Sciences Esslingen
Department of Information Technology

Supervisor: **Alex Brenner**
German Aerospace Center
Solar High Temperature Technologies – System Modelling

University of Applied Sciences Esslingen
Kanalstraße 33
73728 Esslingen am Neckar

German Aerospace Center (DLR)
Institute of Solar Research
Linder Höhe
51147 Köln (Cologne)
Germany

Abstract

Defects and faults in parabolic trough power plants often lead to lower energy production. The automated detection of such anomalies could reduce downtimes and increase efficiency.

The anomaly detection is based on sensor data which are measured anyway for the operation of the plant. The measured data are high-dimensional data in a spatio-temporal context.

In the first step, the efficient preprocessing and visualization of the solar field and weather data is realized. From the existing sensor data useful features are extracted, which build the input for further approaches of anomaly detection models.

Regarding the given problem, approaches that make use of methods of artificial intelligence are presented. One approach is pursued. The basic idea of this approach is the consideration of multivariate time series on loop level, which are segmented and clustered. For the segmentation of the time series three different segmentation methods based on the bottom-up or sliding-window principle are compared. For clustering, the density-based DBSCAN algorithm is used.

The results of the anomaly detection show that about 93 % of the detected time series segments behave unusually from the data point of view due to manual interventions in the operation of the power plant. Suggestions for an improvement are given to reduce the number of false detections. Furthermore, weaknesses of the implemented approach are pointed out.

Kurzfassung

Defekte und Fehler von Parabolrinnen-Kraftwerken führen häufig zu einer geringeren Energiegewinnung. Durch das automatisierte Erkennen solcher Anomalien könnten Ausfallzeiten verringert und die Effizienz gesteigert werden.

Die Anomalie-Erkennung basiert auf Sensordaten, die für den Betrieb der Anlage ohnehin gemessen werden. Bei den Messdaten handelt es sich um hochdimensionale Daten im räumlich-zeitlichen Kontext.

Im ersten Schritt werden die effiziente Aufbereitung sowie die Visualisierung von Solarfeld- und Wetterdaten realisiert. Aus den vorhandenen Sensordaten werden nützliche Features extrahiert, die als Input für weitere Ansätze zur Anomalie-Erkennung dienen.

Bezüglich der Problemstellung werden Ansätze vorgestellt, die auf Methoden der künstlichen Intelligenz zurückgreifen. Dabei wird ein Ansatz weiterverfolgt. Die Grundidee dieses Ansatzes ist die Betrachtung von multivariaten Zeitreihen auf Loop-Ebene, die segmentiert und geclustert werden. Für die Segmentierung der Zeitreihen werden drei verschiedene Segmentierungsverfahren basierend auf dem Bottom-Up oder Sliding-Window Prinzip verglichen. Für das Clustering wird der dichtebasierte DBSCAN-Algorithmus verwendet.

Die Ergebnisse der Anomalie-Erkennung zeigen zu ca. 93 % eine Detektion von Zeitreihen-Segmenten, die sich aufgrund von manuellen Eingriffen in den Anlagenbetrieb aus Sicht der Daten ungewöhnlich verhalten. Es werden Verbesserungsvorschläge gegeben, um häufige Fehldetektionen zu verringern. Zudem werden erkannte Schwächen des implementierten Ansatzes aufgezeigt.

Contents

Index of abbreviations

AI *Artificial Intelligence*

CNN *Convolutional Neural Network*

CSP *Concentrated Solar Power*

DBSCAN *Density-Based Spatial Clustering of Applications with Noise*

DLR *Deutsches Zentrum für Luft- und Raumfahrt*

DNI *Direct Normal Irradiance*

FMEA *Failure Mode Effects Analysis*

FN *False Negative*

FP *False Positives*

GUI *Graphical User Interface*

HTF *Heat Transfer Fluid*

IAM *Incident Angle Modifier*

LCOE *Levelized Cost of Electricity*

LOF *Local Outlier Factor*

LSTM *Long-Short Term Memory*

MDB *Microsoft Access Database*

PAA *Piecewise Aggregate Approximation*

PC *Principal Component*

PCA *Principal Component Analysis*

PTPP *Parabolic Trough Power Plant*

RGB *Red-Green-Blue*

RNN *Recurrent Neural Network*

SCA *Solar Collector Assembly*

TP *True Positive*

Index of figures

Index of tables

Symbols

Solar field symbols

Symbols	Description	Unit
θ_i	Incident angle	°
φ_{track}	Track angle of collector	°
φ_{sca}	SCA angle	°
$\varphi_{sca\ dev,\ loop}$	SCA angle deviation of loop	°
$\varphi_{sca\ dev,\ col}$	SCA angle deviation of collector	°
$\dot{V}_{sub}$	Volume flow of HTF at inlet of subfield	$m^3 \cdot s^{-1}$
$\dot{V}_{loop}$	Volume flow of HTF at inlet of loop	$m^3 \cdot s^{-1}$
$\dot{m}_{sub}$	Mass flow of HTF at inlet of subfield	$kg \cdot s^{-1}$
$T_{in,\ sub}$	Inlet temperature of subfield	$°C$
ρ_{HTF}	Density of the HTF	$kg \cdot m^{-3}$
f_{focA}	Focusing factor	-
η_{cos}	Cosine loss or efficiency	-
$\eta_{endloss}$	End loss or efficiency	-
K	IAM	-
G_{bn}	Direct Normal Irradiance	$W \cdot m^{-2}$
G_{col}	DNI of collector	$W \cdot m^{-2}$

General Symbols

Symbols	Description	Unit
D	Total number of dimensions	-
N	Total number of time points	-
n	Number of time points	-

b_N	Number of breakpoints	-
Δt_{seg}	Duration of time series segment	s

Evaluation of time series segmentation symbols

Symbols	Description	Unit
$s_{paa,org}$	Score for placement of breakpoints of segmentation due to original signal	-
$s_{paa,std}$	Score for placement of breakpoints of segmentation due to moving standard deviation of original signal	-
s_{n_segs}	Score for number of segments	-
s_{seg}	Total segmentation score	-

Evaluation of clustering symbols

Symbols	Description	Unit
p_{core}	Core point ratio	-
$p_{outlier}$	Outlier point ratio	-
k	Number of nearest neighbors	-
$n_{clusters}$	Number of clusters	-

Hyperparameters symbols

Symbols	Description	Unit
r_{hood}	DBSCAN: Radius of neighborhood	-
n_{min}	DBSCAN: Minimal number of points in neighborhood	-
ε	Window-sliding: Threshold for reconstruction error	-
τ_{drop}	PCA-based: Dropping tolerance	-
M	PCA-based: Fusion tolerance	-
n_{seg}	Periodic: Number of time points of segment	-

1. Introduction

Parabolic trough is a common technology of concentrated solar power (CSP). CSP plants use mirrors to concentrate a large area of sunlight and convert it to heat. The used solar energy belongs to the renewable energies. Comparing to other technologies of renewable energies CSP plants can store energy during the night or at cloudy days more easily [1].

There are several technologies for CSP like parabolic trough, solar power tower, Fresnel reflectors and Dish Stirling. Parabolic trough is the most common and mature CSP technology [2]. In 2017, it accounts worldwide for about 90 % of CSP [3].

This chapter clarifies the motivation of anomaly detection for parabolic trough systems and its realization with techniques of artificial intelligence (AI) as well as the objective of this thesis.

1.1 Motivation

The motive for detecting anomalies is to reduce the levelized cost of electricity (LCOE) for parabolic trough power plants (PTPP). How anomaly detection can possibly reduce the LCOE as well as the reason why techniques of AI are chosen to detect anomalies is elucidated in the following subsections.

1.1.1. Anomaly detection for PTPPs

Detecting anomalies helps to find failures and defects of the system, which could lead to less maintenance costs and a potential higher power output. This coherence will be explained more accurate in the following:

Through the early detection of failures, these can be corrected immediately. That introduces two advantages. First of all, the downtime of the system is shorter. The proper functionality of all components results in a general higher efficiency, availability and reliability of the power plant. Secondly, it prevents from further damage. A failure that is not repaired for a long time can cause other failures. Less consequential damages lead to reduced maintenance work and a longer operational lifetime.

The reduced maintenance costs and a potential higher power output decreases the LCOE for PTPPs. This could result in lower electricity costs and therefore in a higher acceptance by consumers for solar energy and renewable energies in general.

1.1.2. Anomaly detection with artificial intelligence

Most PTPPs have already installed a lot of measuring technology for operational reasons. Mostly the measurements are recorded but never used for other purposes. Here we use the existing recordings for anomaly detection, which is a common use case of artificial intelligence. With its techniques it is possible to analyze a huge amount of measured data automatically.

Our approach primarily needs software to be implemented. Besides that, it only requires hardware with suitable computational power. In comparison to other approaches, which use new sensors or devices, acquisition costs are possibly lower.

1.2 Objective

In this use case anomaly detection is equal to failure detection of PTPPs. The anomaly detection takes place on multidimensional time series with spatial distribution. The aim is to narrow the time point and location of occurred anomalies. The anomaly detection should make a rough proposition about where and when the solar field reacts anomalous.

To reach the overall objective of detecting anomalies, the following steps are done:

- Preprocess measurement data (e.g. interpolation, normalization, etc.).
- Find suitable approaches in the field of AI, that include deep learning and machine learning methods.
- Implement and validate promising approach with different hyperparameters.
- Visualize measurement data, intermediate results of methods and results of the anomaly detection.

There are various configurations of PTPPs. In this research the anomaly detection is implemented and validated for a determined PTPP to serve as a reference. Furthermore, the objective has several limitations:

The observation area is reduced to the solar field of the PTPP which is its crucial difference to conventional power plants. Moreover, the focus is on detecting the most relevant failures that occur frequently or result in extensive damages. Finally, the anomaly detection only needs to work offline. There is no real-time capability required.

2. Parabolic trough power plants

PTPPs belong to the type of line focused CSP plants. With the help of mirrors, they concentrate sunlight on one axis, where an absorber pipe absorbs the solar energy. Hence, the absorber pipe heats up and conduct the heat to a heat transfer fluid (HTF).

The following sections describe the layout of the power plant, the individual components, important parameters, the built-in measuring technologies as well as operational strategies. To understand the evolved and implemented approaches of this research this chapter gives necessary knowledge about PTPPs

2.1. Layout and components

Most of the PTPPs have a similar installation layout. It is divided into a solar field and a power block with an energy storage.

In the following the typical schematic layout of the solar field will be shown. Furthermore, the principal energy generation in the power block and the energy storage system will be described shortly. Moreover, components of the solar field as there are collector, receiver and heat transfer fluid (HTF) will be explained.

2.1.1. Solar field

Figure 2.1 shows exemplarily the layout of a PTPP. The solar field is divided in subfields which is typically a quarter of the whole solar field. The HTF is pumped from the power block to the subfields via cold runner pipes and comes back via hot runner pipes.

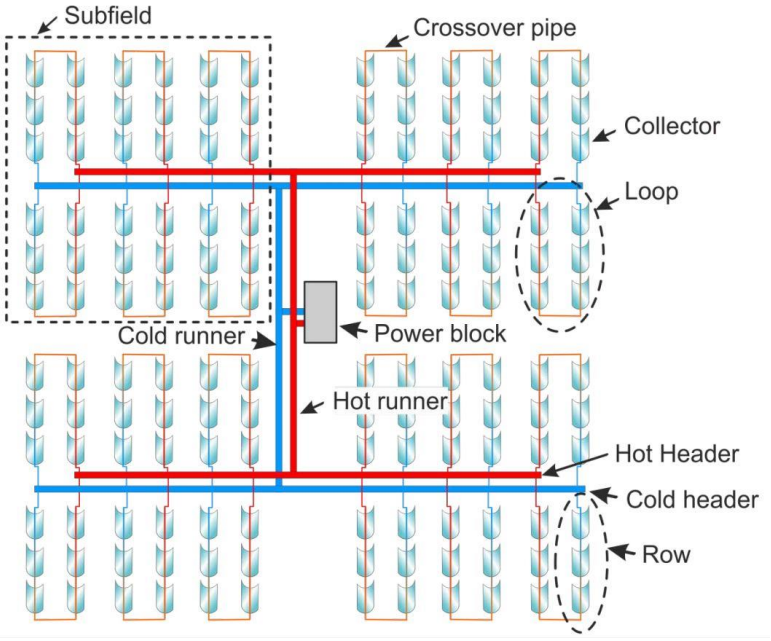

Figure 2.1: Schematic of a typical H-layout of a PTPP. [4]

Each subfield contains so called loops that are connected in parallel by a cold header and hot header pipe. Each loop consists of a certain number of collectors connected in series. Often the collectors of one loop are arranged in two rows that are connected with a crossover pipe. [4]

2.1.2. Power block

The power block processes the heated HTF and contains a steam turbine, an electrical generator and an energy storage system.

The HTF comes from the solar field via the hot headers and leads into the power block. Here the thermal energy is either converted to electrical energy with a conventional steam turbine or stored in the storage system. Usually storage systems of CSP plants store the energy in form of thermal energy. Thereby the thermal energy is often stored in molten salt. In both processes, energy conversion and storage, the HTF cools down and is pumped back to the solar field via the cold headers.

2.1.3. Collector

The collector is the assembly of the parabolic mirror facets, the receiver and the construction that connects both of them. The construction must have a high stiffness so that the mirror does not deform to assure the optical accuracy. The huge collector is very heavy and its mirror facets offers a large target area for wind. The construction has to withstand the weight force and external influences such as wind. Multiple collectors are connected to a solar collector assembly (SCA) that is around 150 m long. Often an entire SCA is referred to as collector. The same wording will be adopted by this research.

Figure 2.2: Multiple parabolic trough collectors joined as SCA. [5]

To focus the sunlight the collector is aligned with the sun. In most PTPPs the collectors are oriented along the north-south axis. During the day, the collector rotates around its longitudinal axis from east to west. The collector's longitudinal axis is also called collector axis. The alignment with the sun position is determined with an actuating motor that is mostly placed at the center of the SCA. [6]

There are ball joints at both ends of the SCA, which connect the moving receiver with the static header or crossover pipe.

2.1.4. Receiver

The receiver (sometimes also called heat collection element) is a part of the collector and is placed in the focus line of the parabolic mirror. It consists of the absorber pipe and its thermal insulation. The HTF, which absorbs the heat, is pumped through the absorber pipe. The composition of a receiver is shown in Figure 2.3.

The absorber pipe is a metal pipe with a diameter of around 7 cm. A selective coating minimizes the radiation losses.

To reduce the thermal losses of the heated HTF it needs an insulation that also lets the sunrays through. Hence, a cladding tube of glass is around the absorber pipe. In the annulus, which is the space between cladding tube and absorber pipe, is a vacuum that reduces the convectional heat loss to a minimum.

Over time the quality of the vacuum decreases due to molecules that penetrate through the absorber pipe or cladding tube. Therefore, a getter is installed, which binds molecules like H_2. As long as the getter is unsaturated it keeps the quality of the vacuum at a certain level.

The receiver is exposed to a high temperature range, which is why it expands strongly. The expansion is compensated with folding bellows at both ends of the receiver. [6]

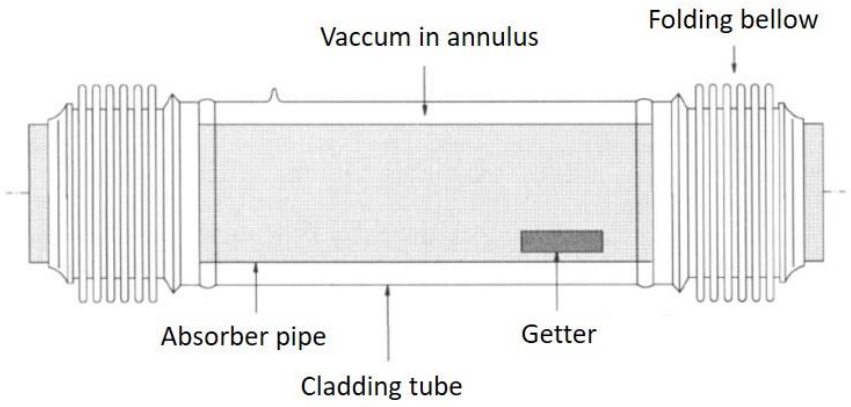

Figure 2.3: Composition of receiver. [6]

2.1.5. Heat transfer fluid (HTF)

The heat transfer fluid is the fluid that flows through the PTPP. It absorbs the heat in the receiver pipes. The most common fluid types are thermal oils, molten salt and water.

The reference PTPP uses a thermal oil. The disadvantage of its lower heat capacity compared to water is compensated by the advantage of the higher boiling point. The oil can be used as a liquid medium in a temperature range from 12°C to 400°C. The higher the temperature of the HTF the better is the thermal efficiency for the power block. Below 12°C the oil solidifies, which makes the power plant operation impossible. Above 400°C the oil degrades very fast and therefore emits H_2 increasingly. These operational conditions should be avoided. [6]

2.2. Geometrical key figures

The position and size of the mirrors in reference to the sun is important to absorb the maximum of solar irradiance. Therefore, important geometrical key figures are explained in this section.

2.2.1. Incident angle

An incident angle is the angle between a beam radiation on a surface and the normal to that surface. The angle depends on the sun position defined by the solar altitude and azimuth angle and therefore on the time of the day and year, as well as the latitude of the surface location. [7]

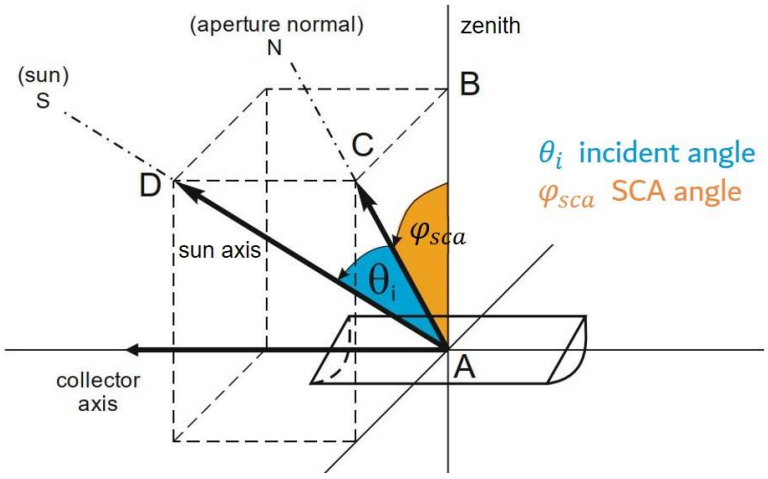

Figure 2.4: Incident angle and track angle for a parabolic trough collector in north-south orientation. Adapted from [4].

Regarding PTPPs the incident angle θ_i is the angle between the DNI and the collector's aperture normal (see Figure 2.4). The minimum incident angle is reached if the collector is fully focused. The incident angle is 0° if the sun is right above the aperture area. [7]

2.2.2. SCA angle

The SCA angle φ_{sca} is the angle between zenith and the aperture normal of the collector (see Figure 2.4). The SCA angle is 0° if the collector normal is pointing to the zenith. For a north-south orientation, it is -90° if the collector normal is pointing to the east and 90° to the west.

2.2.3. Track angle

The track angle φ_{track} is the angle between zenith and the plain which is spanned by the collector axis und sun axis (see Figure 2.4). The track angle is -90° when the sun rises and 90° when the sun goes down. A collector is fully focused when SCA angle and track angle are equal.

2.2.4. Focusing factor

The focusing factor f_{focA} describes how the collector is tracked to the sun and therefore which proportion of the sunlight or DNI is focused on the receiver. It has a value between zero and one. Zero means that the collector is not focused, whereas one means that it is fully focused. The focusing factor can be approximated by a polynomial function

$$f_{focA} = a_n \cdot \Delta\varphi^n + \cdots + a_1 \cdot \Delta\varphi^1 + a_0 \tag{2.1}$$

of order n where $\Delta\varphi$ is the difference between SCA angle and track angle

$$\Delta\varphi = \varphi_{SCA} - \varphi_{track} \tag{2.2}$$

The focusing factor is very sensitive with respect to Δφ. Figure 2.5 shows an example of the focusing factor for a certain collector. In this example the focusing factor changes significantly, if Δφ is greater than 0.5° or lower than -0.5°.

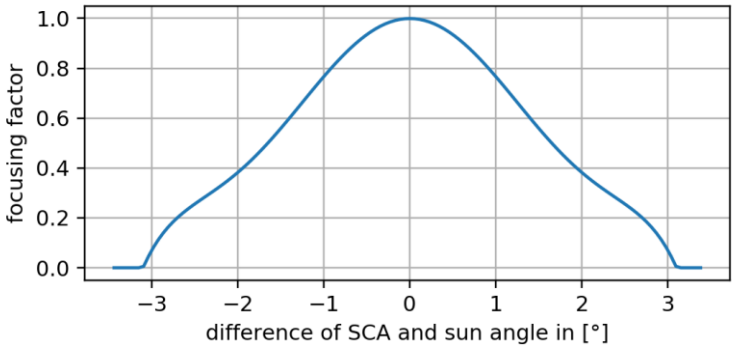

Figure 2.5: Exemplary course of focusing factor-

2.2.5. Aperture area

The aperture area of one collector is defined by the length and the aperture width of the mirror (see Figure 2.6).

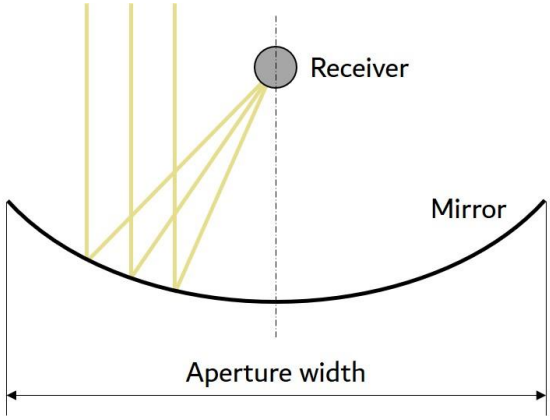

Figure 2.6: Cross section of mirror showing the aperture width

2.3. Geometrical losses

Several losses occur in the solar field process from absorbing the solar energy of the direct sunlight to the outcoming thermal energy of the solar field. In [4] the losses of the solar field are categorized into geometrical, optical, thermal and field losses (see Figure 2.7).

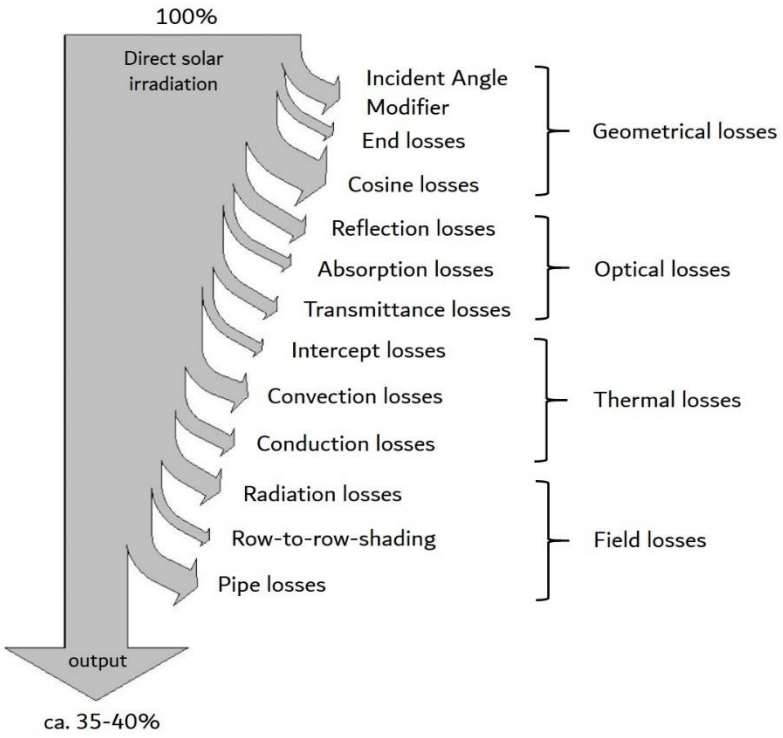

Figure 2.7: Chain of solar field losses that are considered from the solar energy resource to the outcoming thermal energy of the solar field. [4]

2.3.1. Cosine losses

The cosine losses $\eta_{\cos}$ are caused by the reduced aperture area. Since the collectors are tracked uniaxially, most of the time the collector aperture area is not perpendicular to the incident sunrays. Because of that, the apparent aperture area as seen from the sun is reduced (see Figure 2.8).

