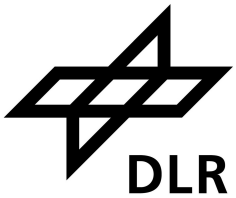


Department of Computer Science and Automation
Data-intensive Systems and Visualization Group

Master Thesis

Shapelet based clustering and anomaly detection for compromised, noisy and missing multivariate time series

Registration Date: 07. January 2023
Submission Date: 24. July 2023

Academic Chair: Prof. Dr.-Ing. Patrick Mäder
Academic Supervisor: M.Sc. Ferdinand Rewicki, M.Sc. Matthias Schmal

Submitted by: Tobias Merlin Fischer
born 29.07.1997
in Wickede, Deutschland
tobias-merlin.fischer@tu-ilmenau.de
Course of Studies: Wirtschaftsinformatik Master

Acknowledgments

I would like to thank my supervisors, my family and my girlfriend.

Abstract

Multivariate time series (MTS) data are widely prevalent in various domains, including medical screening, system monitoring or astronomy, where each instance consists of multiple sequences with inherent temporal ordering. Detecting anomalies in MTS is a critical research area aimed at identifying time points or patterns that deviate from normal behaviour. The demand for not only detecting anomalies but also processing them representatively, has led to the development of shapelet-based anomaly detection methods, which offer both interpretability and accuracy for MTS analysis. Despite the increasing interest in anomaly detection methods, shapelet-based anomaly detection remains a relatively small research field. This thesis introduces a novel workflow for unsupervised shapelet-based anomaly detection and anomaly prototype identification in MTS data, combining established anomaly detection methods with a shapelet-based classification framework. To evaluate different shapelet detection techniques based on clustering and validate the effectiveness of the workflow, experiments were conducted on both synthetic and real-world telemetry datasets of an exploration greenhouse. It was found, that the proposed workflow successfully detected diverse anomaly types while demonstrating the shapelet's interpretability, especially for the real-world case. Moreover, the potential to enhance anomaly detection appears to rely on the adapted weighting of cluster size and distance during shapelet selection and the issue of vanishing anomalies due to the Euclidean distance used for clustering.

Zusammenfassung

Multivariate Zeitreihendaten (MTS) sind in verschiedenen Bereichen weit verbreitet, beispielsweise in medizinischen Untersuchungen, der Systemüberwachung oder der Astronomie, wo jede Instanz aus mehreren Sequenzen mit inhärenter zeitlicher Ordnung besteht. Die Erkennung von Anomalien in MTS ist ein wichtiger Forschungsbereich, der darauf abzielt, Zeitpunkte oder Muster zu identifizieren, die vom normalen Verhalten abweichen. Die Anforderung, nicht nur Anomalien zu erkennen, sondern sie auch repräsentativ zu verarbeiten, hat zur Entwicklung von Shapelet-basierten Anomalie-Erkennungsmethoden geführt, die sowohl Interpretierbarkeit als auch Genauigkeit für die MTS-Analyse bieten. Trotz des zunehmenden Interesses an Methoden zur Erkennung von Anomalien ist die Shapelet-basierte Anomalieerkennung noch ein relativ kleines Forschungsgebiet. In dieser Arbeit wird ein neuartiger Arbeitsablauf für die unüberwachte Shapelet-basierte Anomalieerkennung und Identifizierung von Anomalie-Prototypen in MTS-Daten vorgestellt, unter Verwendung etablierter Anomalie-Erkennungsmethoden kombiniert mit einem Shapelet-basierten Klassifizierungsverfahren. Um die Identifizierung von Shapelets, basierend auf Clustering, zu bewerten und die Effektivität des Arbeitsablaufs zu validieren, wurden Experimente sowohl mit synthetischen, als auch mit Telemetriedatensätzen eines Forschungsgewächshauses durchgeführt. Es zeigte sich, dass der vorgeschlagene Arbeitsablauf verschiedene Anomalietypen erfolgreich erkannte und gleichzeitig die Interpretierbarkeit der Shapelets demonstrierte, insbesondere für die Realdaten. Darüber hinaus scheint das Potenzial zur Verbesserung der Anomalieerkennung von der angepassten Gewichtung zwischen Clustergröße und -distanz während der Shapelet-Auswahl und dem Problem der verschwindenden Anomalien aufgrund der für das Clustering verwendeten euklidischen Distanz abzuhängen.