The factor that reduces the actual aperture area depends on the cosine of the incident angle. This is why the losses are called cosine losses. [4]

$$\eta_{cos} = cos(\theta_i) \tag{2.3}$$

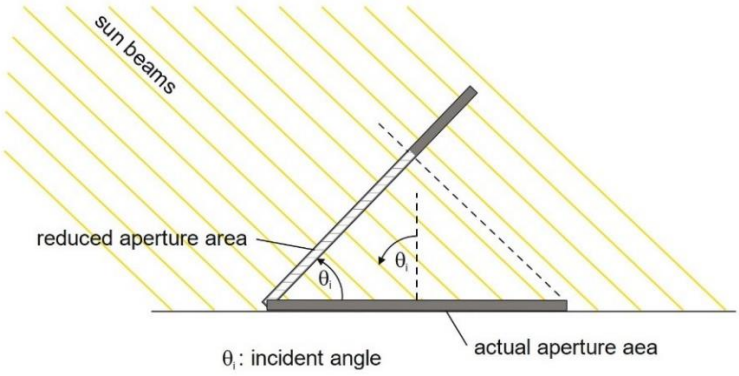

Figure 2.8: Reduced aperture area caused by non-perpendicular sunrays onto the aperture. [4]

2.3.2. Incident angle modifier

Beside the cosine losses there are additional losses depending on the incident angle that are summarized as incident angle modifier K (IAM). These losses are affected by the following physical effects:

- Sunrays reflects at the receiver glass.
- Mirrors are shaded due to collector structures.
- Focusing is not properly due to deformation of the collector depending on SCA angle.
- Sunrays are not hitting the receiver. Since reflected rays are shaped as a cone the projected cone diameter gets bigger with increasing distance for the reflected sunrays between mirror and receiver (see Figure 2.10). This distance depends on the incident angle.
- The same physical effect causes further optical losses with angular dependencies due to imperfect surface and shape of the mirrors.

Between the suppliers of collectors there is no uniform function for the calculation of the IAM. It is provided as approximated polynomial function or lookup table from measurements of the IAM. [4]

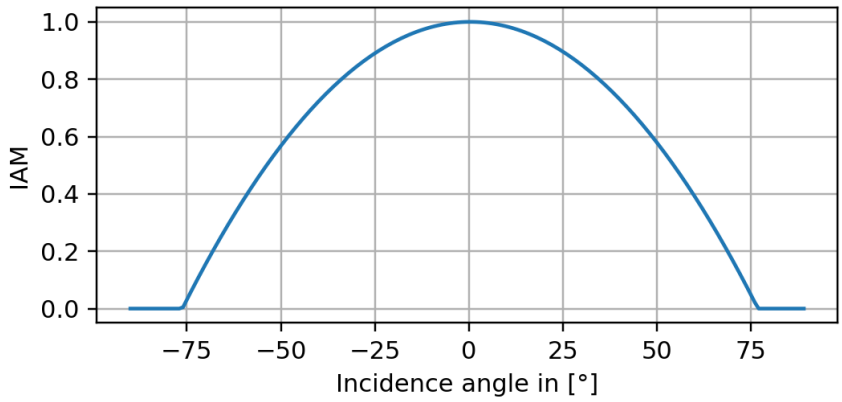

Figure 2.9: Typical exemplary course of IAM

2.3.3. End losses

End losses $\eta_{endloss}$ appear at the rear section of the collector, when sunrays are not perpendicular to the aperture normal. In this context the sunrays of the rear part are reflected but do not hit the receiver. If the SCA angle of a nearby collector is the same, the sunrays hit the receiver of that nearby collector (see Figure 2.10). This reduces the end losses and is also called end gain. The nearby collector could be from the same or from another loop. [4]

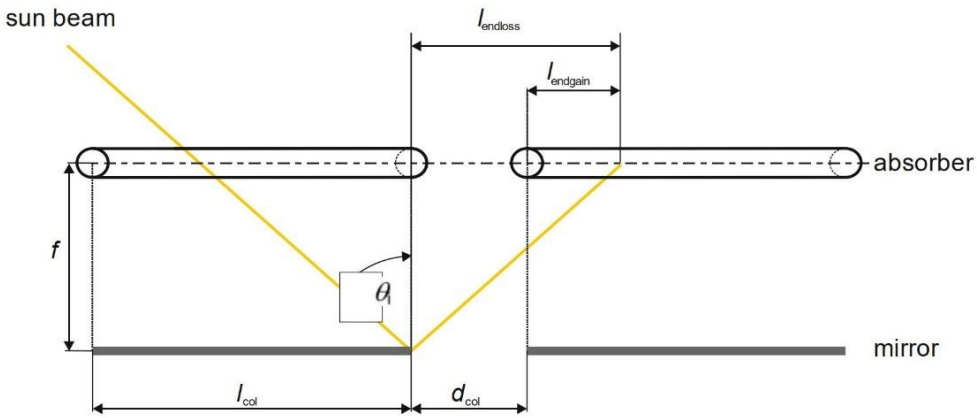

Figure 2.10: End losses and end gains for parabolic trough collectors. [4]

2.4. Measuring technologies of solar field quantities

For the operation of the reference PTPP various physical quantities, such as temperature and volume flow of the HTF, are measured. Furthermore, for the alignment of the collector the SCA angle is measured. The following table shows the measurands with its measuring sites in the solar field of the reference PTPP.

Table 2.1: Overview of solar field measurands of reference PTPP.

Measurand	**Measuring site**
Temperature of HTF	• Inlet and outlet of subfields • Outlets of loops • Middle of collectors
Volume flow of HTF	• Inlet of subfields • Inlet of solar field
SCA angle of collector	• Drive pylon in the middle of the collector

Furthermore, the reference power plant has several weather stations, which are placed at each corner and one in the center of the solar field. They measure environmental conditions like global, direct and diffuse solar irradiance, temperature, wind speed, wind direction, ambient pressure and atmospheric humidity. The direct solar irradiance is also called direct normal irradiance (DNI).

The measurands at a certain measuring site are referred to as features in this research.

2.5. Operational strategies

Operational strategies are aiming to maximize the absorption of energy without overloading components of the power plant. This is achieved by the control of loop outlet temperature and focusing factor as well as a strategy of defocusing that is explained in the following sections.

2.5.1. Control of focusing factor

To maximize the absorption of energy a full focusing is required. Therefore, the focusing factor of the collectors is controlled. The manipulated variable is the SCA angle. The focusing factor depends on the deviation of SCA angle from the calculated or measured track angle.

The power block can only receive the amount of heat from the solar field that the steam turbine and the energy storage system allow. If the power block is overloaded collectors are defocused.

2.5.2. Control of loop outlet temperature

To maximize the thermal efficiency of the power block, it is aimed to keep the temperature of the HTF as high as possible. However, the temperature should be below its critical temperature to prevent damage of the HTF. Therefore, the loop outlet temperature is controlled and its set point is slightly smaller than the critical temperature. It is controlled by the mass or volume flow of the subfield (control variable). With a higher mass flow, the HTF reaches a lower temperature but the absorbed heat is the same.

3. Techniques of artificial intelligence

The anomaly detection is done with the help of artificial intelligence. In this section several techniques, architectures and algorithms are explained to understand the evolved and the implemented approach of this thesis.

3.1. Introduction to anomaly detection

Anomaly detection is a part of the field of outlier analysis. The outlier detection problem has an unsupervised nature because it is much effort to define a ground truth due to the variation of outliers. That is why outlier detection models often use unsupervised or semi-supervised learning methods. [8]

3.1.1. Types of outliers

In [8] outlier detection differs between weak outliers referred as noise and strong outliers referred as anomalies.

Furthermore, outliers are classified into three types: point, contextual and collective outliers (see Figure 3.1). A data instance is termed as point outlier if it is anomalous with respect to the rest of the data. A data instance is termed as contextual outliers if it is anomalous in a specific context but not otherwise. A collection of related data instances is termed as collective outlier if the shape of the collection is anomalous [9].

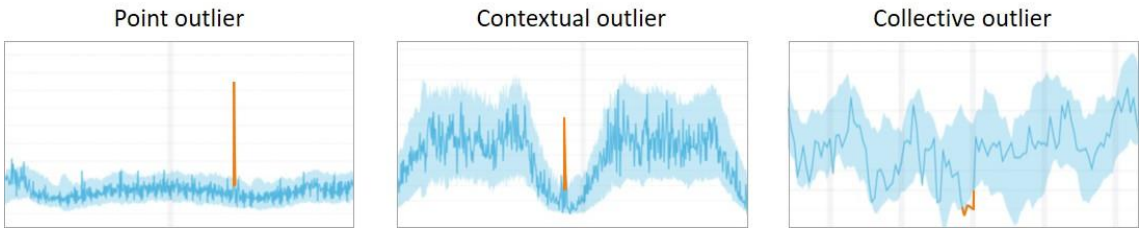

Figure 3.1: Examples for a point, a contextual and a collective outlier. [10]

3.1.2. Basic outlier detection models

There are several approaches to detect outliers. In [8] outlier detection models are classified into five groups:

- Extreme value analysis
- Linear models
- Probabilistic and statistical models
- Proximity-based models
- Information-theoretic models

The common technique of all outlier detection models is to create a model of the normal pattern in the data. Outliers or anomalies are reported as points that do not conform to the normal profile. For high-dimensional data it is easier to find outliers in a projected subspace instead of in the original feature space. The output of an outlier detection model can be an outlier score or a binary label. [8]

In this research a proximity-based model is used. The idea of proximity-based methods is to isolate outliers from the rest of the data on the basis of similarity or distance functions. Principal approaches of this kind are cluster analysis, density-based analysis and nearest neighborhood [8].

3.2. Neural network architectures

This section introduces the basic functionality of convolutional neural networks, recurrent neural networks and autoencoders.

3.2.1. Convolutional neural networks

Convolutional neural networks (CNNs) are widely used in computer vision for image recognition and object detection.

In the architecture of CNN, each network layer is 3-dimensional. It has a spatial extent and a depth with a certain number of features. The spatial extent of one feature is called feature map and can represent hidden features of an image. When the input is an image than the spatial extent corresponds to the height and width of the image and the depth (features) to the color channels like RGB. The input data not necessarily needs to be an image, it can be any 3-dimensional matrix. In CNNs two types of layers are present, which are convolution and subsampling layers. [11]

For convolution layers (see Figure 3.2), a convolution operation is defined. It is defined by a certain number of filters that corresponds to the number of features in the current layer. Each filter has weights and is 3-dimensional. All filters of one layer have the same dimensions. Their depth is equal to the previous layer, their spatial extent (also called kernel size) can be determined freely. In forward propagation each filter runs over every spatial position of the previous layer and builds the dot product of the filter and the matrix of current position. The results are the values of the corresponding outcoming feature map after applying an activation function like ReLU. [11]

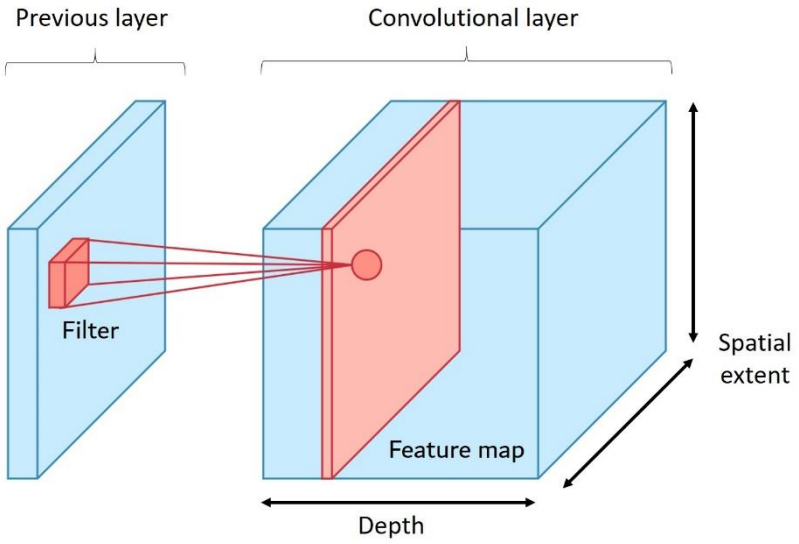

Figure 3.2: Schematic representation of convolution layer. [12]

Subsampling layers reduce the spatial extend to prevent overfitting. They also reduce the number of computing operations.

Depending on the application fully-connected layers are needed to map a 3-dimensional layer of the CNN to a 1-dimensional layer. In image classification applications at least one fully-connected layer is at the end to represent the classes. [11]

Figure 3.3 shows an exemplary CNN architecture for image classification. Like most CNN architectures it has various convolutional, subsampling and fully-connected layers.

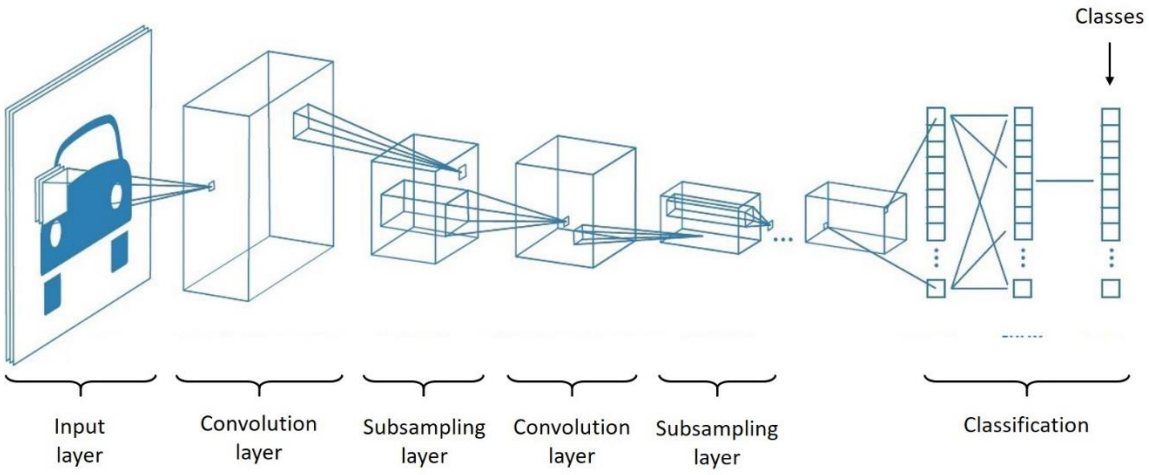

Figure 3.3: Exemplary architecture of a CNN. [13]

3.2.2. Recurrent neural networks

Recurrent neural networks (RNNs) are designed to process sequential data like text sentences, time series or videos. The sequences x_t are successively given to the RNN. One sequence is for example a word of a sentence, a segment of a time series or a frame of a video. [11]

With a function f the input sequence is converted into a hidden state h_t. Thereby the function also considers the previous hidden state h_{t-1}. The current hidden state can therefore be represented by:

$$h_t = f(h_{t-1}, x_t) \tag{3.1}$$

Because the function depends on the previous hidden state the order of the sequences is significant. From the current hidden state, an output o_t can be calculated with a second function g. It learns to predict the output for the current hidden state:

$$o_t = g(h_t) \tag{3.2}$$

It is not necessary to have an output after each timestamp. This depends on the use case. [11]

The following figure shows the folded and unfolded representation of an RNN. The unfolded representation shows the sequence of the timestamps.

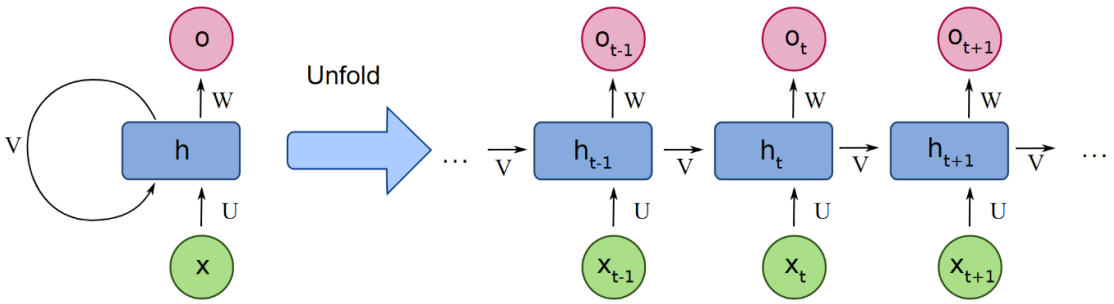

Figure 3.4: Representation of RNN in folded (left) and unfolded form (right). [14]

The same functions f and g are used for each timestamp of the RNN. They are defined by the neural network in advance. A special version of an RNN is the long-short term memory network (LSTM) (see [11]).

3.2.3. Autoencoder

The basic principle of an autoencoder is to compress the input data and subsequently reconstruct it so that input and output data are the same. The compression is done by an encoder, and the reconstruction by a decoder. [11]

It is common that the autoencoder has a symmetric architecture. The number of units in the middle layer is typically fewer than the input or output layer. It is called the bottleneck and holds a reduced representation of the input data. The decoder tries to reconstruct the input data from the reduced representation. The reconstruction is inherently lossy. With a loss function the output is forced to be as similar as possible to the input. The loss function uses for example the sum-of-squared differences between input and output. [11]

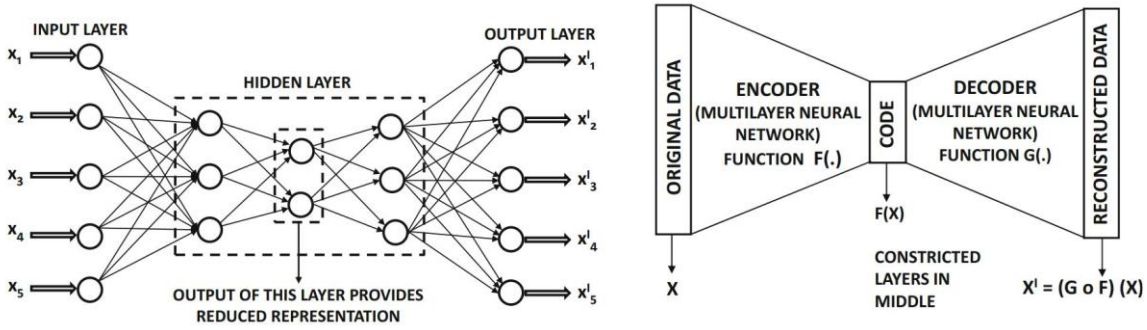

Figure 3.5: Exemplary architecture (left) and general schematic (right) of autoencoder. [11]

Many applications do not even use the output data, but rather the compressed representation of the input data or the loss between input and output. Nevertheless, the output is needed to train the network. [11]

3.3. Clustering

This section introduces the clustering technique and describes the used DBSCAN algorithm in detail.

3.3.1. Introduction

Clustering is an unsupervised machine learning technique. It is the process of grouping similar objects together. Clustering algorithms can be categorized in different types depending on characteristics like the structure of the output clusters, input data, or clustering measures. [15]

For the structure of output clusters there are two different categories of clustering: partitional clustering and hierarchical clustering. In partitional clustering or flat clustering algorithms the objects are partitioned in unique and separated clusters. Hierarchical clustering algorithms create a nested tree of clusters. The tree is often visualized in a dendrogram. [15]

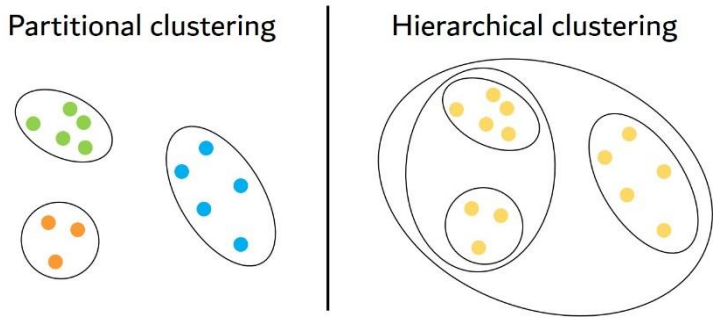

Figure 3.6: Principle visualization of partitional and hierarchical clustering.

For the type of input data there are two other categories of clustering: similarity-based clustering and feature-based clustering. In similarity-based clustering algorithms the input is a dissimilarity or distance matrix. In feature-based clustering algorithms the input is a feature or design matrix. [15]

For different types of clustering measures there are further categories of clustering: e.g. distance-based, density-based, grid-based or spectral clustering algorithms. A very common algorithm is the density-based clustering. It clusters dense areas which are separated from each other by sparse areas in the data space. It is a non-parametric algorithm that makes no assumption about the number of clusters or their distribution and allows arbitrary shapes of clusters. [16]

3.3.2. DBSCAN algorithm

Density-Based Spatial Clustering of Applications with Noise (DBSCAN), as indicated by its name, is a density-based clustering algorithm. Each point has a neighborhood that is defined by a radius r_{hood}. Depending on other points within this neighborhood each point is classified as core, border or outlier point. [16]

- A point is a core point if its neighborhood contains a minimum number of other points n_{min}. (see red points in Figure 3.7)
- A point is a border point if it is not a core point but has another core point in its neighborhood. (see yellow points in Figure 3.7)
- A point is an outlier point if it has no core point in its neighborhood. (see blue point in Figure 3.7)

For the extend of clusters there are three types of connectedness defined: direct density-reachable, density-reachable and density-connected. [16]

- A point q is direct density-reachable from point p if point q is within the neighborhood of point p.
- A point q is density-reachable from point p if there is a path of points where each intermediate point is direct density-reachable from its previous point. This implies that all points on the path are core points including point p.
- Two points p and q are density-connected if there is a third point o from which both points are density-reachable.

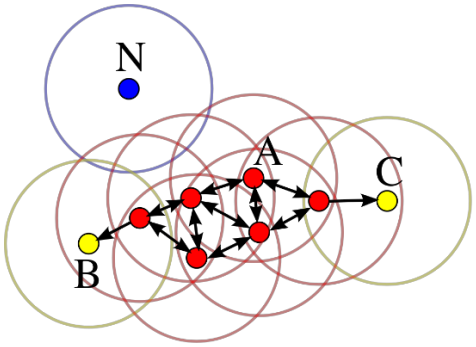

Figure 3.7: DBSCAN: type of points. [17]

A cluster is then a set of density-connected points. Each cluster has at least one core point. Outlier points do not belong to any cluster.

The clustering results depend on the two hyperparameters radius r_{hood} and the minimum point number n_{min}.

3.4. Principle Component Analysis

The Principle Component Analysis (PCA) is a method that is often used to reduce the dimensionality of high-dimensional data. It reduces the number of features, while keeping most of the information from the original data.

The idea of this method is to find new features, denoted as principal components (PCs), which do not correlate with each other. Because when features highly correlate, they contain redundant information. In order to identify the correlations, the covariance matrix is computed. [18]

The PCs are constructed in such a manner that the first PC accounts for the maximal possible variance of the data. For example, if there is a two-dimensional data (see Figure 3.8) the first PC would be the blue line because the projection of the data points onto

that line would be the most spread out. Hence, the larger the variance carried by a line, the more the information it has. [18]

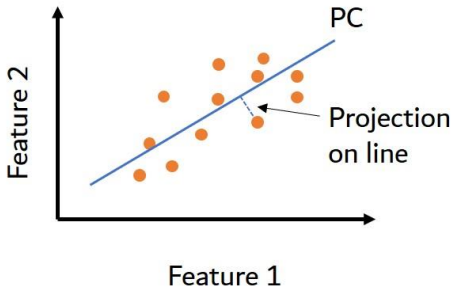

Figure 3.8: Example for principal component

The PCs are found with the help of the eigenvectors and the eigenvalues of the covariance matrix. The eigenvectors point in the directions where there is the highest variance. The respective eigenvalues simply give the amount of variance carried in each PC. Thus, the PCs can be ordered by the carrying amount of information. The first PC always carries the most information of the data. [18]

The dimensionality of data is reduced by calculating its PCs and keeping a smaller number of PCs as the number of features of the original data.