Contents

1. Introduction

Multivariate time series (MTS) data are ubiquitous in many applications, for example medical screening [ZS23] or astronomy [LCX+21a], where each instance is associated with multiple series or sequences which have a natural temporal ordering. Multivariate time series analysis has become an important research topic, which aims to discover time-dependent characteristics or statistics [AAXJ21][ZMK12]. Time series anomaly detection is one form of multivariate time series analysis, aiming to find time points or patterns, that deviate from what is seen as normal behaviour [BKS+18, p. 947]. Anomaly detection has gained growing attention from academic research and industry, initiated by an explosion in the amount of data produced and the number of systems requiring monitoring [GCC+23, p.1]. There exist many anomaly detection methods, but the need for more interpretable anomaly detection resulted in the rise of explainable artificial intelligence and thus in the development of shapelet-based anomaly detection methods [SWP22] [CYPY21][BGCML21]. Time series shapelets are short, discriminative subsequences that have been found not only to be accurate, in terms of classification and clustering but also interpretable for multivariate time series analysis. Their comprehensibility is primarily demonstrated by their visual simplicity, direct applicability within time series for comparative purposes, and their role as prototypes for specific classes. [ZS23, YK09].

The EDEN NEXT GEN Project, supported by the Institute of Data Science at the German Aerospace Center (DLR), is exploring the design as well as the detailed analysis of all essential subsystems of an integrated Bio-regenerative life support system (BLSS) demonstrator [ZBV+15]. Considering that such systems are designed for highly autonomous operations management, their autarky exacerbates the usage of expensive expert knowledge for detecting anomalous system states and types of anomalous states in system control. In order to not only find anomalies and

anomaly types but also to capture them in a visually representative way, shapelets could be a useful approach, focussing on the detection of anomalies in environmental parameters and telemetric data, to counteract, if necessary, with minimal need of an expert [ZBV+15].

In recent years, the number of shapelet-based time series analysis methods steadily rose, providing a range of different techniques for shapelet selection and shapelet learning [ZS23, LCX+21a, ZMK12, GSWST14]. Shapelet-based anomaly detection otherwise is a relatively small field of research [BGCML21, p.26][SWP22][CYPY21]. Current methods can only detect whole time series as anomalous and still rely on heuristics for crucial assumptions or do not work fully unsupervised in other ways [AA22][BKS+18]. This work wants to reduce this gap and propose a novel workflow for real-world multivariate time series anomaly detection with little to no label information present. The contribution of this thesis consists of:

- a workflow for unsupervised shapelet-based anomaly detection and anomaly type identification on time point level
- a synthetic anomaly dataset generator, based on the EDEN ISS dataset, that allows full control over the distribution, number and type of anomalies to be investigated
- the application to a synthetic and the EDEN ISS dataset, to compare different designs based on machine and deep learning methods

The structure of this work is presented in the following. Chapter 2 provides an introduction to time series analysis basics, anomaly detection fundamentals and existing approaches in shapelet-based anomaly detection found in the relevant literature. After presenting the in this thesis proposed workflow in Chapter 3, Chapter 4 offers a comprehensive overview of the utilized methods and algorithms, along with a discussion of their respective advantages and disadvantages. It describes the integration of previous shapelet-based anomaly detection techniques with time series classification techniques for unsupervised anomaly detection and anomaly type identification. In Chapter 5 and Chapter 6, the proposed workflow undergoes testing in multiple realizations to evaluate its anomaly detection capabilities, while giving meaningful

shapelets for anomaly types. This evaluation is carried out on both the EDEN ISS dataset and a synthetic dataset. The synthetic dataset is generated using a custom time series generator specifically developed for this research, based on the EDEN ISS dataset. Chapter 7 comprises a thorough analysis of the experimental results and offers suggestions for future improvements before concluding with a summary and overall conclusion in Chapter 8.

2. Related Work & State of the Art

This Chapter will introduce the basic concepts of anomaly detection and shapelets that are relevant to this thesis as well as their application for multivariate time series anomaly detection in general. Section 2.1 provides necessary definitions and will give a short review of how anomaly detection integrates itself in time series analysis. It also highlights other important analysis tasks and used methods associated with current anomaly detection techniques. The subsequent section 2.2 gives an insight into interpretable unsupervised multivariate time series analysis using shapelets and shapelet detection and why they are preferable in contrast to other representation methods. State of the art shapelet-based anomaly detection is presented in section 2.3 showing how shapelets can be used for finding anomalies and outlining what the current limitations are. Finally, a state of the art shapelets based classification method is explained in section 2.4 that can be combined with current shapelet-based anomaly detection for unsupervised shapelet-based anomaly detection and anomaly type identification for multivariate time series.

2.1. Time series analysis

Time series analysis is the task of utilizing methods to understand or model the underlying statistics and other characteristics of time-dependent data [CC08, p. 1]. The definition of a time series is based on [SWP22, p. 1780][NIMK20, p. 1191]:

Definition 2.1.1. *A time series X of length n is an ordered set of n real-valued data points $X = \{x_1, x_2, ...x_n\}$ with data point $x_t \in \mathbb{R}^d$, $1 \leq t \leq n$ and $t \in \mathbb{N}$. For $d = 1$, X is called univariate, for $d \geq 2$, X is referred to as multivariate.*

For certain time series characteristics, for example, stationarity, time series are usually not analyzed as a whole but on subsequence level [RDN23, p. 2][NIMK20, p. 1191]:

Definition 2.1.2. *A time series subsequence* $X_{i,j} = \{x_i, ..., x_j\} \subseteq X; i < j$ *is a contiguous segment of* X*, starting from position* i *with length* $\mid X_{i,j} \mid = j - i + 1$.

Time series analysis can be divided into different areas, depending on the research purpose: subsequence matching, anomaly detection, motif discovery, indexing, clustering, classification, visualization, segmentation, pattern identification, trend analysis, summarization and forecasting [ASW15, p. 1-2]. Aspects of time series analysis relevant to this work are classification, clustering and anomaly detection. First, time series classification and clustering are discussed since anomaly detection is closely related to them and many detection algorithms use classification or clustering techniques internally [SWP22, p. 1781].

2.1.1. Time series classification

Classification aims to specify which of C classes prior unknown input data falls into by learning a mapping $f : \mathbb{R}^n \to \{1, \ldots, C\}$ from a given training dataset. [GBC16, p. 97]. Numerous time series classification applications exist, for example, gesture recognition, finance, multimedia or electroencephalogram (EEG) [BB17, p. 1]. The problem definition for the task of time series classification is taken from [FFW+19, p. 921]:

Definition 2.1.3. *A given dataset* $\mathcal{D}$ *consists of tuples* (X_i, Y_i)*, where* $X_i \subseteq \mathbb{R}^d$ *represents a univariate or multivariate time series and* $Y_i \in \{0, 1\}$ *is the corresponding label vector. In the case of a dataset with* C *classes, the label vector* Y_i *is a binary vector of length* C*. Each element* $j \in [1, C]$ *in* Y_i *is set to 1 if* X_i *belongs to class* j*, and 0 otherwise. The goal of Time Series Classification (TSC) is to train a classifier on dataset* $\mathcal{D}$ *to learn a mapping* $f : X \to Y$ *from the input space to a probability distribution over the class labels. This classifier aims to predict the class label for new, unseen time series data.*

By definition, classification falls into the category of supervised tasks. Existing methods for classification can be grouped into five main categories, depending on what they are based on [RFL+20, p. 403]. In the following, the focus is mainly on categories that also play a role in shapelet-based anomaly detection.

distance based methods A widely used approach in time series classification (TSC) involves employing a 1-nearest neighbour classifier along with a customized distance function that addresses potential offset issues by allowing some realignment of the series. It aims to classify time series by comparing their distance values, where smaller distances indicate greater similarity and larger distances indicate greater dissimilarity. Dynamic Time Warping is the favoured distance function for this purpose [RFL+20, p. 403f.].