The PCA is sensitive to the variance of the features, that is why the input data needs to be scaled with the z-score standardization. [18]

4. Domain-specific anomalies

In PTPPs anomalies can appear in the solar field, energy storage system or power block. This research only focuses on anomalies or failures that occur in the solar field. These include failures that disrupt the solar field process which starts with the absorption of the solar irradiation and ends with the heated and returned HTF to the power block.

The following introduces the reasons and effects of the most relevant failures of the solar field. After this, a tabular overview of the introduced and further failures is given.

4.1. Failures and their effects

An internal failure mode effects analysis (FMEA) [19] about the most common failures is the basis for the mentioned failures and their effects. The FMEA results in a risk priority number which is the multiplication of the severity of the consequences (severity number), frequency of occurrence (probability number), and ease of detectability (detection number). Each criterion has a number between zero and ten, where ten is the worst case. Therefore, the highest risk priority number is 1000. This section presents some failures with a risk priority number greater than 60.

4.1.1. Soiling of mirror

Over time dirt builds up on the mirrors. The dirt comes from dust which is blown up in the air. It sticks to the mirrors with the wind or falls down with the rain.

The degree of soiling depends on the weather and the location of the mirrors. The soiling is particularly high on hot and dry days as well as on mirrors near the cooling tower or roads. For example at the power plant in Kramer Junction the reflectance decreases on average by 0.45 % daily [20].

A soiled mirror has a lower reflectance. Therefore, the receiver absorbs less solar irradiation. Studies from [20] show that a decrease of reflectance of 1 % results in a 1.2 % lower efficiency.

An effective measure against soiling is cleaning the mirrors which is done regularly. For that the collector is manually rotated to the side so that a machine with an arm of water nozzles can pass by. Because of the big size of the solar field it needs to be cleaned almost continuously (e.g. every three days) to keep a certain degree of reflectance.

4.1.2. Vacuum loss of cladding tube

A vacuum loss is caused by a saturated getter that cannot bind more foreign molecules or a destroyed cladding tube, so that air enters into it. This leads to a higher heat loss.

4.1.3. Damage of heat insulation

A damage of heat insulation or its material is due to penetration of another fluid or destruction of its structure. A structure damage can be caused by installation faults or the frequent expansion of the pipe. A soaking of fluid can be caused by a leakage of the pipe or environmental water like rain. It leads to a higher thermal conductivity and thus to heat loss. The damage can appear suddenly but it can also be a creeping process.

4.1.4. Leakage of pipe

A leakage of pipe can be caused by installation work, material expansion as well as a leaky weld seam, seal or valve. It results in loss of HTF that leads to a heat and pressure loss. Moreover, it has a negative environmental impact. The failure often causes other failures like damage of heat insulation or vacuum loss as well.

4.1.5. Blockage of ball joint

Abrasion, dirt or incorrect installation, e.g. screws tightened too much, can cause a hardly movable or even blocked ball joint. If this happens the collectors end will heavily move with the rotation of the collector's actuator that is placed in the middle. This deforms the collector or mirror and the sun light is not properly focused on the receiver. It leads to a lower absorption of solar irradiation.

4.1.6. Deformation of collector

A deformation of the collector construction can be caused by a blockage of a ball joint, forces due to wind or manufacturing defects. With the deformation the sun light is not properly focused on the receiver which leads to lower absorption of solar irradiation.

4.1.7. Incorrect tracking of collector

Reasons for a bad tracking can be for example a defect of a component of the tracking system like angle sensor, motor, etc. Another reason can be a bad adjustment of the controller. It results in a lower absorption of solar irradiation.

4.2. Overview of failures

The following table shows an overview of the already described and also further failures. The table states possible reasons and negative effects of the failures. Possible reasons like production and installation faults as well as the negative effects like component damages are not mentioned.

Table 4.1: Overview of relevant failures with possible reasons and their negative effects. *Described in detail in section 4.1.

Failure	Possible reasons	Negative effects
Break of mirror	Strong wind	Less solar irradiation
Soiling of mirror *	Deposit of dust	Less solar irradiation
Vacuum loss *	Air penetration, Getter saturation	Heat loss
H_2 degassing of HTF	Critical temperature	Heat loss HTF degradation
Damage of heat insulation *	Fluid penetration, Structure destruction	Heat loss
Leakage of pipe *	Frequent material expansion, Leaky weld seam, seal or valve	Heat loss Pressure loss Fluid loss Environmental impact
Blockage of ball joint *	Attrition, Dirt deposit	Less solar irradiation
Deformation of collector *	Strong wind, Blockage of a ball joint	Less solar irradiation
Misplacement of receiver	Strong wind, Gravity	Less solar irradiation
Incorrect tracking of collector *	Defect component of tracking system, Bad control of focusing factor	Less solar irradiation
Soiling of cladding tube	Deposit of dust	Less solar irradiation
Degradation of anti-reflection coating	Cleaning process, Environmental impacts	Less solar irradiation
Degradation of selective coating	Air	Less solar irradiation

5. Data identification and preparation

In this chapter the identification, organization and preprocessing of the data is described. The data identification addresses the type, instances and resolution of the data. The data organization elucidates the structuring of data and provision of data instances. The data preprocessing includes the interpolation and extraction of features.

The data preprocessing and methods are implemented in python. Used libraries are listed in appendix B.

5.1. Data identification

Every measurement of the PTPP has a spatial and temporal information, hence it is referred to as spatio-temporal data. The physical quantities are recorded at a certain time and location.

5.1.1. Data types and instances

The survey of [21] differs between four types of spatio-temporal data: spatio-temporal events, trajectories, spatio-temporal point reference, and spatio-temporal raster. The measuring data of the PTPP is categorized into spatio-temporal raster data. In raster data, measurements of a continuous or discrete spatio-temporal field are recorded at fixed locations in space and at fixed points in time. Hereby, it does not matter if the distribution of the locations and the time points are regular or irregular.

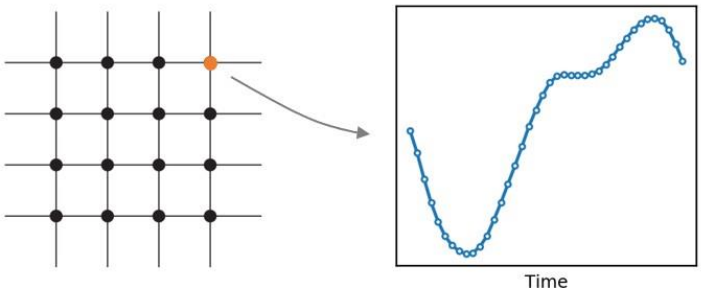

Figure 5.1: Spatio-temporal raster data with fixed locations and time points.

From raster data three types of data instances can be built: *time series, spatial map*, and *spatio-temporal tensors*. Each type treats objects and features differently. In this research the two data instance types *time series* and *spatial map* are considered. For a *time series* instance, the fixed locations are treated as objects and features are defined by measurements over time or as time series (e.g. see Figure 5.2 on the left). For a *spatial map* instance, fixed time points or periods are treated as objects and features are defined

by spatial maps that carry the measurements collected from all spatial locations (e.g. see Figure 5.2 on the right). If a spatial map does not represent a time point, but a period, the values over time are summarized to a single value for example by building the mean. A spatial map should not be considered as image.

Both data instance types can be multi-dimensional. Therefore, a spatial map is a 3-dimensional matrix, with spatial distribution of the values and different features in the depth. A time series is a 2-dimensional matrix, with the timestamps along the first axis and the different features along the second axis.

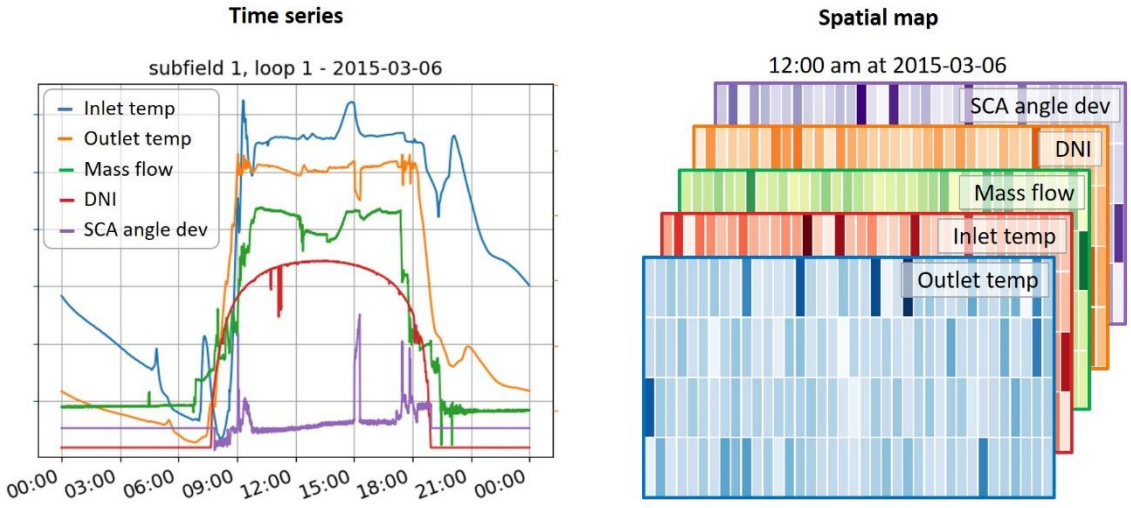

Figure 5.2: Data instances: time series and spatial map.

5.1.2. Spatial and temporal resolution

The data of time series as well as the data of spatial maps has a spatial and temporal resolution. Often the resolution is reduced to decrease the amount of data or to filter out effects of high frequency.

The temporal space of the data is discrete, because measurements are recorded at a certain time point. With interpolation the discrete temporal space becomes semi-continuous. This means the frequency can be chosen freely but with respect to loss of information.

The spatial space for the data is discreetly defined by the observed sections of the PTPP. It is differed between three different spatial resolutions: subfield, loop and collector.

5.2. Data organization

In the following sections, the structuring of the data as well as the provision of data instances are described.

5.2.1. Structuring of data

The data recordings are restructured so that it can be accessed more easily for further usage. Python classes are implemented for each type of section or measuring location of the solar field. The following table shows which features the different classes are saving.

Table 5.1: Accessible features of the implemented python classes.

Class	Features
Loop	Inlet temperature Outlet temperature
Collector	SCA temperature SCA angle
Subfield	Volume flow Inlet temperature Outlet temperature
PTPP	Incident angle Track angle
Weather station	DNI 1 DNI 2

The data of a solar field is structured as follows: The PTPP object contains all subfield and weather station objects. The subfield objects contain their loop objects and the loop objects contain their collector objects. The following diagram illustrates the structure of the objects. In general, one PTPP object and its subobjects contain the data of maximal one day.

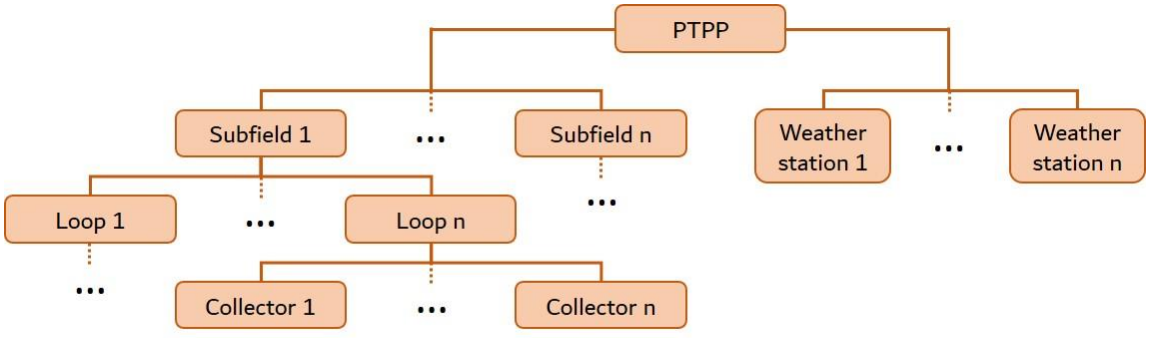

Figure 5.3: Structure of the python objects that save the features.

This structure allows an easy access of the data. For example, with the following line of python code the SCA angle of the loop 8 in subfield 2 at a certain date can be accessed:

```
ptpp.subfield[2].loop[8].get('sca_angle')
```

5.2.2. Provision of data instances

The previous described data structure is used to provide instances of the data. The needed instance type depends on the applied approach of anomaly detection. All instances that are observed for the implemented anomaly detection are pooled in one dataset. All samples in one dataset have the same spatial and temporal resolution.

5.3. Data preprocessing

The measuring data from the reference PTPP are provided as Microsoft Access databases (MDB). Each database saves the data of one day's period. The anomaly detection is done with python. To access the data for processing in an efficient and fast way it is read in with python and saved in the python specific pickle format. Features (e.g. temperatures, DNIs) which are relevant for the later implemented approach, are selected and subjected to a further data preparation. This includes interpolation and feature engineering.

5.3.1. Interpolation

The timestamps of the measured signals are not synchronized and therefore slightly different. For example, the SCA angle of collector 3 in loop 1 of subfield 1 is measured at different timestamps than the SCA angle of the collector 4 of the same loop (see Figure 5.4). To get a better comparability between the different signals, their timestamps are synchronized by linear interpolation (see Figure 5.5).

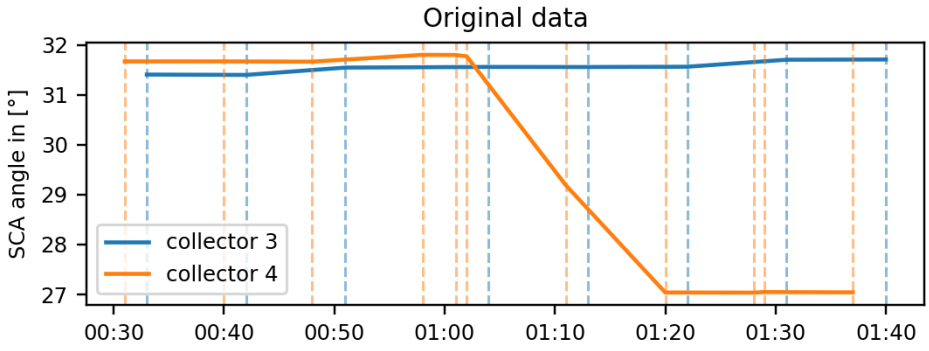

Figure 5.4: Measuring at different timestamps.

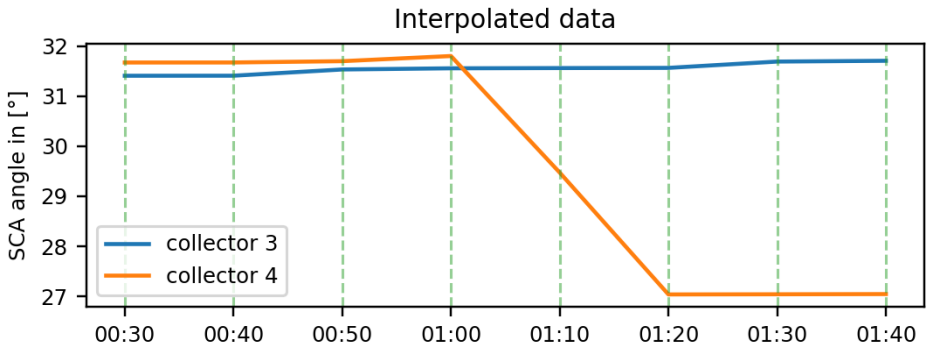

Figure 5.5: Synchronized timestamps (green dotted lines) for interpolated data.

For the interpolation of the features their measuring frequency or period is important. The features at loop level or lower level have an unregular recording period. The recording is on demand, but at least every 10 s. The features at subfield level and higher level as well as in the weather stations are recorded every second, but sometimes a record is missing. The period for interpolation is set to 10 s or 1 s depending on the feature (see Table 5.2).

Table 5.2: Measuring and interpolated period of certain features.

Recording period	**Interpolation period**	**Features**
10 s (or less)	10 s	Loop inlet temperature, Loop outlet temperature, SCA temperature, SCA angle
1s	1 s	Volume flow, DNI

The time series have a maximal duration of one day. The timestamps are within a half-open interval that lasts from 0:00 am to 0:00 am of the next day. Often the recording does not start at 0:00 am. Therefore, a linear extrapolation is implemented that estimates the missing records between 0:00 am and the first record. The extrapolation is only done for short time ranges at midnight.

5.3.2. Feature Engineering

Other features are extracted from the data by using domain knowledge to improve the performance of AI algorithms. The following features are extracted:

- DNI of loop
- SCA angle deviation of loop
- Volume flow of loop
- Mass flow of loop

How this is done is described in the following sections.

Extracting DNI

Each weather station measures two DNI values. At first, it is calculated the mean of these two values. In the next step, it is done a spatial interpolation of the DNI values. Thus, for each loop a DNI value is calculated from the DNI values of the weather stations. For this calculation the DNI values of the three nearest weather stations are taken like it is done in [22].

Each mean DNI value of the weather stations G_i is weighted with w_i. The DNI value at e.g. the collector or loop position G_{pos} is the sum of the weighted DNI values of the weather stations.

$$G_{pos} = \sum_{i=1}^{3} w_i \cdot G_i \tag{5.1}$$

Each weight depends on the distance d_j between the position of the collector or loop (respective center) and the respective weather station (see Figure 5.6).

$$w_j = \frac{1/d_j}{\sum_{i=1}^{3} 1/d_i} = \frac{1}{d_j \cdot \sum_{i=1}^{3} 1/d_i} \tag{5.2}$$

With this calculation the sum of weights is always one.

$$\sum_{i=1}^{n} w_i = 1 \tag{5.3}$$

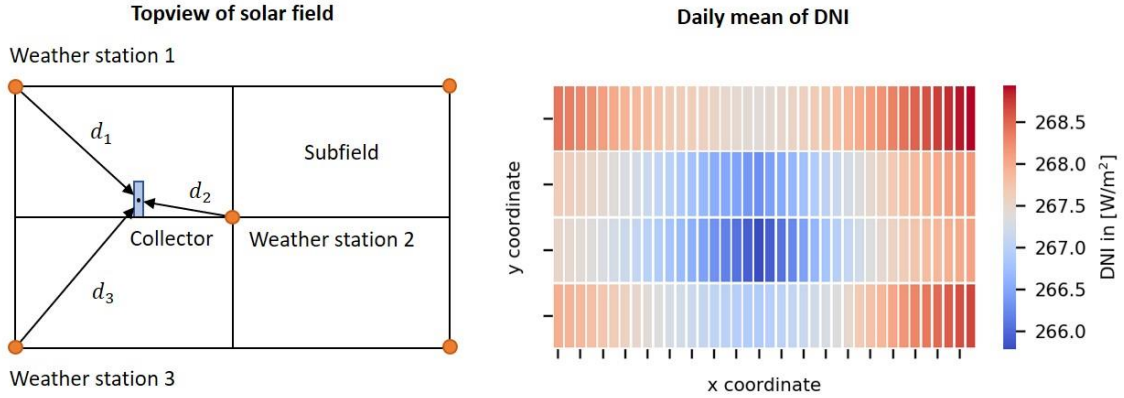

Figure 5.6: Left: Principle of spatial interpolated DNI. Right: Resulting spatial map of loops showing the daily mean of DNI.

Extracting SCA angle deviation

The SCA angle deviation is extracted to reduce the number of features. The SCA angle deviation of a collector $\varphi_{sca\ dev,\ col}$ is the absolute difference between its SCA angle φ_{sca} and the track angle φ_{track}.

$$\varphi_{sca\ dev,\ col} = |\varphi_{sca} - \varphi_{track}| \tag{5.4}$$

The SCA angle deviation of a loop $\varphi_{sca\ dev,\ loop}$ is the mean of the corresponding collectors SCA angle deviations.

$$\varphi_{sca\ dev,\ loop} = \frac{1}{n}\sum_{i=1}^{n} \varphi_{i,sca\ dev,\ col} \tag{5.5}$$

whereby n is the number of collectors in the loop.

It is taken the absolute value for the SCA angle deviation to avoid that the deviations eliminate each other. Here an example: The first collector has a deviation of $-3°$ and the second collector has a deviation of $3°$. Every other collector has a deviation of $0°$. By simply building the mean of all deviations it would result in a mean deviation of $0°$. This is not acceptable because two collectors of the loop are not fully focused. By taking the mean of the absolute deviations it results in a mean deviation of $1.5°$.

Extracting mass flow & volume flow

The volume flow of a loop is not measured, but can be approximated from the volume flow of the subfield. The volume flow of a loop is the volume flow of the solar field $\dot{V}_{sub}$ divided by its number of loops n.

$$\dot{V}_{loop} = \frac{\dot{V}_{sub}}{n} \tag{5.6}$$

This calculation assumes the ideal flow distribution so that every loop of a subfield has the same volume flow. In practice the flow is not evenly distributed due to pressure losses in the pipes as well as inaccuracies in the manufacturing process of the pipes and loop inlet valves.

The mass flow at the inlet of the subfield $\dot{m}_{sub}$ is calculated with the volume flow $\dot{V}_{sub}$ and the density of the HTF ρ_{HTF}. The density depends on the HTF inlet temperature of the subfield $T_{in,\,sub}$, where the Volume flow $\dot{V}_{sub}$ is measured.

$$\dot{m}_{sub} = \dot{V}_{sub} \cdot \rho_{HTF\,(T_{in,\,sub})} \tag{5.7}$$

The temperature dependent density is obtained from an interpolated lookup table.

The mass flow of the loops is calculated in analogy to the calculation of the loop volume flow from the subfield volume flow (see equation (5.6)).

6. Evolved approaches of anomaly detection

This chapter introduces the evolved approaches for the anomaly detection problem of PTPPs. Four promising approaches are evolved:

- Efficiency model
- Clustering of time series
- Autoencoder
- Recurrent-Autoencoder

The data does not provide a ground truth of the occurrence of an anomaly in the data. Therefore, the evolved approaches include models with unsupervised and semi-supervised learning methods.

The approaches should be capable of detecting point, contextual and collective outliers (see 3.1.1). The basic ideas of the approaches are explained in the following sections.

6.1. Approach: Efficiency model

Most of the failures of a PTPP end up in less absorption of solar irradiation or increased heat loss. Both affects the efficiency of the component or section.

The idea of this approach is to create a model for the calculation of the momentary thermal efficiency. Afterwards an anomaly detection method is applied on the time series of the momentary efficiency.

The inputs of the efficiency model are the features, which are relevant for the calculation of the thermal efficiency of the observed section (e.g. subfields, loops or collectors). The outputs of the anomaly detection model are the anomalous time periods of the observed section. The knowledge about the observed section give the spatial information for a detected anomaly.

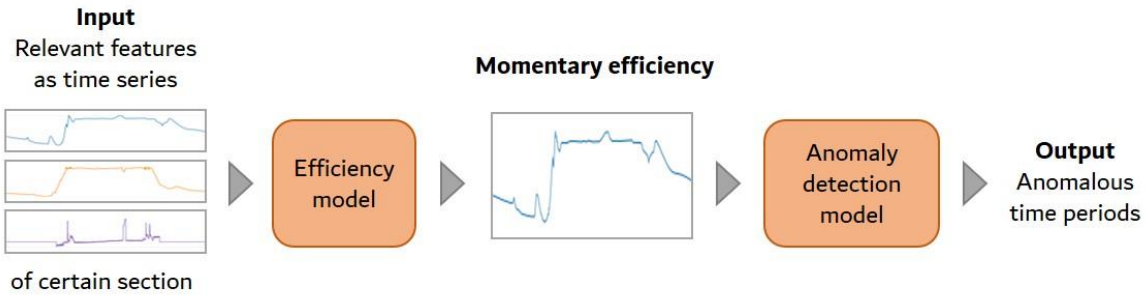

Figure 6.1: Schematic *efficiency model* approach.

The disadvantage of this approach is that it needs knowhow about the physical processes as well as parameters of the observed PTPP.

However, the resulting momentary efficiency is intuitive and easy to understand. The data is reduced to a univariate time series that enables a less complex anomaly detection model. The anomaly detection could be done with a simple extreme value analysis.

An approach for the efficiency model which has been developed by this research is described in the appendix 0.

6.2. Approach: Clustering of time series

The idea of this approach is to divide time series into elementary segments that are further classified by a clustering algorithm. The input of the segmentation is a multivariant time series with certain features of a certain section.