Dynamic Time Warping (DTW) [SC07, p. 2f.] is a similarity measure for temporal sequence comparison. It aligns sequences by minimizing the total distance between corresponding elements. Given time series X of length n and Y of length m, the distance between elements (i, j) is calculated using a distance function $dist(i, j)$. The cumulative distance matrix D is computed as:

$$D(0,0) = 0$$

$$D(i,0) = \infty;\ i = 1, ..., n$$

$$D(0,j) = \infty;\ j = 1, ..., m$$

$$D(i,j) = dist(i,j) + \min(D(i-1,j), D(i,j-1), D(i-1,j-1));\ i = 1, ..., n, j = 1, ..., m$$

The optimal alignment distance is $D(n, m)$, representing the minimum cumulative distance. The warping path is obtained by backtracking through the matrix D from (n, m) to $(0, 0)$, following the path with the lowest cumulative distances at each step. Each step corresponds to a pair of elements in the aligned sequences. DTW is versatile, accommodating variations in length, temporal distortions, and irregularities. It is commonly used in speech recognition, gesture recognition, time series analysis, and other applications involving temporal data. Recent studies have demonstrated

that DTW is often considered a challenging benchmark for time series classification. A conducted review by [BLB+16, p. 657] confirms the findings of its robust performance, with 7 out of 19 evaluated algorithms failing to surpass DTW.

feature based approaches In these methods, a high-level representation of a time series is constructed by using pre-determined features.

Definition 2.1.4. *Time series representation, also called time series transformation: "Given a time-series data* $X_i = \{x_1, \ldots, x_t, \ldots, x_n\}$*, representation is transforming the time-series to another dimensionality reduced vector* $X'_i = \{x'_1, \ldots, x'_m\}$ *where* $m \leq n$ *and if two series are similar in the original space, then their representations should be similar in the transformation space too."* [ASW15, p. 20]

The features can be examined in two categories. Local features are typically generated from subsequences of a time series while for global features the whole time series is used [AB21, p. 1491][SL17, p. 3]. For local features, each real-valued subsequence is discretized into a discrete *word*, a sequence of symbols over a fixed, pre-defined alphabet. The model builds a histogram, a feature vector, from word counts and finally utilizes a simple, e.g. linear, classifier on these feature vectors. Models that proceed in this way are called bag-of-patterns and are categorised by their discretization functions. [SL17, p. 3][RFL+20, p. 406f.] A commonly used bag-of-patterns algorithm is WEASEL+MUSE, a model utilizing the truncated Fourier transform and which showed to be competitive to even deep learning classification methods. [SL17, p. 8]

deep learning A comprehensive overview of deep learning-based time series classification is neither necessary nor within the scope at this point, the reader is referred to the relevant literature [GBC16, FFW+19]. Used deep learning methods and related basics are explained in chapter 4 to the required extent. At this point, it is sufficient to define deep learning methods as all methods, which are considered complex machine learning models [FFW+19, p. 921].

Definition 2.1.5. *Machine learning: "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."* [GBC16, p. 97]

A conducted deep learning review by [FFW+19] revealed, that deep learning methods based on so-called convolutional filters f for time series classification tend to yield better results on average compared to other methods. In the context of time series, a convolutional filter f of size $fs \leq |X|$ can be seen as a vector of weights $w \in \mathbb{R}$. The concept is to use filters as a transformation, by applying a striding dot product between the filter f and a given time series X, from position i in X:

$$\sum_{j=0}^{fs-1} X_{i+j} \times f_j \tag{2.1}$$

and utilizing the transformed features as input to another, simpler, mostly linear, classifier [DPW20, p. 4f.]. The learning procedure for convolutional filters [GBC16, p. 350ff.], as defined above, is done by a problem dependent performance measure P, the weight-dependent loss function $\mathcal{L}(w)$. The gradients of the loss function with respect to the weights are computed, specifically the partial derivative of the loss with respect to a specific filter weight w_{ij}, denoted as $\frac{\partial \mathcal{L}}{\partial w_{ij}}$. To optimize the loss function, an iterative process is employed w.r.t weights, the gradient descent:

$$w_{ij}^{(t+1)} = w_{ij}^{(t)} - \eta \cdot \frac{\partial \mathcal{L}}{\partial w_{ij}^{(t)}} \tag{2.2}$$

where $w_{ij}^{(t+1)}$ represents the updated weights at iteration $t+1$ and η is the learning rate. [1]

While traditional filter-based methods are extensively trained and optimized on different datasets, for the best possible results and generalisation abilities, [DPW20] demonstrated, that instead of learning individual weights, utilizing a multitude of

[1] For more information about the training in deep learning see: [GBC16]. The in this work used loss function for adopted deep learning methods can be found in section 4.1.3 and its derivation in A

random filters, which may only approximate relevant patterns individually, proves to be highly effective in capturing discriminative patterns in time series when used in combination.

By defining certain classes $C_i \subseteq C$ as anomalous, time series anomaly detection can be seen as a specialized field of time series classification, referring to the whole time series or subsequences of it. Because classification falls into the category of supervised learning methods, in the following section the unsupervised time series clustering will be explained, as a labeless way of class determination.

2.1.2. Time series clustering

Time series clustering is a time series analysis task, which aims to discover similar characteristics and statistics among multiple time series or subsequences and partition them into several subsets [ZS23, p. 1]. This section will introduce basic concepts that are relevant to this work.

Definition 2.1.6. *Time series clustering: "Given a dataset of n time-series data* $D = \{X_1, X_2, \ldots, X_n\}$*, time series clustering is the process of unsupervised partitioning of* D *into* $C = \{C_1, C_2, \ldots, C_k\}$ *such that homogenous time series are grouped based on a specific similarity measure. A cluster* C_i *is defined as a subset of* D *where* $D = \bigcup_{i=1}^{k} C_i$ *and* $C_i \cap C_j = \emptyset$ *for* $i \neq j$*."* [ASW15, p. 17]

Various types of clustering algorithms exist, the ones important for this work are partitioning and density-based methods. Since all methods used later in this thesis are explained in more depth in section 4, they are only discussed here to the extent necessary to understand anomaly detection.

partitioning methods A partitioning clustering technique aims to divide a set of n unlabeled objects into k distinct groups, ensuring that each group contains at least one object [AAXJ21, p. 8f.][ASW15, p. 26ff.]. The k-Means [AV07] algorithm is one of the commonly used methods for partitioning clustering, where each cluster

is represented by a centre that corresponds to the mean value of its constituent objects. The primary objective of k-Means clustering is to minimize the overall distance, often measured using Euclidean distance, between a set of n data points $X \subset \mathbb{R}^d$ within a cluster and their respective cluster centre c by iteratively choosing the k centres C:

$$\phi = \sum_{x \in X} \min_{c \in C} \|x - c\|^2 \tag{2.3}$$

The clustering is implicitly defined by the selection of these centres - for each centre, a cluster is designated as the set of data points that are closer to that centre than to any other centre. However, when applying k-Means to time-series clustering, the task becomes challenging and non-trivial.[AV07, p. 1028][ASW15, p. 28]

density based methods Density-based clustering is a clustering approach where clusters are formed as subspaces containing dense objects, with separation between clusters occurring in subspaces with objects having low density [AAXJ21, p. 9] [ASW15, p. 29]. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a well-known algorithm that operates based on the density-based concept. It expands clusters by considering the density of neighbours, allowing the cluster to grow in regions where points are densely packed together. Given a dataset $\mathcal{D}$ with data points $X = x_1, x_2, \ldots, x_n$ and a distance function $dist(x_i, x_j)$, a point x_j is said to be density-reachable from another point x_i if there exists a chain of data points $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ such that $x_j = x_{i_k}$ is reachable from $x_i = x_{i_1}$ and each successive point in the chain satisfies the density condition, i.e., $dist(x_{i_m}, x_{m+1}) \leq eps$ for a specified distance threshold eps and the minimum number of points $minpts$. Based on this density reachability definition, DBSCAN identifies clusters by connecting density-reachable points and considering their neighbourhood density. Points that are not density-reachable from any other point are considered noise or outliers. This method for time-series data offers several advantages, including its ability to detect clusters of arbitrary shapes and outliers, its parameterization based on spatial closeness, and its flexibility in not requiring a pre-defined number of clusters k. [EKSX96, p. 2f.][SSE+17][SEKX98]

By focusing on unusual and unexpected patterns in time series with a high distance regarding the similarity measure, time series clustering is closely related to time series anomaly detection [ASW15, p. 17].