The resulting time series segments are transferred into a feature space, where the clustering takes place. Depending on the clustering, the time series segments are referred as normal or anomalous. Segments or its corresponding points in the feature space can be taken as anomalous if one of the following conditions is true:

- It does not belong to any cluster
- It is in an anomalous cluster
- It is far away from its cluster centroid

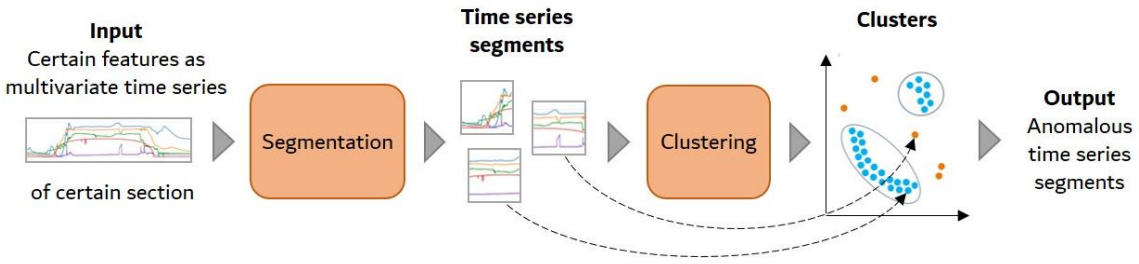

Figure 6.2: Schematic *clustering of time series* approach.

The location of an anomalous time series segment and therefore of an anomaly is known from the linked section (e.g. collector or loop) of the input time series. This is done under the assumption that the data reacts anomalous where the reason of the anomaly take place. The point of time of the anomaly is known from the start and end point of the anomalous time series segment.

6.3. Approach: Autoencoder

The idea of this approach is to run the data through an autoencoder. The autoencoder encodes and decodes the data (see functionality of autoencoder in section 3.2.3). After running the data through the autoencoder the input and output of the autoencoder are compared to calculate a dissimilarity map. From the dissimilarity map anomalous areas are derived.

The autoencoder is trained with all the data: anomalous and normal data. By nature, the normal data is dominant because anomalous data is very rare. Therefore, it is assumed that the reconstruction of normal input data is good and that of anomalous data is bad.

For this approach there are two variants that differ in the type of data input. The input can be a spatial map or a time series as described in 5.1.1.

The approach is further explained on the variant with spatial map as input as shown in Figure 6.3. The input data (spatial map) is encoded by an CNN and decoded by another CNN (see detailed description of CNN in section 3.2.1). The output is the reconstruction of the input data. From the output and input a dissimilarity map is built. The dissimilarity map has the same spatial extent as the input data but is only one-dimensional. The spatial areas that are anomalous in the spatial context result in a high value in the same area of the dissimilarity map. The dissimilarity values are considered as anomaly scores. If an anomaly score of a certain area is above a specified threshold the area is referred to as anomalous. The areas are linked to a certain location in the solar field.

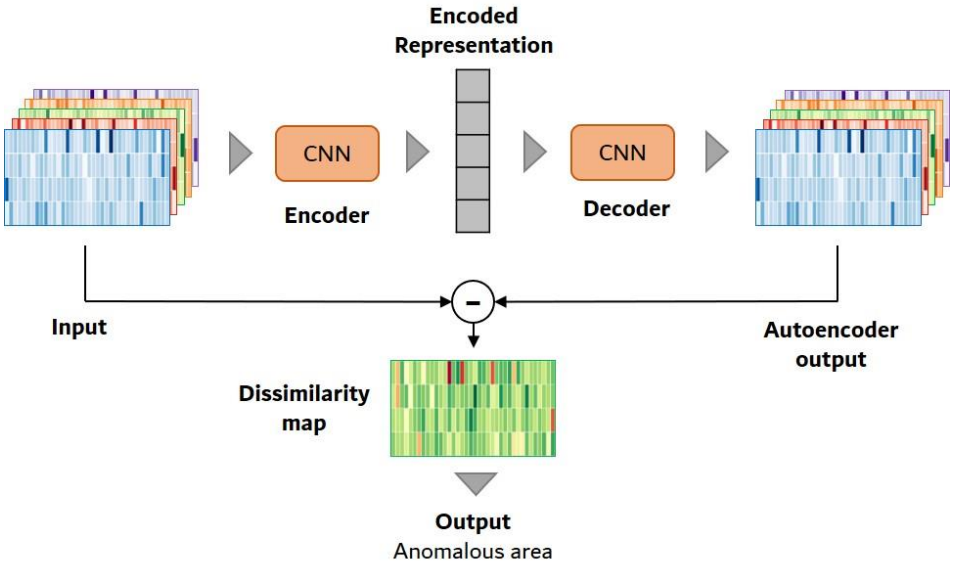

Figure 6.3: Schematic *autoencoder* approach with spatial map as input.

The second variant, with time series as input, works in the same way, but instead of a CNN for the encoding and decoding another artificial neural network is used. In this variant, the dissimilarity map is a univariate time series. The anomalous areas in the dissimilarity time series are linked to the time points of the input data.

Both variants can also work in cooperation. The spatial map variant considers the spatial dependence of the data, whereas the time series variant considers the temporal dependence. The resulting anomalies of both variants are validated with each other and possibly joined.

6.4. Approach: Recurrent-Autoencoder

This approach improves the previous described autoencoder approach (only spatial map variant). The idea is to implement the autoencoder within an RNN (see detailed description of RNN in section 3.2.2). The carried hidden state could improve the encoding and decoding of the autoencoder. That could lead to a better result for the autoencoder output and therefore to more precise dissimilarity maps.

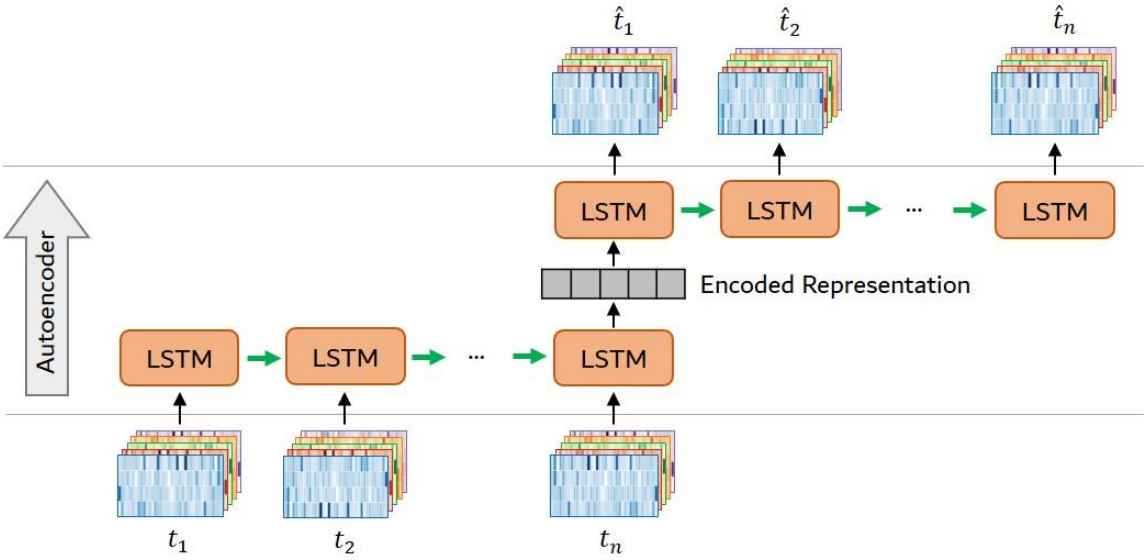

Figure 6.4: Schematic *recurrent autoencoder* approach.

The model considers the spatial map of the current time point as well as the extracted hidden state of the spatial maps of the previous time points. That means it considers the spatial and temporal information of the data in one model. The other approaches only consider one of these two information and ignore the other one.

7. Anomaly detection with clustering of time series

The anomaly detection is implemented with the *clustering of time series* approach as described in section 6.2.

In comparison to the other evolved approaches it has some advantages. The model does not need knowledge about the physical processes or parameters of the observed PTPP as it is required in the *efficiency model* approach. Furthermore, it uses well known machine learning methods that often need less computational power than the *autoencoder* and *recurrent-autoencoder* approach, which uses methods of deep learning.

Beside these advantages, the clustering of the time series could be used for other purposes and not only for anomaly detection. The clustering could also be used to identify certain operational behaviors or states of the PTPP. Possible clusters could be the passage of clouds, the ramp-up or ramp-down of the PTPP, the cleaning of mirrors or the normal operation.

This chapter describes how the clustering of time series approach is implemented and shows the result of the implemented model. For the visualization of solar field data and results of applied methods of this approach a GUI is implemented as described in appendix B.

7.1. Input samples and dataset

This section describes which data is considered for building the anomaly detection model. It points out the used features within the input samples and the scope of the dataset.

7.1.1. Definition of input samples

The inputs for this model are multivariate time series that correspond to a certain observed section of the solar field. The sections are specified as loops to make a compromise between a fine spatial resolution and a low amount of data to process. If the sections are specified as subfields, the spatial resolution is too rough to identify the location of a detected anomaly. If the sections are specified as collectors, the amount of data, which needs to be processed, would be a lot higher.

The temporal resolution is set to the approximate measuring frequency of the measured loop features.

In short, the resolution of the subsequent approach is set to the following:

- Spatial resolution: Loop
- Temporal resolution: Period of $10\,s$

One sample is a multivariate time series. The maximal duration of one sample is set to one day. By doing this the daily and yearly seasonality is not considered. But the seasonality is rated as less important, because for normal behavior it is only important that the solar field data fits to the current conditions.

The features of the time series correspond respectively to the observed loop. The following features are considered:

- Estimated inlet temperature of loop
- Outlet temperature of loop
- Estimated mass flow of loop
- Mean SCA angle deviation of loop
- Spatially interpolated DNI of loop

The outlet temperature is directly measured. The features volume flow, SCA angle deviation, and DNI are extracted as described in section 5.3.2. The loop inlet temperature is estimated from the subfield inlet temperature and is provided in the PTPP data. The estimation considers the volume flow and therefore the time delay between inlet of subfield and inlet of respective loop.

7.1.2. Dataset

In total, three years of measurement data from the PTPP operation company is provided. The data has no gap in time. Due to computational power the entire data cannot be used. Therefore, 24 days are selected from the entire data. The days are sampled from the whole year 2015. They are randomly picked under the condition that there are on average two days per month. Over the 24 days there is a random variation of weekdays as well as weather conditions like cloudy, sunny, hot and cold days.

The total number of samples is determined by the number of loops of the PTPP and the number of days. With 152 loops of the reference PTPP and 24 days there are in total 3648 samples.

7.2. Distribution analysis and trimming

The distribution of the underlying data is analyzed by the histogram of each feature as shown in Figure 7.1. The entire dataset of 24 days is considered for this analysis.

Most of the features show two main value ranges: One at night times, and another one at daytime. The inlet temperature at daytime is between $250°C$ and $300°C$. At night it cools down to $100°C$ - $250°C$. The outlet temperature at daytime is controlled to around $395°C$, if feasible. At night the HTF also cools down to $100°C$ - $250°C$. The mass flow at daytime is between $2.5\ kg/s$ and $7.5\ kg/s$ in normal operational mode. In the

morning and evening hours the PTPP is in partial operational mode and has a mass flow of around $1.2\ kg/s$. At night the mass flow is nearly zero. The SCA angle deviation only takes positive values because it takes the absolute value of the difference between the SCA angle and the track angle as described in 5.3.2. At night hours the SCA angle deviation is artificially set to $0°$. At daytime the SCA angle deviation approaches zero when the focusing factor is controlled as described in 2.5.1. The SCA angle deviation is higher if the SCAs are defocused. This occurs due to a cleaning of mirrors, overheating of the loop outlet, or overload of the power block. The DNI is around $0\ W/m^2$ at night or when clouds are passing by. At clear sky it reaches values at around $900\ W/m^2$ and at noon up to $1100\ W/m^2$.

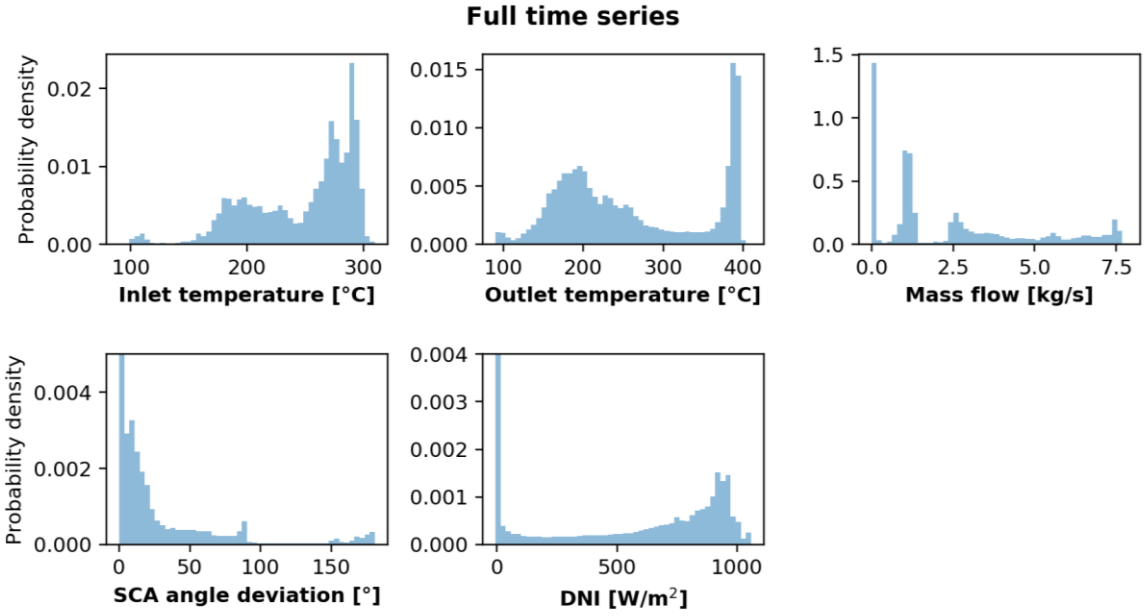

Figure 7.1: Histograms of features considering full time series.

To avoid two peaks in the histogram (one at night, and another at daytime) the time series are trimmed. The periods during the night and the morning and evening hours are ignored for the further anomaly detection. The actual operation of the PTPP is at daytime when the sun shines. The values at night time are irrelevant and can be neglected.

To identify the solar field's start and end times the track angle is chosen as auxiliary. In theory the track angle is $-90°$ when the sun rises and $90°$ when the sun goes down. However, the start point is set to the time when the track angle is $\varphi_{track} = -70°$, so that the PTPP have already been ramped-up. The end point is set to the time when the track angle is $\varphi_{track} = 80°$. With this setting the inlet and outlet temperature, mass flow and DNI are in the most cases already higher as in the night. At the end point the values are still at a day-operation level before they decrease in the evening hours. In the example which is shown in Figure 7.2 the night hours are between 0:00 am and 7:30 am

as well as 21:00 pm and 24:00 pm. The ramp-up of the PTPP is between 7:30 am and 9:10 am. The ramp-down is between 18:15 pm and 21:00 pm. The normal operation is between 9:10 am and 18:15 am.

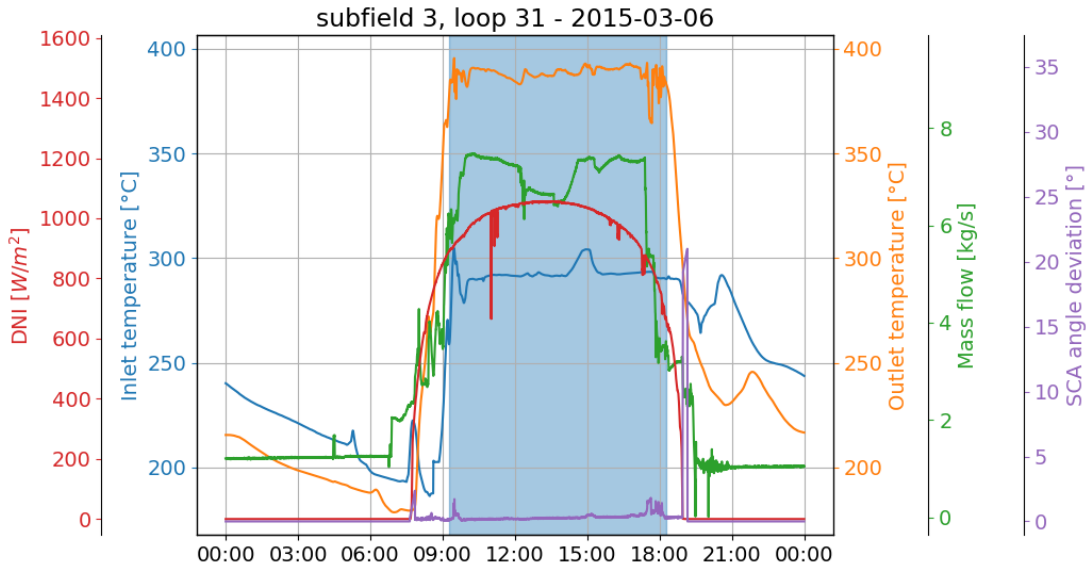

Figure 7.2: Example for trimmed time series with $\varphi_{track} = -70°$ as starting point and $\varphi_{track} = 80°$ as end point.

The trimmed time series changes the distribution of the observed data compared to the distribution of the full or non-trimmed time series. The frequency of values during the ramp-up and ramp-down in the morning and evening hours are reduced (see Figure 7.3). This enables a better scaling of the features with z-score standardization in further methods.

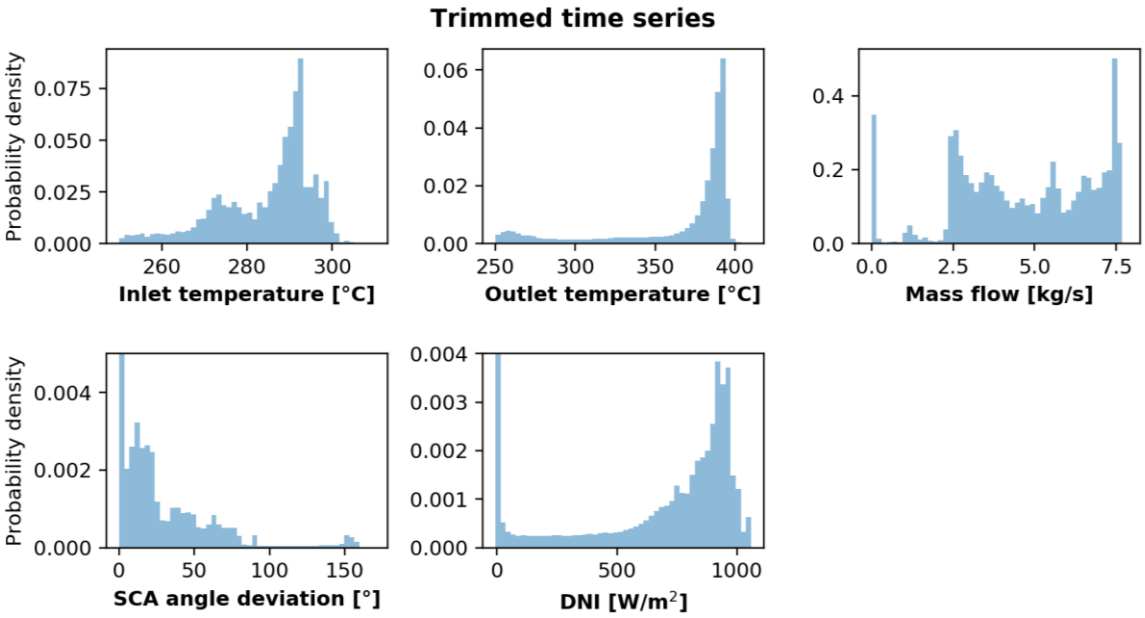

Figure 7.3: Histograms of features considering trimmed time series.

7.3. Time series segmentation

The segmentation splits the time series of long duration in smaller homogeneous pieces, called segments. The data in each segment can be described accurately by a simple model. It reduces the complexity of the representation of the original data. [23]

The segments are later clustered by a clustering algorithm. This has the advantage that the time series is not clustered as whole piece. All the segments of the time series can be assigned to different clusters. Clustering small time ranges of a time series allows a more detailed clustering.

It is important that on the one hand the duration of segments is not too short. A small segmentation leads to more segments, which need to be processed by the clustering algorithm. This might end in a very long runtime of the clustering.

One the other hand the duration of segments should not be too long. In too long segments a relatively short anomalous event might be overseen, because the rest of the data is dominant. Even if an anomalous event is not overseen, it is harder to make an estimation of the exact time point of the anomalous event.

To find a well fitted segmentation three methods are validated:

- PCA-based segmentation
- Window-sliding segmentation
- Periodic segmentation

7.3.1. PCA-based segmentation

The segmentation method as described in [24], is denoted as PCA-based segmentation in this research. It is a bottom-up approach and works for multivariate time series. Bottom-up algorithms promise in most cases significantly better results than top-down or sliding-window algorithms. This statement refers to the survey in [25].

The rough concept of the PCA-based segmentation is as follows: First, find a fine initial segmentation. Then reduce the dimensionality of each segment by applying a PCA. After that, the reconstruction error of the PCA of each segment is used as a criterion to possibly merge adjacent segments. The merged and non-merged segments build the final segmentation.

The initial segmentation uses change points based on extreme values of each individual signal. Other change points could also be based on inflection points or saddle points. Nevertheless, due to many extreme values of the signals and the number of dimensions, the extremes provide an initial segmentation that is fine enough to start with.

For each dimension the minima and maxima are calculated. The time points of the minima and maxima of every dimension are the breakpoints for the initial segmentation.

The average duration of an initial segment depending on the feature is shown in Table 7.1. The average is built over 100 non-trimmed samples out of the dataset with 24 days.

Table 7.1: Average duration of initial segments of individual signals or features.

Feature	Average duration
Inlet temperature	247.8 s
Outlet temperature	451.3 s
Mass flow	21.5 s
SCA angle deviation	21.0 s
DNI	30.2 s

If every local extreme value is taken, the initial segmentation is very fine for the features mass flow, SCA angle deviation, and DNI as shown in Figure 7.4 In the figure the turquoise lines show the time points of extreme values. The high number of extremes is due to the frequent and fast fluctuation of the measurands. The inlet and outlet temperature fluctuate relatively slow due to the thermal inertia of the fluid and the sensing element itself.

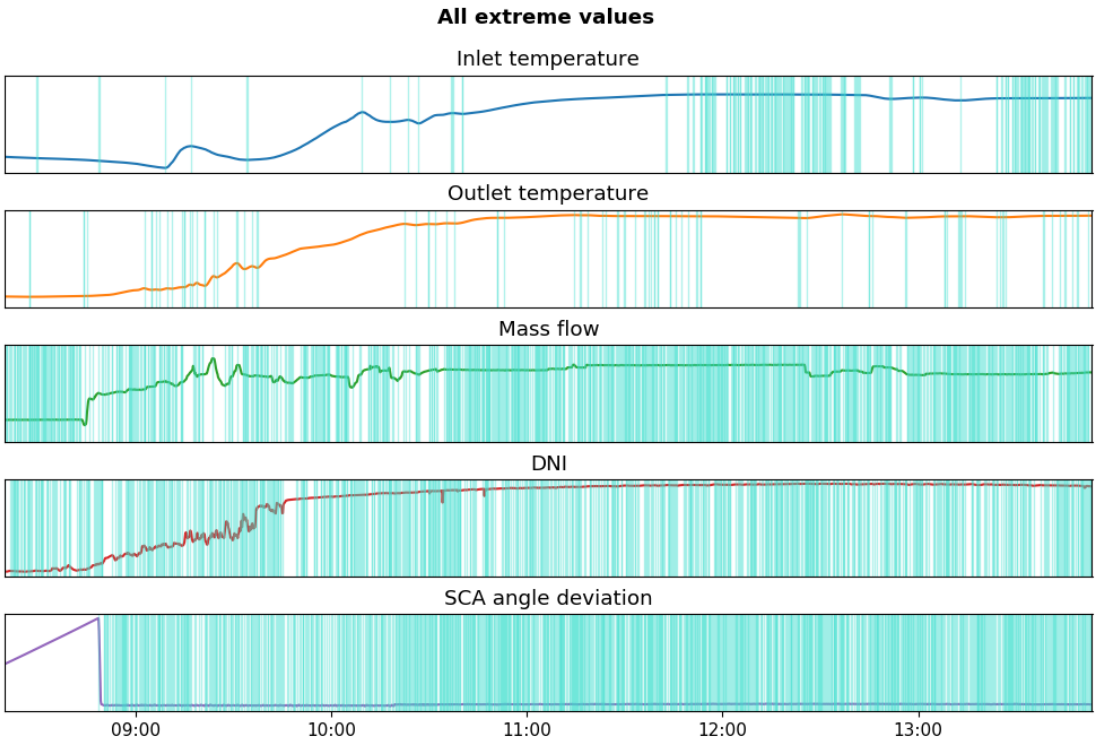

Figure 7.4: All extreme values of example data sample.