2.1.3. Time series anomaly detection

Anomaly detection is the task of detecting observations that significantly deviate from what is considered normal. In terms of time series, anomalies can be from various sources, a out of rhythm heartbeat in healthcare or unexpected high bandwidth usage in cybersecurity [RDN23, p. 1f.].

The definition of anomalies is vague and varies with the use case, but these definitions have in common, that anomalies are rare events [GCC+23, p .1]. Commonly used general definitions describe an anomaly as a sequence of data points $X_{i,j}$ of length $j-i+1 \geq 1$, which deviates from frequent patterns in the time series X with respect to a characteristic embedding, model, or similarity measure [SWP22, p. 1780].

Definition 2.1.7. *Time series anomaly: "An anomaly deviates so much from the other observations as to arouse suspicion it was generated by a different mechanism"* [CYPY21, p. 120045] [BGCML21, p. 2].

Definitions for anomaly types are as manifold as anomaly definitions themselves. Different authors have tried to give a universal categorisation. Anomalies can be univariate or multivariate depending on whether they affect one or more time-dependent variable [LZX+21][CYPY21][BGCML21].

Definition 2.1.8. *Point outlier: A point outlier is a timestep that behaves unusually in a specific time instant when compared either to its neighbouring points (local outlier) or all other values in the time series (global outlier)* [BGCML21, p. 5][LZX+21, p. 3].

Definition 2.1.9. *Collective outlier: This type of anomaly refers to subsequences, which show a gradually different pattern over time, although each observation point individually is not necessarily a point outlier* [RDN23, p. 2] [LZX+21, p. 3][CYPY21, p. 120045] [BGCML21, p. 56].

Definition 2.1.10. *Contextual outlier: They represent an observed data point or subsequence that is anomalous considering the specific context e.g. a measured temperature value that can be considered normal in another environment, but corresponds to an anomaly in the observed environment.* [RDN23, p. 2][LZX+21, p. 3].

The variety of anomaly types, pattern models, and time series properties has given rise to a wide range of detection algorithms [SWP22, p. 1780]. These algorithms originate from different research areas and authors condensed them to different method families [SWP22, p. 1781]. An intuitive method for anomaly detection is forecasting a given time series and verifying if the predicted time steps match the original time series. These forecasting models are mostly trained in a way, that the training part of the dataset covers only normal class and is used to learn the normal model. A deviation from this expected, normal behaviour in the test part is regarded as anomalous. Some models learn the normal model directly from the raw, unknown dataset using some initial timepoints, the context window. For every context window, these models assume normality. The model is periodically rebuilt on different context windows, or the mean of a selection, to adapt to changes in the data. [SWP22, p. 1782-1783][CYPY21, p. 120050][BGCML21, p. 15]

Other methods rely more directly on data stochasticity [BRGD19, p. 2-3]. Distribution methods fit or estimate a distribution model to the data. Calculating distributions, in this algorithm family, can either be done over data points or subsequences, obtained through windowing. While the similarity of points and subsequences can affect the distribution fit and very similar patterns are counted as equal, abnormality is judged based on frequency rather than distance. Typically, anomaly scores are measured using probabilities, likelihoods or distances of the points or subsequences with respect to the prior calculated distributions. This approach is generally unsupervised under the assumption that anomalies are often found in the tails of the

distributions. However, in the semi-supervised case, the distribution is estimated over a training time series that only includes normal behaviour. Later, the points or subsequences of the test time series are examined against the previously learned distribution. [SWP22, p. 1784][BGCML21, p. 17]

As describing all employed methods in full depth would by far exceed the scope of this thesis, other method families, like reconstruction, encoding and isolation tree methods, will not be described and reference is made to the attached literature [SWP22] [CYPY21] [LZX+21][BGCML21].

shapelet-based anomaly detection, in general, falls into the category of distance-based methods. Subsequences displaying anomalous behaviour are anticipated to have larger distances to other subsequences than subsequences with normal behaviour. Algorithms in this family can use all other subsequences, only some nearest neighbours, or certain cluster centroids as distance reference points [BRGD19, p. 2]. Some methods also include a mapping of the subsequences into an appropriate representation space before computing the distances. Distance-based clustering methods cluster similar subsequences together and compute the distances to dense areas [CYPY21, p. 120051][van19]. Often, subsequences are generated using a sliding window with a stride of one on the test time series. Distance-based methods do not typically require training data and are thus unsupervised. [SWP22, p. 1784][LZX+21, p. 7][CYPY21, p. 120051]

This work draws upon a variety of algorithms that utilize distance-based methods: DBStream [HB16], k-Means [YKH01] and PhaseSpaceSVM (PS-SVM) [MP03]. DBStream and k-Means are nearest neighbour methods, calculating anomaly scores by computing the distance of points or subsequences to their nearest neighbours. Infrequent or unusual subsequences exhibit larger distances from their neighbours and are therefore considered anomalous. For k-Means, the anomaly score can also be computed by using the distances between subsequences and their corresponding cluster centroids as anomaly scores. PS-SVM fits a one-class support vector machine to a transformed representation of the subsequences and utilizes the inverse distance to the decision boundary as the anomaly score. A one-class support vector machine is a machine learning algorithm that separates normal data instances from outliers

by creating a hyperplane with the largest margin around the normal data. [SWP22, p. 1784].

A few anomaly detection approaches already relied on feature-based classification with some visual interpretability. The work of [HWL15] presents an approach, where they address the task of learning from an unlabeled set of time series that may contain anomalies. Their method involves extracting basic representative features such as mean and trend, as well as domain-specific features like the number of zero-crossings. Anomalies are detected by estimating the density of the feature transformed time series [BKS+18, p. 948]. However, these features still lack a visual component, whereby individual time series or subsequences can be visually assigned to a class or anomaly type. This task is performed by time series shapelets.

2.2. Time series shapelets

The in 2.1.1 introduced feature extraction is a form of dimension reduction which helps to lower the computational cost of dealing with high-dimensional data and achieve higher accuracy of classification or clustering. Mentioned discretization is often required for feature-based techniques, but can cause information loss in time series data. To address this [YK09] introduced time series shapelets, or short shapelets. Shapelets can be directly applied to time series. The similarity of time series sequences is measured based on comparing subsections of shapes, therefore the name shapelet. [AAXJ21, p. 7f.]

Definition 2.2.1. *"A shapelet S of length $|S|$ is a time series subsequence of class C_j, where $C_j \in C$, which represents class C_j and discriminates C_j from other classes i.e. $C \setminus \{C_j\}$. This accounts for all time series X_j having the label C_j which $dist(X_j, S)$ is smaller than $dist(X_k, S)$, where X_k is a time series having a label in $C \setminus \{C_j\}$"* [LCX+21b] [ZMK12, p. 3].

Shapelets are phase independent, meaning they are patterns within a time series that define a class, but the location of this pattern is irrelevant. For instance, an abnor-

mal ECG measurement can exhibit an uncommon pattern that sporadically appears at any given moment during the measurement. Shapelets are subseries that capture such distinctive characteristics, enabling the identification of phase-independent localized similarities among series belonging to the same class while giving a easy to interpret visualization [BLB+16, p. 622].

shapelet discovery Discovering shapelets methods can be divided into two groups, shapelet selection and shapelet learning[ZS23, p. 1]. Shapelet selection is closely related to prototype clustering, which is an essential subroutine in time-series clustering approaches. In partitioning clustering algorithms, such as k-Means, a cluster's prototype, denoted as P_j, minimizes the distance between all subsequences X_n within the cluster and the prototype. A subsequence P_j that minimizes the value of $E(C_i, R_j)$ is referred to as a Steiner sequence.

$$E(C_i, P_j) = \frac{1}{n} \sum_{h=1}^{n} \text{dist}(X_h, P_j); C_i \in \mathcal{X} = \{X_1, X_2, \ldots, X_n\} \tag{4.1}$$

In this definition, a cluster prototype and a shapelet can be seen as synonymous. [ASW15, p. 25]

In contrast to conventional shapelet selection methods, [GSWST14] proposed a shapelet learning approach. This is achieved by employing a heuristic gradient descent approach for a differential shapelet-based representation and employing a pseudo-classification objective function. The primary goal is to learn shapelets that enhance the linear separability of the time series data following the shapelet transformation. Notably, the shapelets learned through this method are not necessarily present in the original data [ZS23, p. 1]. Other methods utilize machine learning methods, like neural networks, for shapelet learning [MRDD21].