The average duration of the initial segments over all dimensions is shorter than the shortest average duration of each individual signal. This means the duration of the initial segmentation would be shorter than 21.0 s in this case. Such a fine segmentation results in a lot of segments and could lead to runtime problems by processing the further PCA on each segment. The number of change points needs to be reduced.

The trial to decrease the number of extreme values by smoothing the signals with a moving mean or Savitzky-Golay filter does not produce the intended results. Even with a high value for the window length of the Savitzky-Golay filter the average duration of segments does not increase significantly (see Figure 7.5). This is due to the fact that the smoothing does not eliminate the fluctuation of the signal. It just reduces the range of the fluctuation. But the signal is still fluctuating, which leads to a similar number of extreme values.

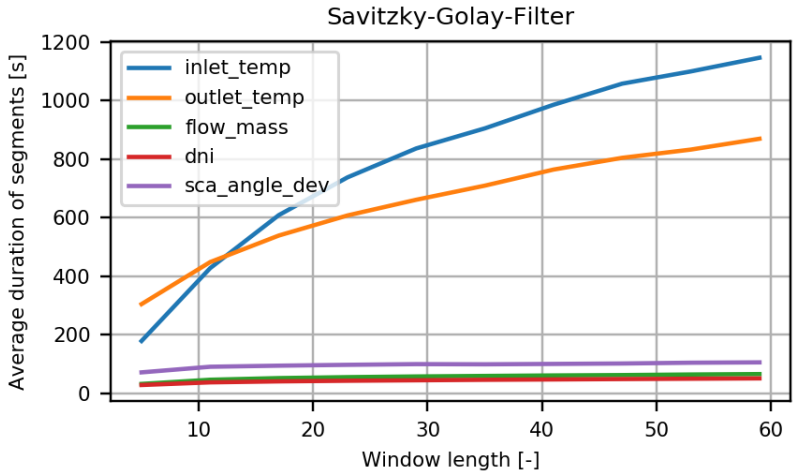

Figure 7.5: Average duration of segments depending on the window length of Savitzky-Golay filter for smoothing the signal.

Therefore, the number of change points is reduced by dropping insignificant extreme values. Hence, an extreme value must differ significantly from the previous extreme value. This means an extreme value e must lay out of a certain range around the previous extreme value e_{t-1}:

$$(e_t < e_{t-1} - o_{rel}) \text{ or } (e_t > e_{t-1} + o_{rel}) \tag{7.1}$$

Whereby o_{rel} is a relative offset which is defined by a dropping tolerance τ_{drop} on the maximal range R of the respective signal.

$$o_{rel} = R \cdot \tau_{drop} \tag{7.2}$$

The average duration of the segments is calculated for a tolerance between 0 and 0.01 as shown in Figure 7.6.

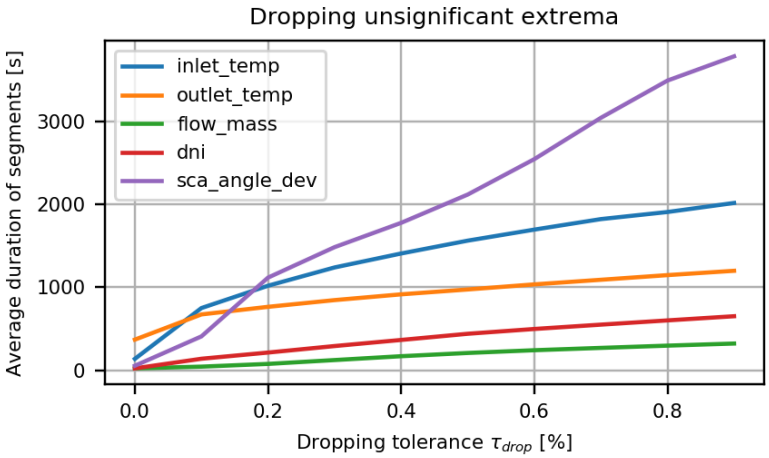

Figure 7.6: Average duration of segments depending on the tolerance of dropping extreme values.

Figure 7.7 shows the initial segmentation of the same data sample as in Figure 7.4 but with the dropping tolerance of $\tau_{drop} = 0.7\,\%$. The turquoise lines show the time points of extreme values. The result shows less change points, but the significant extreme values remain.

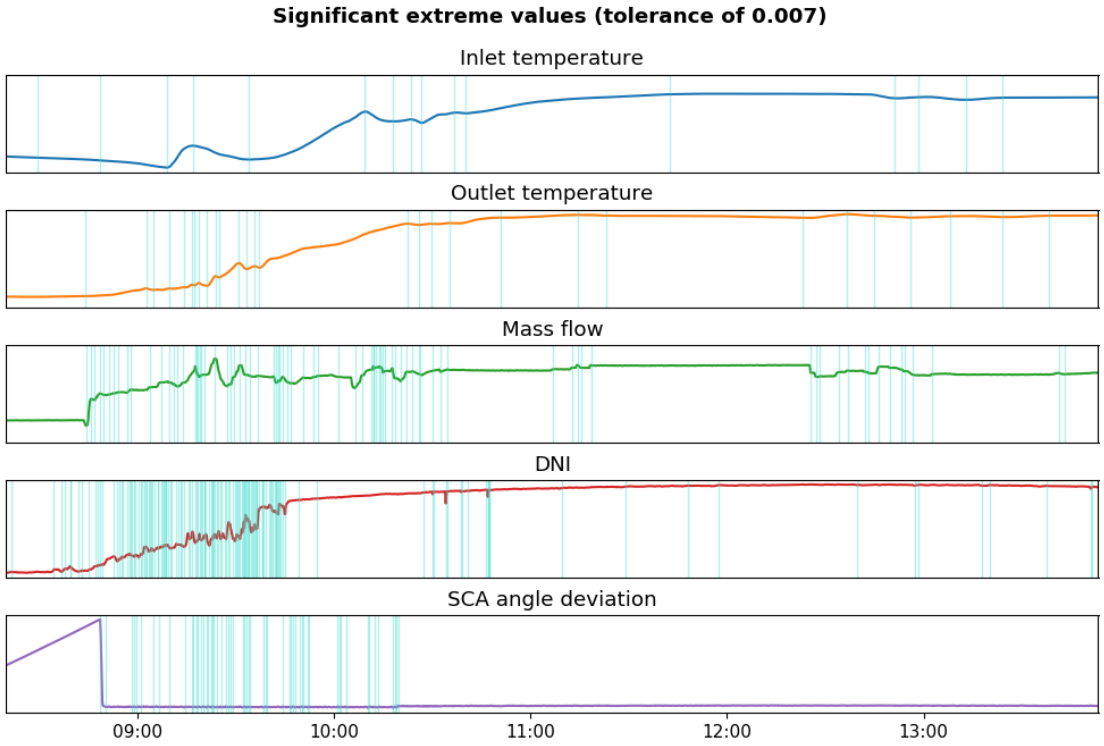

Figure 7.7: Remaining significant extreme values of example data sample.

The dropping tolerance of insignificant extreme values is set to $\tau_{drop} = 0.7\,\%$ for further segmentations.

The resulting initial segmentation is the base for the merging of the segments. On each initial segmentation a PCA is applied. Therefore, each feature is scaled with a z-score standardization. For the z-score standardization the entire dataset of 24 days is considered.

Out of the PCAs first two principal components the original data x is reconstructed to the projection p. The resulting reconstruction error is used as basis for the segment fusion.

A segment is defined by $s_t(a_t, b_t)$ where a_t is the starting time point and b_t the ending time point of the segment. The segments original data x, with N time points, is reconstructed to its projection p. The reconstruction error is defined as the squared distance between the original and projected values of the time series over all D dimensions.

$$error\big(s_t(a_t, b_t)\big) = \frac{1}{N \cdot D} \sum_{d=1}^{D} \sum_{i=1}^{N} (x_{d,i} - p_{d,i})^2 \tag{7.3}$$

For a fusion of two segments it is considered the reconstruction error of the first segment $e_1 = error(s_t(a_t, b_t)$, of the second segment $e_2 = error\ (s_{t+1}(a_{t+1}, b_{t+1})$, and the fusion segment of the first and second segment $e_{1,2} = error(s_{t,t+1}(a_t, b_{t+1})$.

A fusion takes place, whenever the sum of the reconstruction errors of the individual segment with a fusion tolerance M is greater than the reconstruction error of the fusion segment.

$$e_{1,2} < (e_1 + e_2) \cdot M \tag{7.4}$$

The fusion process starts with the first pair of consecutive initial segments at the beginning of the time series. If a fusion takes place, the next considered pair of segments is the resulting fusion segment and the next segment. If no fusion takes place, the next considered pair of segments is the second segment of the current pair and the next initial segment. The fusion process ends after the consideration of the last initial segment at the end of the time series.

The hyperparameter for the PCA-based segmentation is the tolerance M that controls the fusion of the initial segments.

7.3.2. Window-sliding segmentation

The window-sliding algorithm as described in [26] is a fast and relatively simple method for change point detection. It is an approximate search method, that yields towards an approximate solution of the change points.

The algorithm computes the discrepancy d between two adjacent windows. The window pair slides along a signal y of length T. For the two adjacent windows and the joined window there are calculated costs for a given cost function $c(y)$. The discrepancy between two windows is given by

$$d(y_{a,t}, y_{t,b}) = c(y_{a,b}) - c(y_{a,t}) - c(y_{t,b}) \quad (1 \leq a < t < b \leq T) \tag{7.5}$$

In this research the L2 cost function is used, which is defined by

$$c(y_{a,b}) = \sum_{t=a}^{b} (y_t - \bar{y}_{a,b})^2 \tag{7.6}$$

The discrepancy reaches large values, when the segments of the two windows are dissimilar. It is calculated for each time point of the time series and results in a discrepancy curve (see Figure 7.8). Peaks in the discrepancy curve are identified as change points. [26]

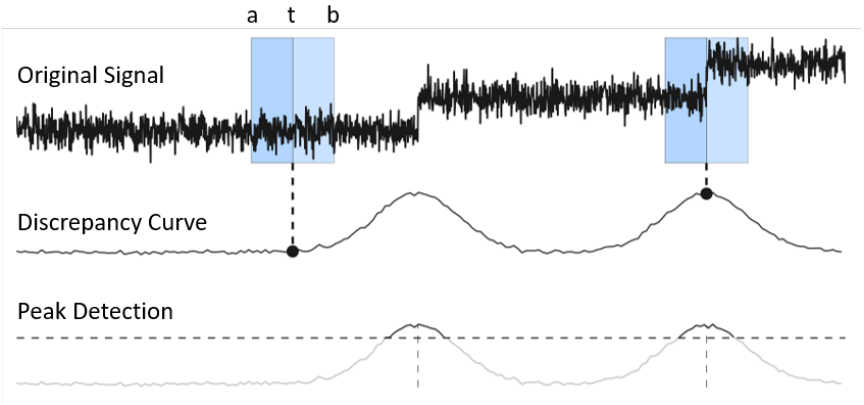

Figure 7.8: Schematic view of window-sliding algorithm. [26]

The window-sliding algorithm helps finding the breakpoints for the segmentation. Therefore, the time points of the signal are sorted by its discrepancy. The time points, considered as breakpoints, are successively added to a list of breakpoints b beginning with the breakpoint with the highest discrepancy. For each iteration a reconstruction error is calculated by

$$error(b) = N \cdot \sum_{i=b_1}^{b_N} Var(x_{i,i+1}) \tag{7.7}$$

where N is the current number of breakpoints and x the time series.

If $error(b)$ becomes greater than a certain threshold ε the list of breakpoints b is returned. The resulting segmentation depends on the window length and the threshold ε, which are considered as hyperparameter.

That procedure is done for each feature of the time series. The breakpoints of every feature result in the total segmentation.

Due to the calculation of the reconstruction error (see equation (7.7)), the scale of the features has an impact on the error value. Hence, each feature is scaled with a z-score standardization. For the z-score standardization the entire dataset of 24 days is considered.

7.3.3. Periodic segmentation

Another method for segmentation is to divide the time series in regular pieces. The duration of segments is defined by the number of points n_{seg} for each segment. It is the simplest way of segmentation. Figure 7.9 shows an example of a segmentation with $n_{seg} = 60$ that corresponds to a duration of $\Delta t_{seg} = 10\ min$ for each segment.

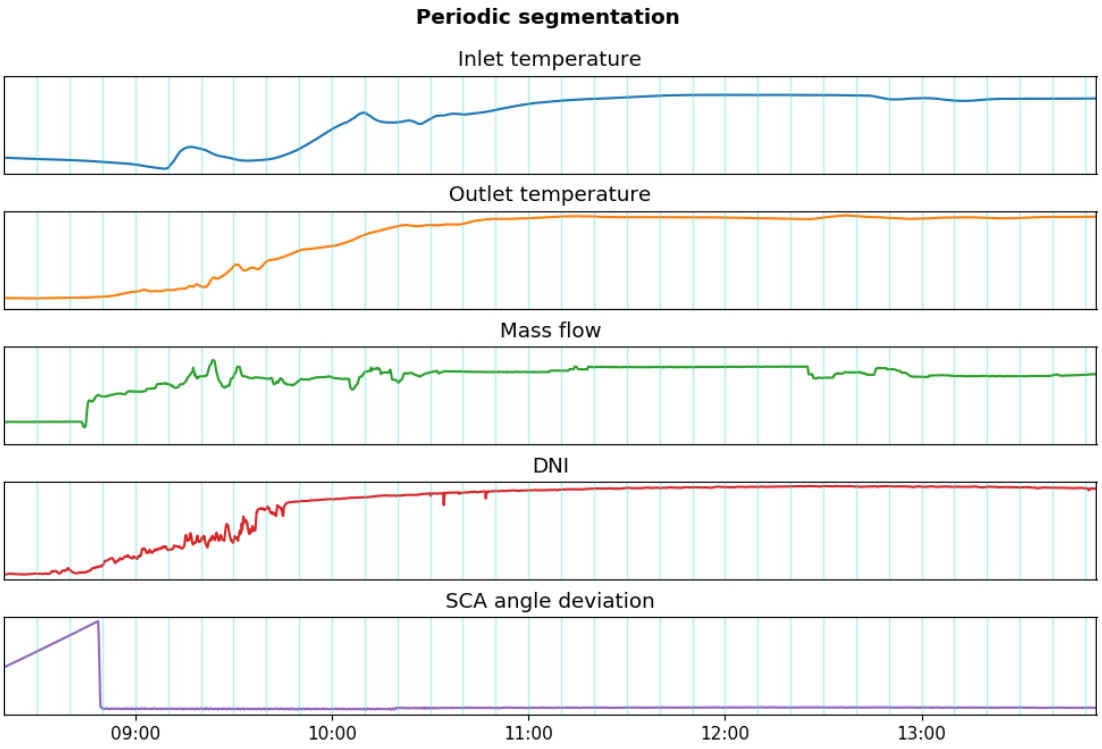

Figure 7.9: Example for periodic breakpoints for segmentation.

7.3.4. Evaluation of segmentation performance

The optimal hyperparameter of the segmentation method is obtained with the help of a grid search. Therefore, the segmentation is done with different determined values for the hyperparameters. For each resulting segmentation a validation or segmentation score is calculated.

The segmentation score is designed so that the higher its value, the better the segmentation. To do so, it needs to evaluate the placement of the breakpoints as well as the number of segments.

Evaluate placement of breakpoints

The placement of the breakpoints is evaluated with the help of a piecewise aggregate approximation (PAA). An PAA is usually used to represent a time series with a reduced number of time points. The number of time points is reduced to the number of segments. The time points of the PAA lay in the center of each segment and its value is the mean of the respective segment.

In our case, a time series with the same number of time points as the original time series is built out of the PAA. The values of all time points within one segment are the mean of this segment. This is done for each dimension separately. Figure 7.10 shows a resulting PAA time series (black lines) for a given segmentation (turquoise lines).

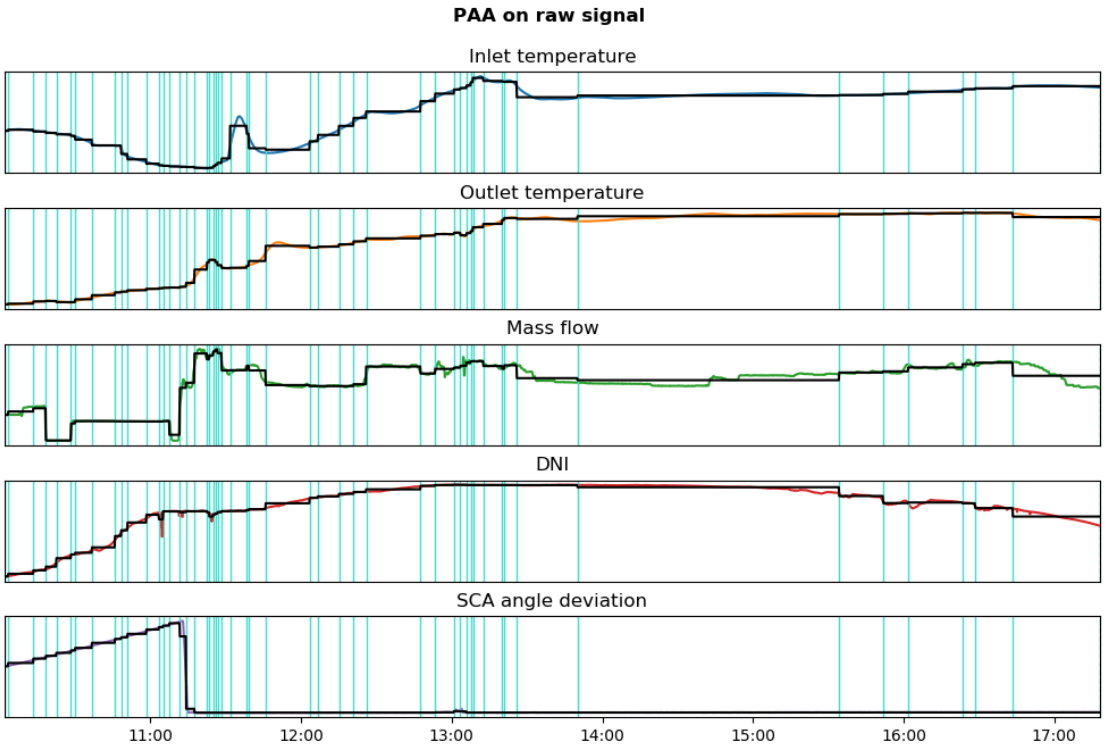

Figure 7.10: Example of PAA on raw signal.

Distance and evaluation score of PAA

The distance $dist(x,p)$ between a time series x and its time series built out of its PAA p is calculated with the Euclidean distance:

$$dist(x,p) = \frac{1}{N \cdot D} \sqrt{\sum_{d=1}^{D} \sum_{i=1}^{N} (x_{d,i} - p_{d,i})^2} \tag{7.8}$$

Whereby D is the number of dimensions and N the number of time points of the time series.

To get a score value in a range of $0 < s_{paa} < 1$, where zero stands for the worst and one for the best segmentation, a scaling with the maximal distance is done:

$$s_{paa} = 1 - \frac{dist(x,p)}{dist(x,p_{worst})} \tag{7.9}$$

The maximal distance is obtained with the worst segmentation $\mathrm{p_{worst}}$, which considers the whole time series as one segment.

PAA on original signal and its moving standard deviation

The PAA is performed for each dimension on two different signals. One is done on the original data of the signal and another one on the calculated moving standard deviation of the signal.

The reason for the first PAA is explained with the example shown in Figure 7.11. In the example the mean of the original signal jumps at a certain time point.

A good segmentation would be to have a breakpoint at the change point of the mean. In this case the time series of the original signal and its PAA are very similar, which leads to a low distance between these two time series.

A bad segmentation would be to have a breakpoint at a random other time point. In this case the time series of the original signal and its PAA are very dissimilar. This leads to a high distance between these two time series.

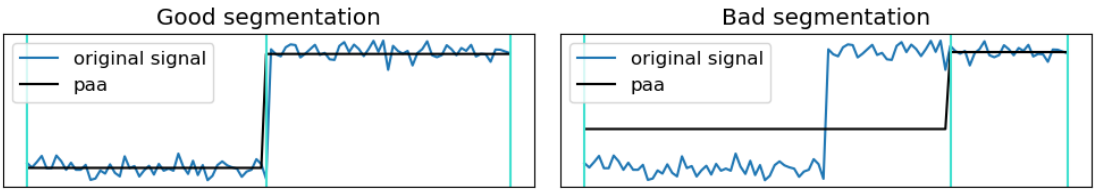

Figure 7.11: Example for PAA on original signal comparing a good and bad segmentation.

The reason for the second PAA with moving standard deviation of the signal is explained on another example shown in Figure 7.12. In this example the mean of the whole original signal is constant, but the standard deviation changes at a certain time point.

A good segmentation would be to have a breakpoint at the change point of the standard deviation. A bad segmentation would be to have a breakpoint at a random other time point. The PAA of the original signal would be equal for the good and bad segmentation in this example. See upper left and upper right plot of Figure 7.12. Therefore, it results in the same loss between the original time series and its PAA.

To overcome this problem the PAA is done on the moving standard deviation of the signal. The calculation of the standard deviation is done with a small window length of $w = 3$. The good segmentation results in a lower distance between the moving standard deviation of the signal and its PAA. See lower left and lower right plot of Figure 7.12.

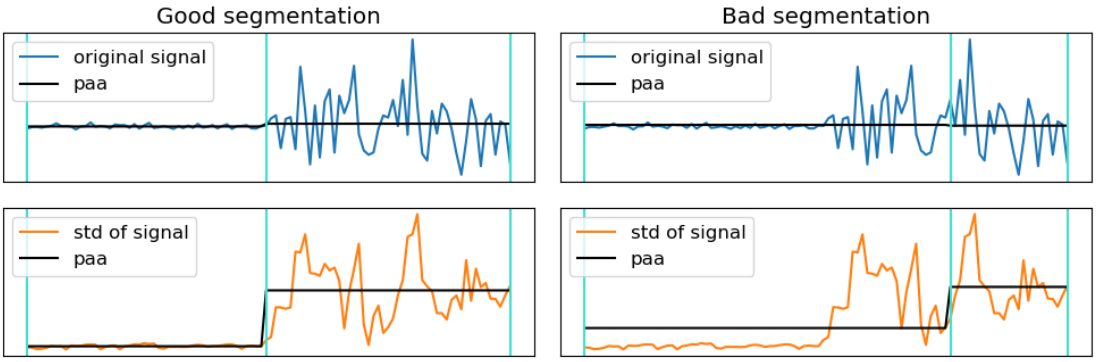

Figure 7.12: Example for PAA on calculated standard deviation of the original signal comparing a good and bad segmentation.

The PAAs of the original signals and its moving standard deviations result in the evaluation scores $s_{paa,org}$ and $s_{paa,std}$.

Evaluate number of segments

A segmentation that is only evaluated with a consideration of the placement of the breakpoints as it is described in the subsection above would lead to very small segments. This is because the smaller the segments the better the calculated PAA. In this case, the best segmentation would have segments that have the length of one time point.

To overcome this problem, a partial score is implemented that works as penalty of small segments. This penalty score s_{n_segs} considers the number of segments with

$$s_{n\_segs} = 1 - \frac{n_{segs}}{N} \tag{7.10}$$

whereby n_{segs} is the number of segments and N the number of time points.

Total evaluation score

The total evaluation score or segmentation score is calculated with the three partial scores: The score of the PAA with the original signal $s_{paa,org}$, the score of the PAA with the moving standard deviation of the signal $s_{paa,std}$, and the score of the number of segments s_{n_segs}. Each partial score is weighted by an exponent.

$$s_{seg} = \sqrt[W]{{s_{paa\_org}}^{w_1} \cdot {s_{paa_{std}}}^{w_2} \cdot {s_{n\_segs}}^{w_3}} \tag{7.11}$$

The order of the root W is the sum of all weights

$$W = w_1 + w_2 + w_3 \tag{7.12}$$

The multiplication of the partial scores provide a balancing between all scores. A summation of the partial scores could lead to a domination of one score. For example, if we get the partial scores $s_{paa\_org} = 0.1$, $s_{paa\_std} = 0.1$, and $s_{n\_segs} = 1$ the total segmentation score becomes $s_{seg} \approx 0{,}22$ with $w_1 = w_2 = w_3 = 1$. If we get more balanced partial scores with the same sum as in the previous example, e.g. $s_{paa\_org} = 0.4$, $s_{paa\_std} = 0.4$, and $s_{n\_segs} = 0.4$, the total segmentation score becomes $s_{seg} = 0{,}4$. The total segmentation score can be seen as volume, which we try to maximize.

An empirical investigation shows that a weighting with $w_1 = 1$, $w_2 = 1$, and $w_3 = 40$ results in a segmentation with a balanced number of segments.

Evaluation results

For each introduced segmentation method, a grid search is done to find the optimal parameters.

The result of an empirical investigation indicates, that a window length of $16 < n < 22$ for the window-sliding segmentation results in a good placement of the breakpoints. Therefore, the window length is set to $n = 18$ to simplify the hyperparameter optimization with only the error threshold ε as hyperparameter.

Table 7.2 shows the considered hyperparameter and its range for each segmentation method. The values for the hyperparameters are received from the closed intervals and the step sizes Δ.

Table 7.2: Considered hyperparameter and its range for each segmentation method.

Segmentation method	Hyperparameter	Values
PCA-based	Fusion tolerance M	$[1.0, 1.22]$ with $\Delta = 0.02$
Window-sliding	Error threshold ε	$[20, 300]$ with $\Delta = 20$
Periodic	Number of points n_{seg}	$[20, 100]$ with $\Delta = 10$

Each hyperparameter value results in a different segmentation. For each segmentation the segmentation score is calculated as described in the previous subsection. The results are shown in Figure 7.13. The periodic segmentation reaches the highest segmentation score of $s_{seg} \approx 0.941$ with $n_{seg} = 80$. The score of the best window-sliding segmentation with $\varepsilon = 160$ is slightly under the best score of the periodic segmentation. The scores of the PCA-based segmentation are lower than the scores of the other both segmentation methods in each case.

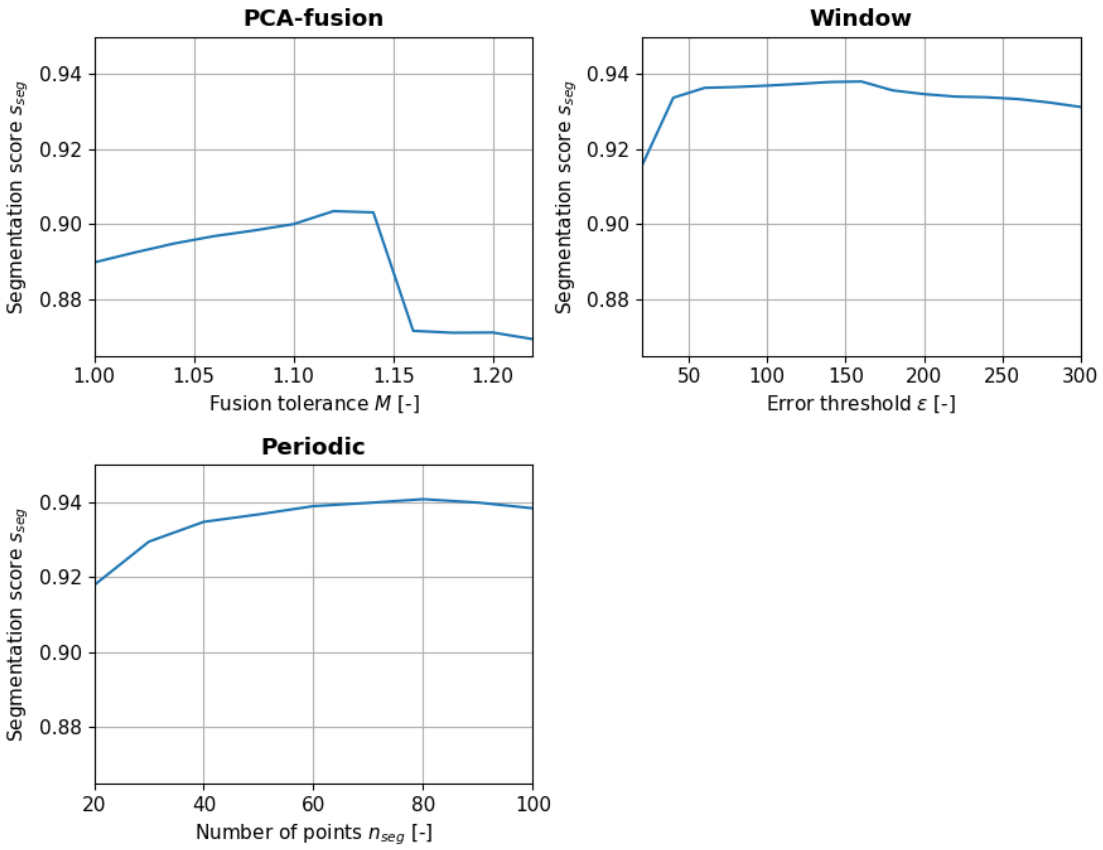

Figure 7.13: Segmentation scores for the hyperparameter grid search on the different segmentation methods.

The example in Figure 7.14 shows why even the best window-sliding segmentation is over all samples on average worse than the periodic segmentation. In some areas the

segmentation is very fine (e.g. 13.00 pm until 14.30 pm) and results in a high number of segments. In other areas the segmentation is very sparse (e.g. 14.30 pm until 19.00 pm) and results in a high reconstruction error. The PCA-based segmentation shows a similar behavior.

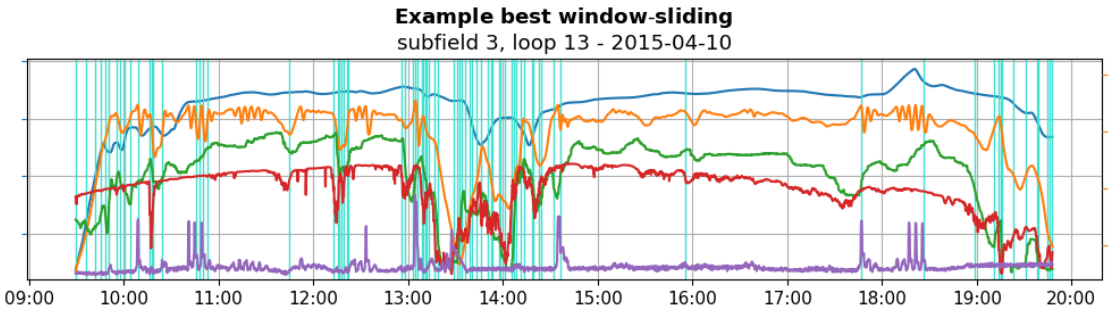

Figure 7.14: Best window-sliding segmentation on example loop.

In comparison to the other methods the best periodic segmentation has a smaller number of segments with a good representation of the time series for each segment. Therefore, the segmentation is done with the periodic method with $n_{seg} = 80$, which corresponds to a segmentation period of approx. $13\,min$.

7.4. Feature extraction

After the segmentation of the time series, the input data for the clustering algorithm needs to be created. The implemented clustering is a density-based clustering algorithm that can have a distance or feature matrix as input.

One disadvantage of building a distance matrix is that it needs segments of the same length, so that the distance between two segments can be calculated. Furthermore, the computational time complexity is $O(n^2)$, where n is the number of segments. That means, with an increasing number of segments the runtime increases quadratic.

The computational time complexity to calculate a feature matrix is $O(n)$. Because of the lower time complexity, a feature matrix is calculated as input of the clustering algorithm.

To build a feature matrix, features from each segment have to be derived. For time series thousands of interpretable features can be derived. The derived features are chosen in terms of correlation structure, distribution, entropy, stationarity and scaling properties [27]. In this research the following features are extracted:

- Mean values of each individual dimension
- Standard deviations of each individual dimension

It is only chosen these two kinds of features to avoid the problem of high-dimensional data. Furthermore, the features mean and standard deviation are well understandable.

Five dimensions and two features per dimension result in a total number of $m = 10$ extracted features.

The distribution of the data in the feature space has a strong impact on the clustering result of a density-based clustering algorithm. Hence, the feature vectors X_d are scaled with the z-score standardization.

$$Z_d = \frac{X_d - \mu}{\sigma} \tag{7.13}$$

Whereby, μ is the mean value and σ the standard deviation of the feature vector X_d.

The vertical concatenation of the standardized feature vectors Z_d build the feature matrix F. The feature matrix has the size $n \times m$, where n is the number of segments or points and m the number of extracted features. The trimming described in section 7.2. and the segmentation described in section 7.3 on the entire dataset of 24 days result in a total number of $n = 155{,}368$ segments or points. The number of extracted features is $m = 10$.

7.5. Time series clustering

The time series clustering is done with the DBSCAN algorithm as described in section 3.3.2. The advantage of using a density-based clustering is that it can find clusters of arbitrary shape. In Figure 7.15 the different clustering results between a DBSCAN and k-means clustering on various datasets are shown. K-means is a very common distance-based clustering algorithm. In most of these cases DBSCAN finds better clusters than the k-means algorithm. [28]

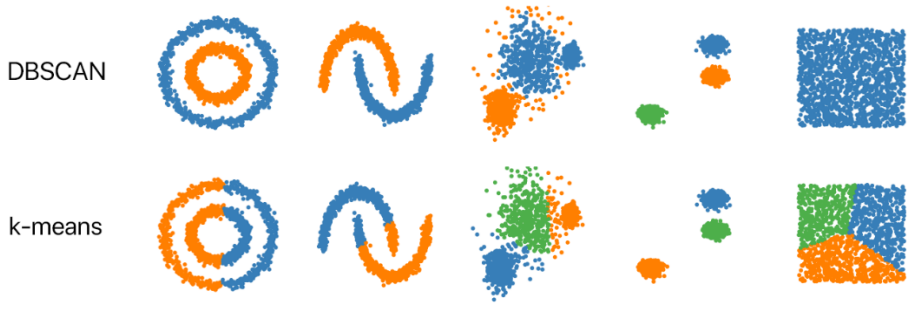

Figure 7.15: Examples for DBSCAN vs. k-means clustering. [28]

Another advantage of DBSCAN is that not every point is put into a cluster necessarily. DBSCAN categorizes points as core, border or outlier points. In the context of anomaly detection, the outlier points can be referred to as anomalies. They do not belong to any cluster.

The runtime complexity of DBSCAN becomes $O(n^2)$ in the worst case [16]. Due to computational power of the provided Hardware, the number of processed loop samples is limited to 1000. In average there are 43 segments per loop sample. Hence, around $n = 43{,}000$ points are expected for the clustering. The 1000 loops are randomly picked from the dataset. The randomly determined loops give a total number of $n = 41{,}984$ segments that lead to the same number of points in the feature space.

7.5.1. Hyperparameter search

The clustering results depend on the two hyperparameters radius r_{hood} and the minimum point number n_{min} as described in section 3.3.2.

The hyperparameter optimization is done with a grid search. Therefore, the range for the parameter n_{min} is set to the empirical range $3 < n_{min} < 40$.

The range for the parameter r_{hood} is narrowed with the help of the relative number of core points p_{core}. The relative number of core points is aimed to be $97\% < p_{core} < 99\%$. With the set ranges for the minimum point number n_{min}, and relative number of core points p_{core}, the range of the radius r_{hood} can be calculated.

Therefore, the mean distances of k nearest neighbors are calculated for each point and different k. The means are calculated for $k = 3$, which corresponds to the minimal n_{min}, and $k = 40$, which corresponds to the maximal n_{min}. For each k, the points are sorted by the mean distances (see Figure 7.16). In the next step, the 0.97-quantile $Q_{0.97}$ and 0.99-quantile $Q_{0.99}$ of the sorted mean distances are calculated. The resulting minimal mean distance is taken as minimal value for the radius r_{hood}. The resulting maximal mean distance is taken as maximal value for the radius r_{hood} (see red marks in Figure 7.16). This results in the range $0.75 < r_{hood} < 2.92$.

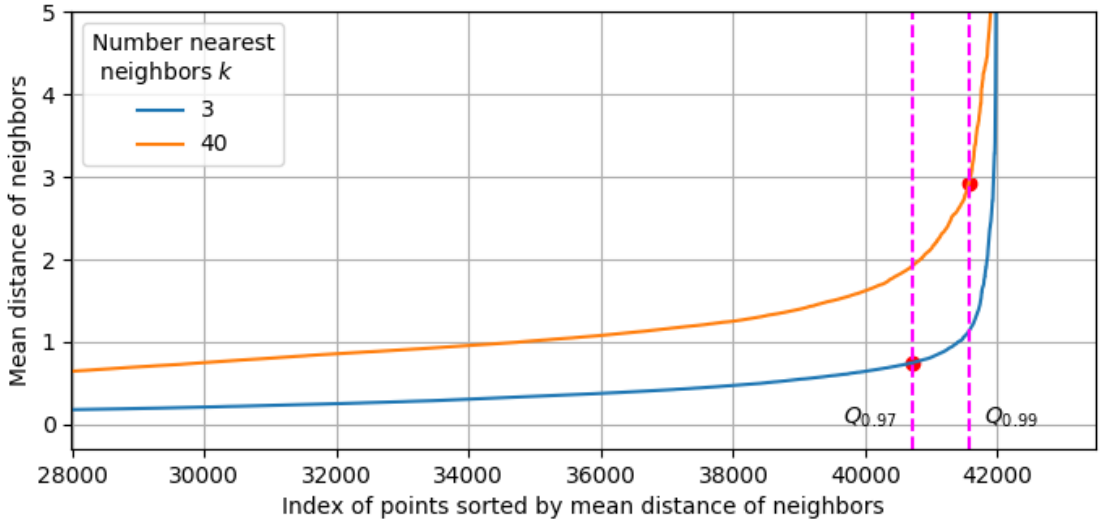

Figure 7.16: Sorted mean distances of k nearest neighbors.

For better understanding an example is given: If the DBSCAN clustering is done with $n_{min} = 3$ and $r_{hood} = 0.75$, then approximately 97 % of the points are core points. The others 3 % are border or outlier points.

The hyperparameter grid search is done with the above explained ranges for n_{min} and r_{hood}. Each DBSCAN clustering result in a certain number of clusters $n_{clusters}$. Figure 7.17 shows the numbers of clusters for different parameter settings. It shows that $n_{clusters}$ decreases with increasing r_{hood} or with increasing n_{min}.

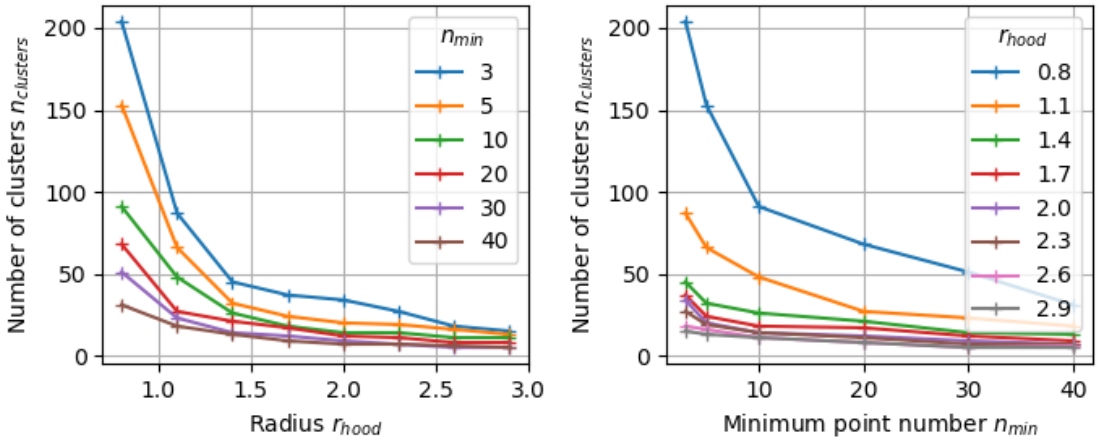

Figure 7.17: Number of clusters for different DBSCAN parameter settings.

Another informative variable about the clustering is the outlier ration $p_{outlier}$. It is the relative number of outlier points to the total number of points. Figure 7.18 shows the outlier ratio for different parameter settings. It shows that $p_{outlier}$ decreases with increasing r_{hood} or with decreasing n_{min}.

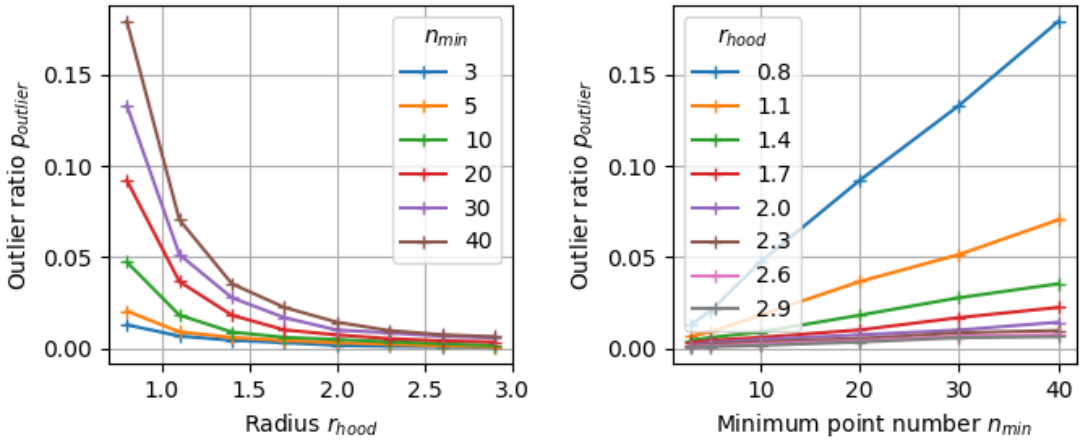

Figure 7.18: Outlier ration for different DBSCAN parameter settings.

The number of clusters $n_{clusters}$ is set to a limit of $n_{clusters} < 10$ to narrow the range for n_{min} and r_{hood} further. This results in the new ranges of $13 < n_{min} < 40$ and $1.6 < r_{hood} < 3$. A clustering within these ranges do not compulsorily lead to less than 10 clusters, but a clustering out of this ranges lead to more than 10 clusters. With the new ranges a grid search is done with a finer resolution.

It is aimed to achieve a clustering with $1 \leq n_{clusters} < 10$ to have a limited number of clusters and $0.7\,\% \leq p_{outlier} \leq 1.2\,\%$ to have a certain number of outlier points. The parameter pairs that fit these conditions are shown in Figure 7.19 in the blue color. Finally, the DBSCAN clustering is set with $r_{hood} = 2.2$ and $n_{min} = 31$ (orange point). The grey points are all other evaluated clustering settings.

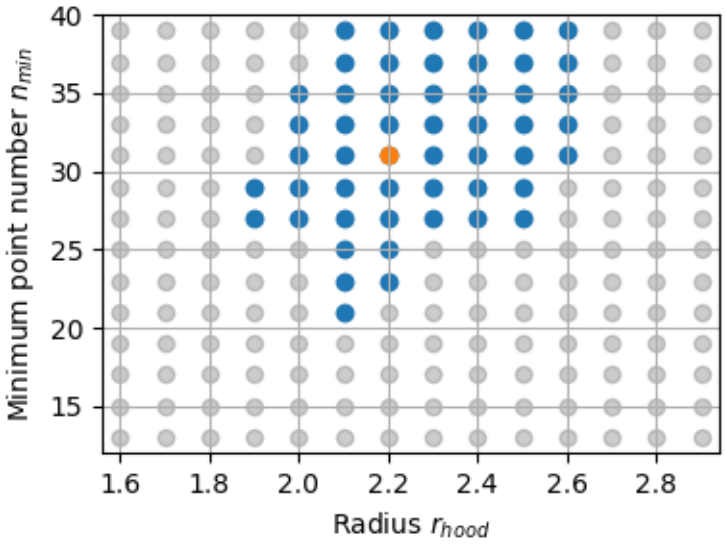

Figure 7.19: DBSCAN hyperparameter pairs.

7.5.2. Clustering results

The DBSCAN clustering with $r_{hood} = 2.2$ and $n_{min} = 31$ on the 1000 loops of the dataset results in 6 clusters. Table 7.3 shows the number of points within the respective cluster. The cluster C-1 is actually not a cluster, it is the collection of the outlier points, which are possibly anomalies. The number of 375 outlier segments refers to around $0.89\,\%$ of all segments. The cluster C0 is the biggest cluster and contains $98.48\,\%$ of the points. The clusters C1 to C5 are small clusters with around 49-56 points per cluster.

Table 7.3: Number of points of respective cluster

Cluster	**C-1**	**C0**	**C1**	**C2**	**C3**	**C4**	**C5**
Number of points	375	41,345	54	56	54	49	51
Percentage of data	0.89 %	98.48 %	0.13 %	0.13 %	0.13 %	0.12 %	0.12 %

Figure 7.20 shows the assignment of points to clusters in the feature space. The feature space is reduced to two dimensions with a PCA for visualization. Hence, sometimes outlier points seem to be very close to other clusters, but the distance in the actual feature space is bigger.

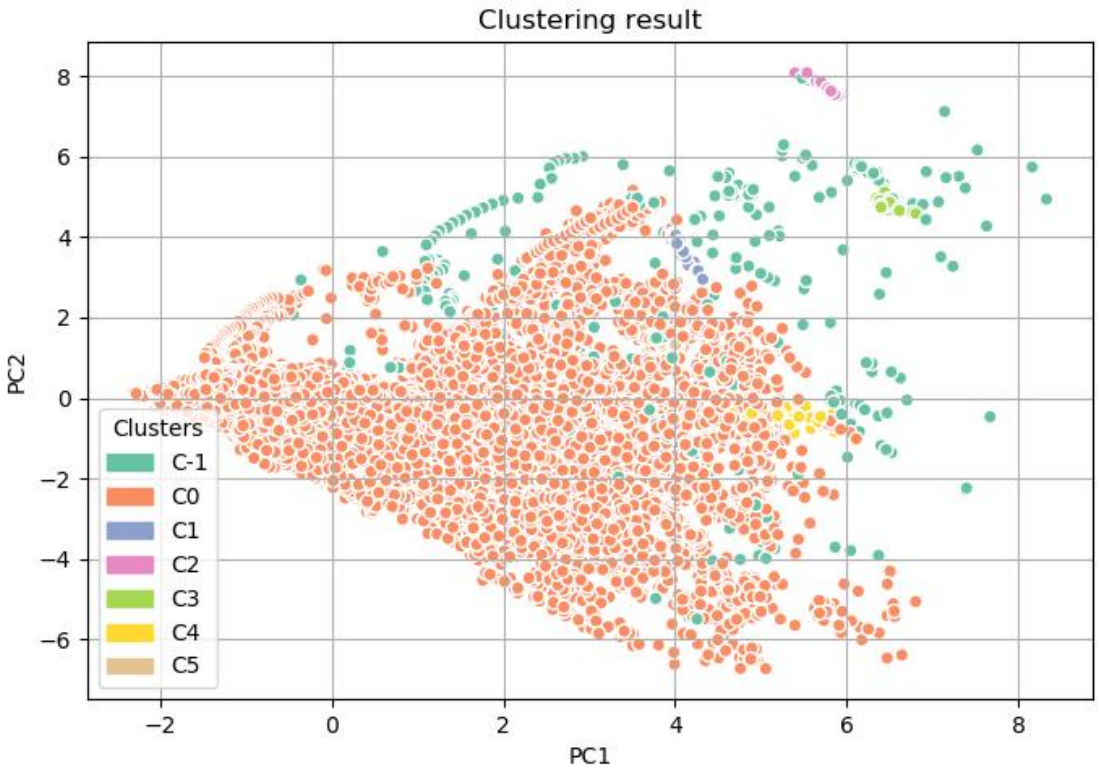

Figure 7.20: DBSCAN clustering result.

By analysis of the side clusters C2-C5 it is apparent that each side cluster only occurs at the same day. There are no segments of two or more different days assigned to a side cluster. Additionally, the clusters cannot be given any meaningful interpretation.

7.6. Evaluation of outlier segments

The segments that are clustered as outlier points by the DBSCAN clustering are possibly anomalies. In total there are 375 outlier segments. By analyzing the outlier segments these are categorized by possible cause of appearance (see Figure 7.21). This results in 266 segments which are related to a manual maneuver of the SCA, 84 segments which are related to maintenance work, 14 segments which are related to other reasons and 11 segments where no reason could be found.

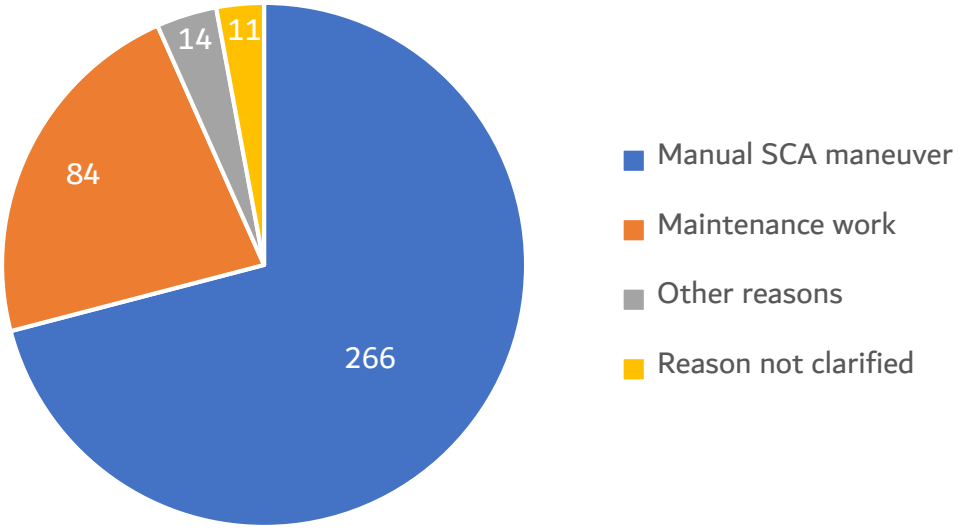

Figure 7.21: Categorized reasons for outlier segment.

The number of 374 outlier segments out of all outlier segments are not regarded as actual anomalies. These false positives (FPs) come primarily from outlier segments that are related to a manual SCA maneuvers or maintenance work. This is because the input data for the anomaly detection model does not provide information about manual operations. Fitting the clustering model with a larger dataset could decrease the FP rate. The FP segments occur more frequently in a larger dataset and will therefore be assigned to the “normal” cluster.

One detected outlier segment is regarded as actual anomaly. Hence, it is a true positive (TP).

The actual number of anomalies in the underlying data is unknown, therefore it cannot be made any statement about the number of actual anomalies that are not found. Hence, the number of false negative (FNs) is unknown.

Examples of detected outlier segments are evaluated manually and discussed in the following sections. For each example the possible reasons for the assignment as anomaly, the possible underlying cause (failure or operational behavior), and the frequency of similar detection or non-detection are discussed. Possible underlying causes can be identified with the command codes of the SCAs that are available in the PTPP data. It provides insights to the commands sent to the SCAs.

7.6.1. Manual SCA maneuver

The detected anomalies in Figure 7.22 between 15.00 pm and 16.00 pm show a high SCA angle deviation. The reason for the high SCA angle deviation could be an intended manual maneuver of the SCAs due to the cleaning of the mirrors, an overload of the power block or a failure of the focusing factor control or SCA actuators. In all viewed anomalies

of this kind the SCAs have been controlled manually (due to send command). The manual maneuver occurs frequently, because of the regular cleaning of the mirrors or operation in manual mode. For the anomaly detection it is seen as rare event because the features DNI, mass flow and loop inlet temperature have a random condition during manual maneuvers. This results in a high FP rate.

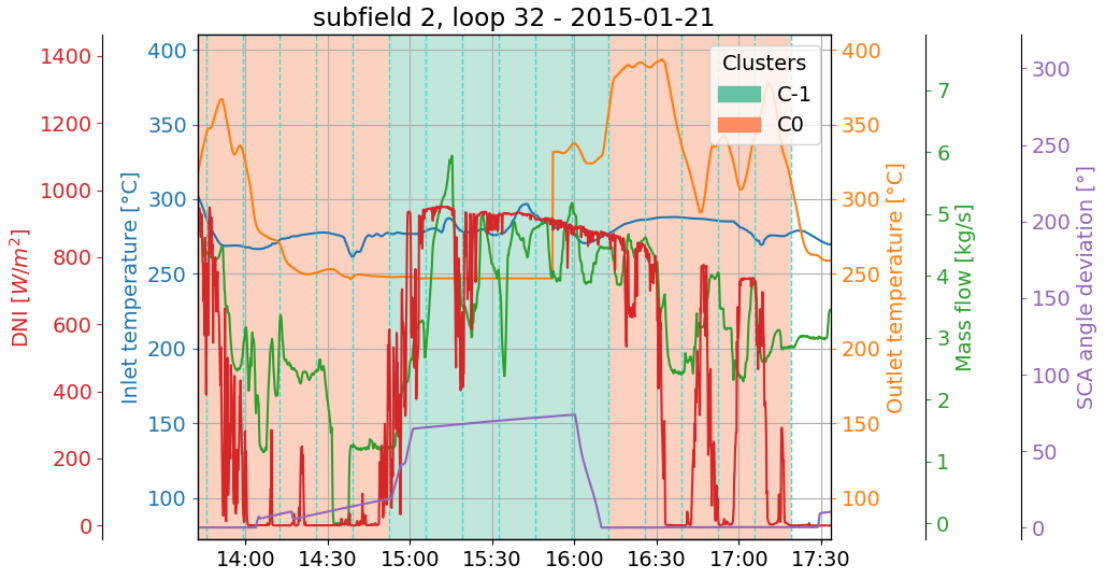

Figure 7.22: Example 1 of a high SCA angle deviation

The detection of a sudden significant increase or decrease of the SCA angle deviation is fairly robust, whereas a constantly high SCA angle deviation is not always detected as shown in Figure 7.23 between 14:55 pm and 15:50 pm.

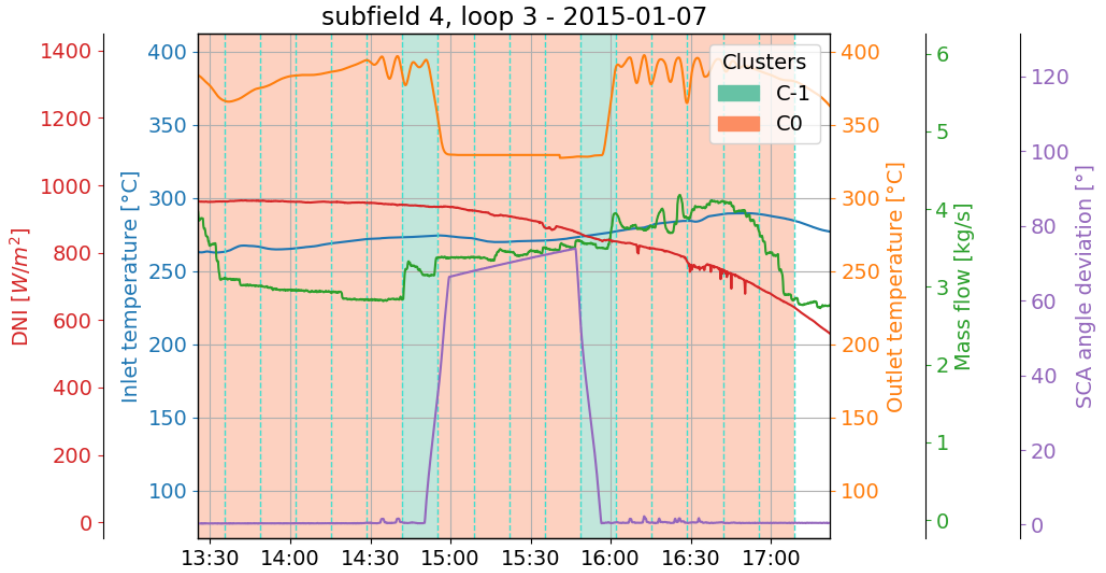

Figure 7.23: Example 2 of a high SCA angle deviation

Manual maneuvers are done for good reasons and are unavoidable for the operation of the PTPP. A failure of the tracking system would unnecessarily lead to a lower solar field output of the thermal power. A failure leads to a similar behavior of the data as a manual maneuver. Therefore, the distinction between a manual maneuver and a failure without considering the control commands is hard for the anomaly detection.

7.6.2. Maintenance work

In the example in Figure 7.24 the entire trimmed time series is considered as outliers. This is because of implausible values of the loop outlet temperature. The values of the temperature are between 0°C and 5°C. This behavior occurs on this day only in this loop. The reason could be a failure of the temperature sensor, a failure of the recording system or maintenance work. These assumptions are additionally verified by the consideration of the collector's temperatures within this loop. The temperatures of all collectors of the loop also show low values between 0° and 85°. This is a strong indicator that the behavior was caused by maintenance work, so that the normal operation of the loop was disabled and maybe temperature sensors were dismounted.

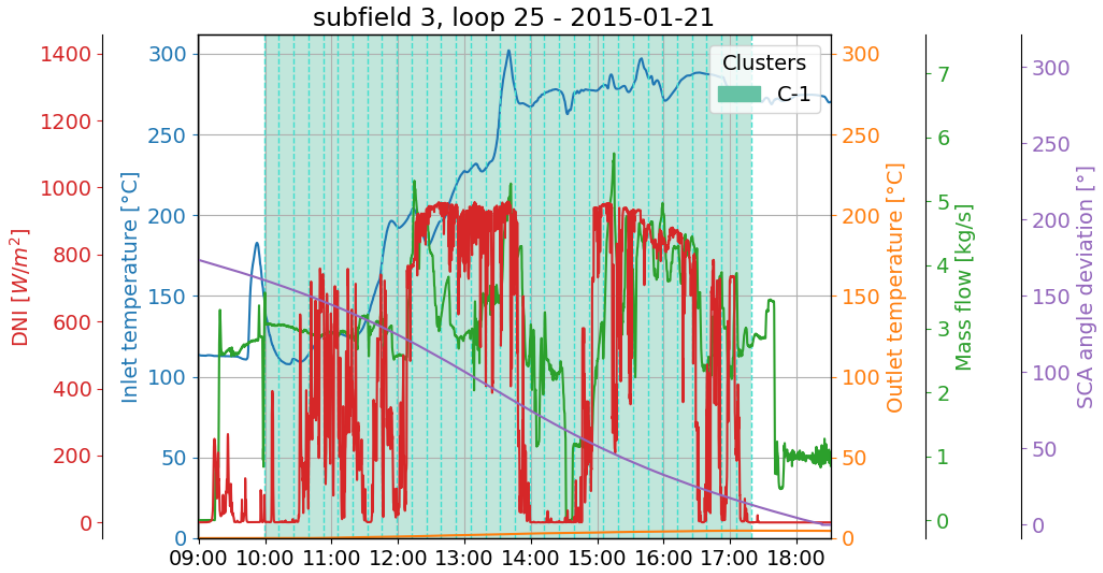

Figure 7.24: Example - low outlet temperature.

7.6.3. Other reasons

In the example in Figure 7.25 the segment is probably taken as outlier because of the contradictory behavior of the DNI and mass flow. The DNI decreases for a short moment. Normally the mass flow should also be decreased to hold the outlet temperature at a certain level, but instead the mass flow is increased. This leads to a reduced outlet temperature. No other loop outlet temperature of the subfield is close to the critical

temperature, which would excuse the increase of the flow. This example might be an actual anomaly and is considered as TP. This kind of behavior also occurs in the loops of the same subfield on the same day. But these segments are not assigned as outlier segments. These segments can be considered as FNs.

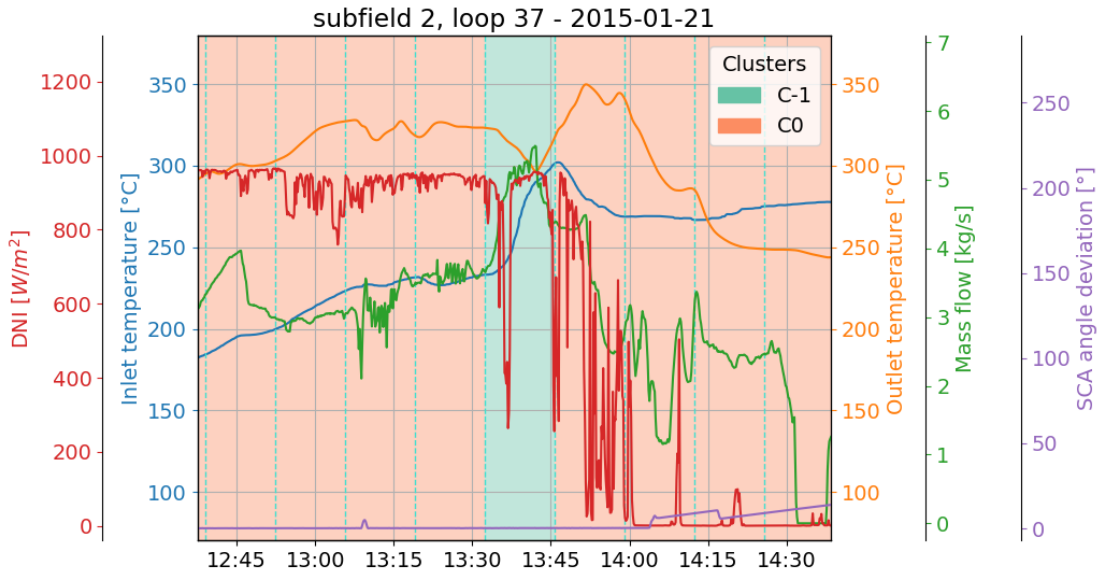

Figure 7.25: Example - significant increase of inlet temperature.

In the example in Figure 7.26 the segment is probably an outlier segment because of the significantly increasing outlet temperature. The temperature rises from 230° to 390°. The DNI also increases very strong. Hence, the behavior is not actually anomalous. The same behavior is detected at the same day in 5 other loops.

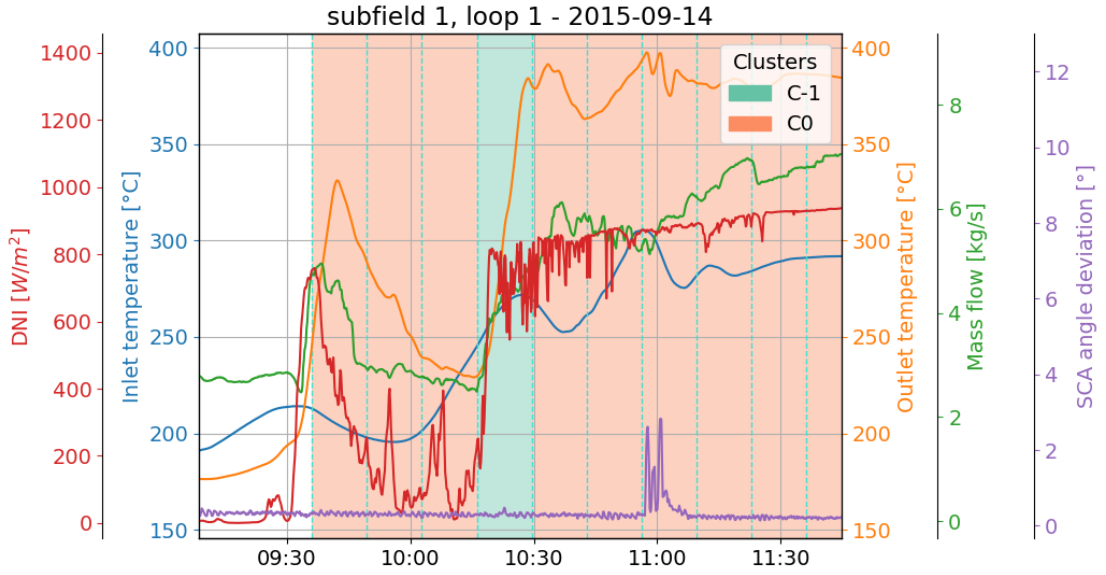

Figure 7.26: Example - significant increase of outlet temperature.

7.6.4. Reason not clarified

The example in Figure 7.27 shows an outlier segment, where the reason cannot be clarified. The SCA angle deviation is nearly zero. The DNI is increasing. This leads to an increase of the loop outlet temperature. The mass flow is increasing, which moderates the increase of the outlet temperature. The inlet temperature is nearly constant and starts to increase at the end of the segment. The behavior is not considered as anomalous. Seven segments with the same behavior that occur at the same day in other loops in the same time range from 9:50 am until 10:05 am are detected as outlier segments.

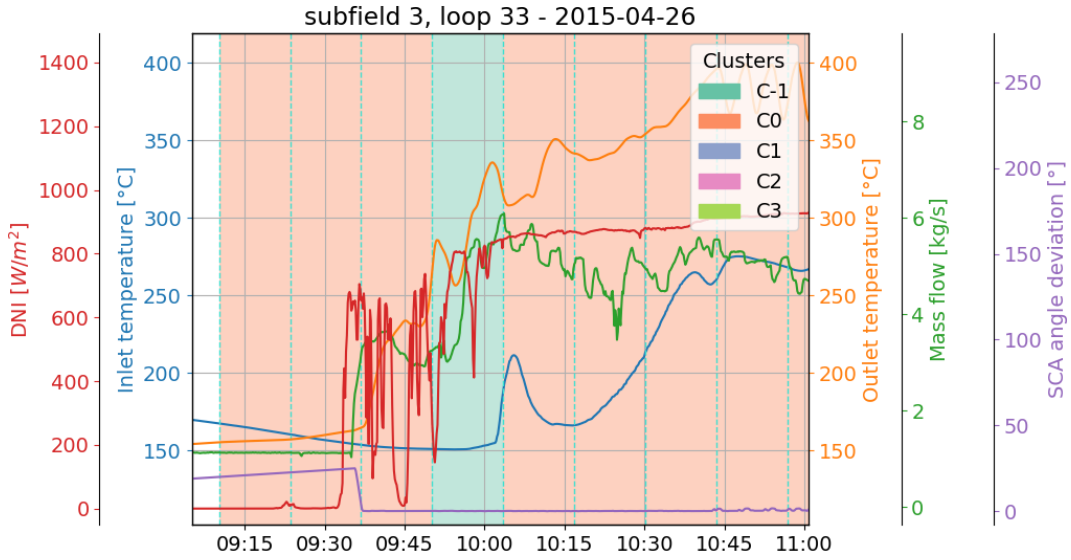

Figure 7.27: Example – reason not clarified.

Also, in the outlier segment in Figure 7.28 no reason can be found, for what this segment is considered as outlier.

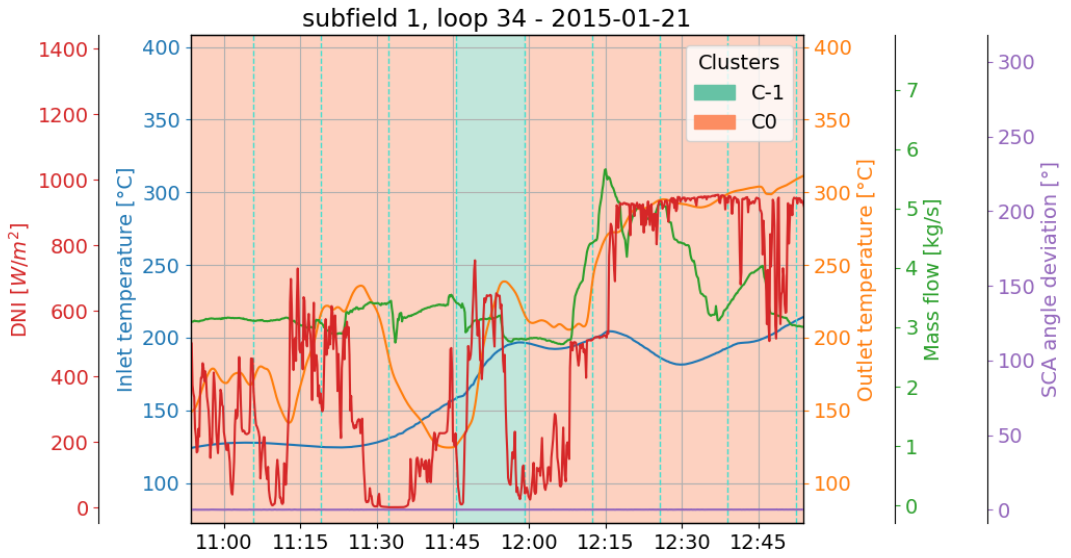

Figure 7.28: Example 2- reason not clarified.

It does not show any anomalous behavior. The DNI is increasing from $0\,W/m^2$ up to $800\,W/m^2$ and the mass flow is slightly decreasing, which explains the increase of the loop outlet temperature. As expected, the mean of the outlet temperature is slightly higher than the mean of the inlet temperature. The same behavior occurs at the same day in two other loops at the same time from 11:45 am until 12:00 am.

These examples of outlier segments where no obvious reason could be found can be used as references for a clustering with a different setting. Thus, various clustering's could be compared to each other in a better way.

7.6.5. False negatives

In the example in Figure 7.29 an unwanted characteristic of the control of the loop outlet temperature or SCA actuator is shown. However, this leads to a fluctuation of the SCA angle deviation and therefore to a fluctuation of the loop outlet temperature.

The clustering should be able to consider a fluctuating loop outlet temperature and SCA angle deviation at a nearly constant DNI, mass flow and loop inlet temperature. This kind of behavior is known by the operands and occurred very frequent in the past. Because of the frequent occurrence, the clustering gets many segments with the same behavior. With other hyperparameter for the clustering these segments probably could be assigned to an own cluster and do not fall into the main cluster. For the anomaly detection, there have probably been too many segments with this behavior to detect it as anomaly.

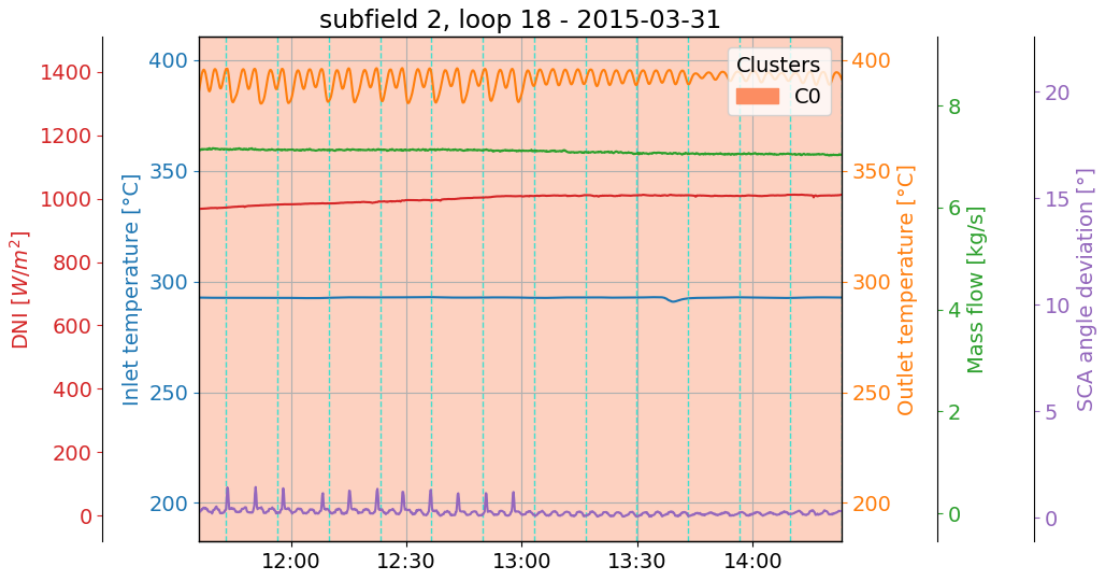

Figure 7.29: Example – Fluctuating loop outlet temperature.

7.7. Conclusion on approach

The analysis of detected outlier segments as it is done in section 7.6 shows that it is fairly hard to directly deduce anomalies to possible failures of the solar field as described in chapter 4. Classifying anomalies would need a ground truth to make use of more powerful supervised learning methods.

The feature selection, the time series segmentation and the feature extraction have a strong impact on the clustering results of the implemented approach.

The segmentation defines the number of points for the clustering and the duration of the individual segments. With the periodic segmentation either long homogeneous sequences are divided in few segments or heterogenous sequences are put in one segment. The fitting of one homogeneous sequence to one segment just happens by chance. Divided homogeneous sequences could lead to a high number of FNs as shown in one of the examples in section 7.6. A heterogenous segment leads to distortion of the extracted features.

The feature extraction with the mean and standard deviation of each dimension depends on the normalization or standardization of the features. That in turn, depends on the distribution of features. As shown in section 7.2 the distribution depends on the trimming of the time series. Therefore, the trimming and standardization has significant impact on the feature extraction. The dependency on the data scaling and trimming should be avoided.

Furthermore, the feature selection has a high impact on the results. The DNI value for the loops is extracted from the weather stations. The real values can be very different from the estimated values. Because of the huge solar field area, it is likely that clouds shade SCAs without shading any of the weather stations or the other way around. Additionally, the features mass flow and loop inlet temperature are estimated from the subfield inlet. This involves an inaccuracy of the features, what makes it more difficult or impossible to recognize relations between these and other features.

The SCA angle deviation has a large range between $0°$ and $180°$. The cleaning of mirrors often leads to high values for the SCA angle deviation. This makes it less sensitive to the normal operation mode where the range is around $-3°$ to $3°$. These small changes of the SCA angle deviation have a strong impact on the focusing range like shown in section 2.2.4. But the consideration of the entire range up to $180°$ let changes in the range of normal operation erroneously look small and insignificant.

Considering a larger dataset with PTPP data of one year or more could lead to better results. With the hardware provided for this research it is only possible to consider a small dataset with 24 days. A larger dataset would lead to a very high number of

segments or points that need to be clustered due to small segments, a huge number of loops and days. With a time complexity of $O(n^2)$ the DBSCAN clustering algorithm might be unsuitable.

In general, the clustering classifies a segment under consideration of all other segments. Observing one segment in the feature space, only the distances to all other segments are seen. The clustering does not consider the temporal or spatial context. The behavior of previous or following segments or of segments of a nearby loop is not considered for the clustering of one segment. The connection between temporally and spatially contiguous sensor data could play a major role in anomaly detection. Especially the large time delay between the inlet and outlet temperature due to the lengthy heating process in the receiver requires an approach that can interpret the correlation between these two features. The implemented approach cannot connect the increase of the outlet temperature in one segment with the increase of the inlet temperature in a previous segment of that loop. Furthermore, every loop outlet temperature of one subfield effects the mass flow due to the control of the loop outlet temperature. If the model sees only the data of one loop it cannot know whether a decrease of the mass flow is reasonable or not.

Because of the mentioned weaknesses of the clustering of segments it is deemed as incapable for the anomaly detection of PTPPs. Nevertheless, the approach could be good for clustering certain operation conditions or states of the solar field.

8. Summary and outlook

This chapter gives a summary of the work of this research. Furthermore, an outlook for future works is given.

8.1. Summary

In this research four approaches for the anomaly detection of the spatio-temporal data of PTPPs is evolved. The approaches differ on the underlying techniques of AI. In the different approaches the methodology of anomaly detection is with the help of an efficiency model, the clustering of time series, an autoencoder and a recurrent autoencoder. The *efficiency model* approach provides a simple outlier detection model and considers the temporal aspect of the data. The *clustering of time series* approach also considers the temporal aspect, but does not need knowledge about the physical processes or parameters of the PTPP. The *autoencoder* approach considers the temporal or spatial aspect of the data, whereas the *recurrent autoencoder* approach considers both aspects in one model.

In this research the *clustering of time series* approach is implemented. In advance, a concept for efficient organization of the existing measuring data is developed and implemented. It includes the read-in of the huge databases in python and saving it in another format for faster access. Additionally, the data is restructured for easier access and creation of datasets with two kinds of data instances. Furthermore, a GUI for the visualization of the solar field data and results of methods like segmentation and clustering is implemented.

For the input of the anomaly detection model the organized data is further preprocessed by extracting features as there are: the spatially interpolated DNI, mean SCA angle deviation and estimated mass flow of loop.

The *clustering of time series* approach includes the segmentation of time series, the feature extraction of the resulting time series segments, and finally the clustering in the feature space. This approach is tested with real data from a representative reference PTPP. Due to the provided hardware, only a small dataset of 1000 randomly picked loop samples of 24 different days covering one year is used.

After the analysis of the feature distribution and before segmentation, the input multivariate time series is trimmed. Hence, the data is reduced to the normal operation at daytime. For the segmentation, three different methods are evaluated: PCA-based, window-sliding and periodic segmentation. For the evaluation, a segmentation score is developed which evaluates the placement of the breakpoints and the number of segments. With the help of the evaluation score it turned out that a periodic segmentation fits best

for the given data. For the feature extraction, the mean and the standard deviation for each dimension of the segments is used. The clustering is done with the DBSCAN algorithm. Its hyperparameters are narrowed by limiting the number of clusters and the outlier point ration. Resulting outlier segments are analyzed manually and categorized to its possible reasons. The implemented anomaly detection model resulted in a high ration of outlier segments which causes are related to manual SCA maneuver or maintenance work. The detection of outlier segments due to other reasons is low, but without further information about failures of the power plant no statements about the FN and the TP rate can be made.

8.2. Outlook

In this section follow-up work is recommended for the anomaly detection of PTPPs.

In general, the SCA angle deviation could be limited to a smaller range (e.g. $0°$ to $3°$) to overcome the large values of the SCA angle deviation of up to $180°$. Either way, for values above $3°$ the focusing factor becomes zero so that the SCA angle deviation does not have any effect on the loop outlet temperature. The spatial interpolated DNI could be improved by developing a separate model for the passage of clouds considering solar field data.

Despite the weaknesses of the *clustering of time series* approach several improvements could be done. At first, sequences where the loop was in manual mode could be filtered out by considering the control commands. This would decrease the FP rate significantly.

Furthermore, the estimated loop inlet temperature could be replaced by the temperature of the first collector in each loop. This temperature is probably closer to the real loop inlet temperature than the estimated loop inlet temperature from the subfield. Either way, an exact loop inlet temperature is not necessarily needed. It just needs a condition close to the inlet and outlet so that the model can classify the process between these two points as normal or anomalous.

Additionally, other features than mean and standard deviation could be extracted from the time series segments. A further analysis is needed to decide if there are other extractable features which are more meaningful in the context of anomaly detection. Possible features could be the number of peaks, the entropy or the autocorrelation.

Moreover, the anomaly detection needs to be applied on a bigger dataset. Therefore, the model should run on a GPU, which is much more powerful than the used CPU. A larger dataset promises more variety in the normal data and decreases the number of FPs. In addition, another method could be used for identifying outlier segments. For example, the local outlier factor (LOF) could be used, which has a lower time complexity.

It would also be interesting to implement another evolved approach for the anomaly detection. Especially the recurrent autoencoder is a promising approach, because it considers the spatial and temporal information of the data in one model. This takes important spatial and temporal relationships greater into account.

References

[1] K. Conley, *Solar Energy*. Minneapolis, MN, United States: ABDO Publishing Company, 2016.

[2] K. Lovegrove and W. Stein, *Concentrating Solar Power Technology : Principles, Developments and Applications*. Cambridge, United Kingdom: Elsevier Science & Technology, 2012.

[3] S. Qazi, *Standalone Photovoltaic (PV) Systems for Disaster Relief and Remote Areas*. Saint Louis, United States: Elsevier, 2016.

[4] T. Hirsch, J. Dersch, and S. Giuliano, "Draft for an Appendix C - Solar Field Modeling," 2017. SolarPACES Guideline for Bankable STE Yield Assessment

[5] A. G.-C. Martelo. (2015, 11.12.2020). *What is a parabolic trough collector?* Available: http://www.theenergyofchange.com/parabolic-trough-collector

[6] J. Jentjens, "Untersuchung der Potentiale einer Onlineüberwachung solarthermischer Kraftwerke," Dipl. Maschinenbau Diplomarbeit, Maschinenwesen, Technische Universität Dresden, 2009.

[7] K. Noureldin, "Modelling and Control of Transients in Parabolic Trough Power Plants with Single-Phase Heat Transfer Fluids," Doctoral thesis, Fakultät für Maschinenwesen, Technischen Hochschule Aachen, 2018.

[8] C. C. Aggarwal, *Outlier Analysis*. New York, NY, United States: Springer New York, 2013.

[9] K. Singh and S. Upadhyaya, "Outlier Detection: Applications And Techniques," *International Journal of Computer Science Issues,* vol. 9, 01.01 2012.

[10] I. Cohen. (2018, 11.11.2020). *Outliers analysis: a quick guide to the different types of outliers*. Available: https://towardsdatascience.com/outliers-analysis-a-quick-guide-to-the-different-types-of-outliers-e41de37e6bf6

[11] C. C. Aggarwal, *Neural Networks and Deep Learning*. Springer International Publishing AG, 2018.

[12] L. Strika. (2019, 07.11.2020). *Convolutional Neural Networks: A Python Tutorial Using TensorFlow and Keras*. Available:

https://www.kdnuggets.com/2019/07/convolutional-neural-networks-python-tutorial-tensorflow-keras.html

[13] S. Saha. (2018, 07.11.2020). *A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way*. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[14] P. T. Perez. (2018, 07.11.2020). *Deep Learning: Recurrent Neural Networks*. Available: https://medium.com/deeplearningbrasilia/deep-learning-recurrent-neural-networks-f9482a24d010

[15] K. P. Murphy, *Machine Learning : A Probabilistic Perspective*. Cambridge, United States: MIT Press, 2012.

[16] C. C. Aggarwal and C. K. Reddy, *Data Clustering : Algorithms and Applications*. Philadelphia, PA, United States: CRC Press LLC, 2013.

[17] Chire. (07.11.2020). *DBSCAN*. Available: https://de.wikipedia.org/wiki/DBSCAN#/media/Datei:DBSCAN-Illustration.svg

[18] Z. Jaadi. (09.12.2020). *A Step-by-Step Explanation of Principal Component Analysis*. Available: https://builtin.com/data-science/step-step-explanation-principal-component-analysis

[19] A. Brenner, "FMEA parabolic trough fields," Unpublished work, Deutsches Zentrum für Luft- und Raumfahrt (DLR), 2020.

[20] G. E. Cohen, D. W. Kearney, and G. J. Kolb, "Final Report on the Operation and Maintenance Improvement Program for Concentrating Solar Power Plants," ed: Sandia National Laboratories (SNL), 1999.

[21] G. Atluri, A. Karpatne, and V. Kumar, "Spatio-Temporal Data Mining: A Survey of Problems and Methods," *ACM Comput. Surv.,* vol. 51, p. Article 83, 2018.

[22] A. C. do amaral Burghi, "Transient Simulation of Line-Focus Solar Thermal Power Plants," Masterarbeit, Linienfokusierende Systeme, DLR Institut für Solarforschung, 2016.

[23] E. Bingham, A. Gionis, N. Haiminen, H. Hiisilä, H. Mannila, and E. Terzi, "Segmentation and dimensionality reduction," presented at the SIAM International Conference on Data Mining, 2006.

[24] J. Hartung, G. Guehring, V. Licht, and A. Warta, "Comparing multidimensional sensor data from vehicle fleets with methods of sequential data mining," *SN Applied Sciences,* vol. 2, 19.03. 2020.

[25] E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration. Data Mining and Knowledge Discovery 7(4), 349-371," *Data Mining and Knowledge Discovery,* vol. 7, pp. 349-371, 10.01. 2003.

[26] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing,* vol. 167, pp. 107-299, 01.02. 2020.

[27] B. Fulcher and N. Jones, "Highly Comparative Feature-Based Time-Series Classification," *IEEE Transactions on Knowledge and Data Engineering,* vol. 26, 15.01. 2014.

[28] Andrewngai. (2020, 02.12.2020). *Understanding DBSCAN Algorithm and Implementation from Scratch*. Available: https://towardsdatascience.com/understanding-dbscan-algorithm-and-implementation-from-scratch-c256289479c5

A Approach for efficiency model

This section shows an approach for the efficiency model, which could be used for the corresponding anomaly detection approach described in section 6.1.

The idea is to calculate the momentary thermal efficiency η_{therm} of a loop with

$$\eta_{therm} = \frac{\dot{Q}_{out}}{P_{in}} \tag{A.1}$$

Therefore, the heat flow $\dot{Q}_{out}$ and the incoming solar power P_{in} needs to be calculated.

The gained heat flow can be calculated with the mass flow $\dot{m}$, the heat capacity of the HTF c_p and the difference between the loop inlet temperature T_{in} and the loop outlet temperature T_{out}.

$$\dot{Q}_{out} = \dot{m} \cdot c_p \cdot (T_{out} - T_{in}) \tag{A.2}$$

The measurements of the loop inlet and loop outlet temperature takes place at different locations, at the inlet and the outlet of the loop. The duration needed by one HTF element (see Figure A.1) to flow from the inlet to the outlet through the entire loop can be fairly long. The duration Δt depends on the mass flow of the HTF and the geometry of the pipes: the lower the mass flow, the longer the duration. Even with a high mass flow at daytime and a total collector length of around 600 m the HTF needs about 5 min from the inlet to the outlet.

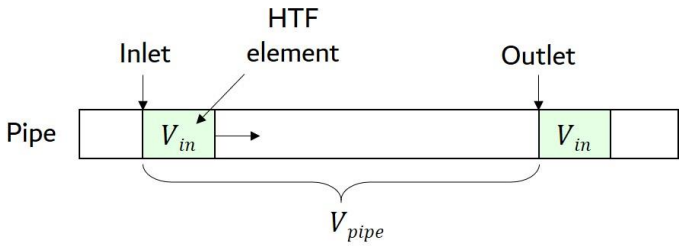

Figure A.1: Observed HTF element.

The outlet time point of the HTF t_{out} is calculated with the volume flow that occurs while the HTF element flows through the loop (see Figure A.2) by

$$\int_{t_{in}}^{t_{out}} \dot{V}\, dt = V_{pipe} \tag{A.3}$$

Whereby, t_{in} is the inlet time point of the HTF element. The algorithm successively calculates the integral of the volume flow beginning at the time point t_{in}. If the integral is equal to the volume of the pipe V_{pipe} the time point t_{out} is found.

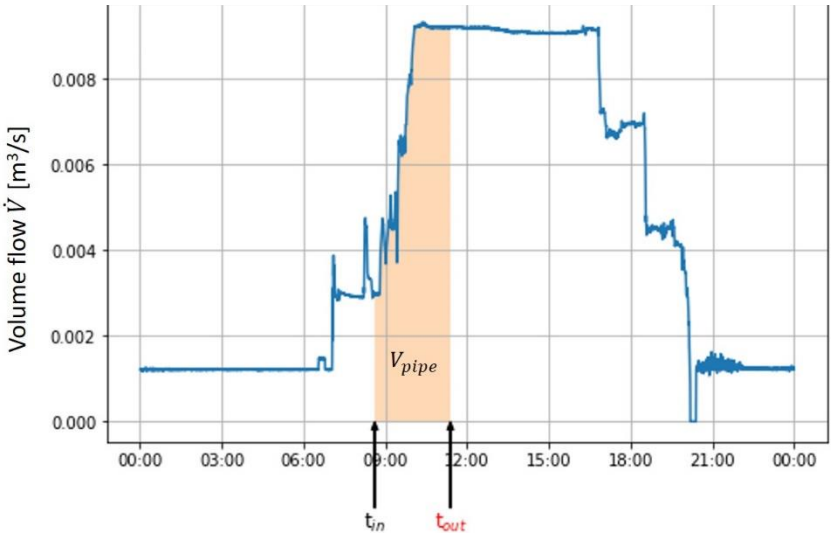

Figure A.2: Integral of volume flow.

With the computed outlet time point t_{out} a loop outlet temperature $T_{out,shift}$ can be determined, which corresponds to the same HTF element.

$$T_{out,shift} = T_{out}(t_{out}) \tag{A.4}$$

The heat flow is then calculated with

$$\dot{Q}_{out,shift} = \dot{m} \cdot c_p \cdot (T_{out,shift} - T_{in}) \tag{A.5}$$

where each value corresponds to the HTF element at the inlet time point t_{in}.

The incoming solar power is calculated by considering the geometrical losses (see section 2.3) and the focusing factor f_{focA} (see section 2.2.4):

$$P_{in} = G_{bn} \cdot A_{nom} \cdot \eta_{cos} \cdot \eta_{endloss} \cdot K \cdot f_{focA} \tag{A.6}$$

Where G_{bn} is the DNI, A_{nom} the nominal aperture area of one loop, η_{cos} the cosine loss, $\eta_{endloss}$ the end loss, K the IAM and f_{focA} the focusing factor.

While the HTF element flows through the loop the DNI changes. Therefore, the mean value of the DNI $\bar{G}_{bn}$ between the inlet time point and outlet time point is calculated so that

$$P_{in,shift} = \bar{G}_{bn} \cdot A_{nom} \cdot \eta_{cos} \cdot \eta_{endloss} \cdot K \cdot f_{focA} \tag{A.7}$$

Finally, the actual momentary thermal energy $\eta_{therm,shift}$ is calculated with

$$\eta_{therm,shift} = \frac{\dot{Q}_{out,shift}}{P_{in,shift}} \tag{A.8}$$

B Programming implementation and visualization

B.1 Programming language and used libraries

The implementation of the methods described in this research is completely programmed in Python 3.7.6. The Table 2.1 shows the most important libraries used for the implementation.

Table B.1: Used python libraries

Python libraries	**Usage**
pyodbc	Read in original Microsoft Access Databases of PTPP data
pickle	Saving data, meta data, and results as python objects
bz2	Compressing and decompressing of files
numpy	Data saving and manipulation in one- or multidimensional arrays.
pandas	Data saving and manipulation in DataFrames and Series (e.g. multidimensional time series or spatial map)
matplotlib	Plotting of figures
scipy	Apply Savitzgy-Golay filter on time series, Interpolation of time series
sklearn	PCA for segmentation and visualization, DBSCAN for clustering, NearestNeighbors for evaluation of DBSCAN clustering
ruptures	Apply window-sliding segmentation
itertools	Building permutation for evaluation with grid search
datetime	Manipulating dates and times
torch	Building dataset
tkinter	Building GUI for interactive visualization of solar field data (see B.2)

B.2 GUI for visualization (SolarView)

For the visualization of solar field data and results from methods like trimming, segmentation, normalization and clustering a graphical user interface (GUI) is created (see Figure B.1). The application is named *SolarView.*

SolarView accesses solar field data from a selected dataset. In the upper left section is a mapped solar field from the top view. Here a certain loop can be selected to show its time series at a certain day.

Depending on the methods of the selected dataset that has been applied before, the following things can be visualized:

- Interpolated time series data
- Period of a selected trimming
- Breakpoints of a selected segmentation
- Time series of normalized or standardized data
- Clustering in a scatter plot
- Clustering as colored segments in the time series
- Histogram of trimmed or non-trimmed feature time series

The applied methods can be selected at the sidebar on the right. Figure options can be made at the lower section of the sidebar.

If an applied clustering method is selected, the loops are colored depending on the clustering. The colors have the following meanings:

- Grey (not colored): Loop not considered for clustering
- Green: All segments of the loop are in the main cluster C0.
- Orange: At least one segment of the loop is in the minor cluster C1 – CN
- Red: At least one segment is clustered as outlier (Cluster C-1)

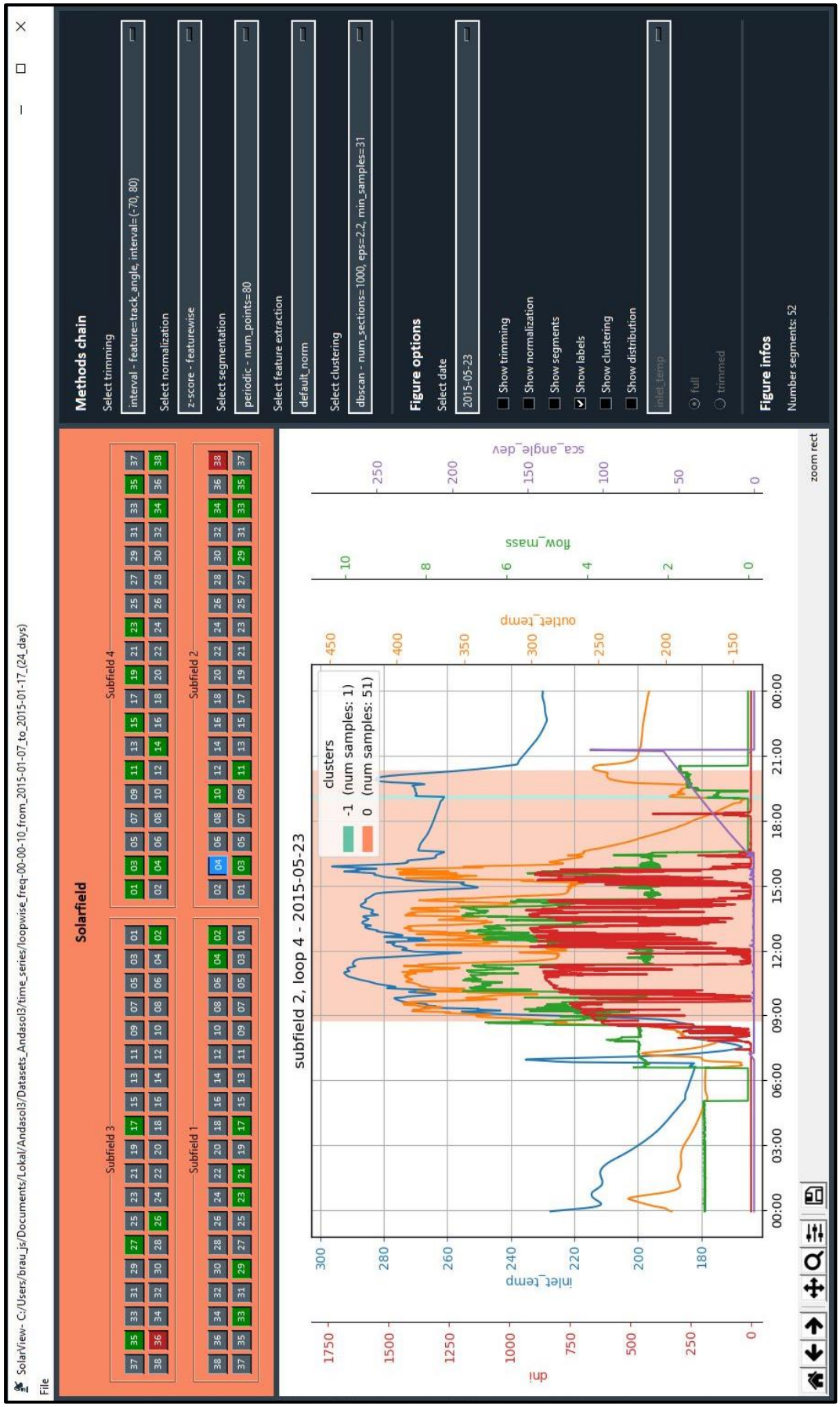

Figure B.1: GUI for visualization of solar field data.

Ehrenwörtliche Erklärung

Name: Josua Braun
Matrikel-Nr: 760121
Studiengang: Angewandte Informatik

Hiermit versichere ich, Josua Braun, dass ich die vorliegende Masterarbeit mit dem Titel „Anomaly detection for solar thermal parabolic trough power plants with artificial intelligence“ selbständig und ohne fremde Hilfe verfasst und keine anderen als die angegebene Literatur und Hilfsmittel verwendet habe. Die Stellen der Arbeit, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen wurden, sind in jedem Fall unter Angabe der Quelle kenntlich gemacht. Die Arbeit ist noch nicht veröffentlicht oder in anderer Form als Prüfungsleistung vorgelegt worden.

Ort, Datum

Unterschrift