Since the first description, shapelets have already been implemented in time series analysis, e.g. classification, clustering and anomaly detection.

shapelet-based classification [YK09] introduced the original shapelet algorithm, which involves an exhaustive search of all potential candidates to find shapelets.

The algorithm selects the best shapelet as the splitting criterion at each node of a decision tree to determine, whether each new subsequence or time series belongs to a class or not. The shapelet decision tree classifier compares the distance between the testing object and the shapelet at each node. It recursively traverses the left subtree if the distance is below the split point or threshold, and the right subtree otherwise. This process continues until a leaf node is reached, providing the predicted class label. [BLB+16, p. 622] [AAXJ21, p. 7f.]

A more common approach was introduced by [BB17], who proposed a shapelet transformation approach that decouples shapelet discovery and classification. They identify the top f_S shapelets in a single run and use them to transform the data, assigning each attribute in the new dataset as the distance $dist(S, X)$ between a series X and one of the shapelets S. They further employ the k-nearest neighbours algorithm for determining the class of each representation [BLB+16, p. 623].

shapelet-based clustering In an unsupervised way, in shapelet-based clustering, the raw time series is transformed into a condensed shapelet-based space, similar to the transformation in classification, and the determination of time series clusters is achieved by applying the k-Means clustering algorithm to the shapelet-transformed representations. [ZS23, p. 1] [ZMK12]

Shapelets can provide intuitive, visually meaningful and interpretable results in time series analysis, particularly for the unsupervised case, helping domain practitioners better understand their data. Since they are local features, it was shown, that shapelets proved to be more accurate/robust in some classification tasks while also improving computation time, compared to the other methods investigated. [YK09, p. 2][ZMK12]

These advantages and the introduced extraction of shapelets from unlabeled data leveraged the use of shapelets for anomaly detection. However, using shapelets for anomaly detection has not gained much attention yet. [BGCML21, p. 55][BKS+18, 946][ZMK12]

2.3. Shapelet-based anomaly detection

Anomaly detecting methods, based on shapelets, fit seamlessly into the above described algorithm family of distance-based methods as a subcategory of methods, which aim to find more interpretable detection results [BGCML21, p. 55]. Instead of utilizing all generated subsequences, like described above, shapelets are short, discriminative and representative time series subsequences of one class [YK09, p. 1-3].

A common workflow observed in the literature incorporates elements of nearest neighbour methods, transformed representation, and occasionally a reference model of normal behaviour, as previously introduced. A supervised algorithm introduced by [AA22], for example, includes a mapping of the time series into a list of vectors, the so called orderline D_s, by calculating the squared Euclidean distance between shapelets $S_j \in \mathbf{S}$ and all time series subsets $X_i \in X$, both of fixed length l

$$dist(S_j, X_i) = \sum_{t=1}^{l} (s_t - x_t)^2 \tag{2.4}$$

and ranks the shapelets, utilizing the transformation, regarding their information gain. If a time series dataset X can be divided into two classes, labeled as C_1 and C_2, the entropy of X can be calculated as:

$$H(X) = -p(C_1)log(p(C_1)) - p(C_2)log(p(C_2)) \tag{2.5}$$

where $p(C_1)$ and $p(C_2)$ represent the proportions of objects in classes C_1 and C_2 respectively. The information gain [YK09, p. 3] of a split is determined by the difference between the entropy of the entire dataset and the weighted sum of average entropies for each split.

Definition 2.3.1. *Information gain: Given a time series dataset X, a shapelet S, and a distance threshold d_t, the dataset X is divided into two sub-datasets Xa and Xb. Each time series in Xa satisfies the condition $dist(Xa_i, S) < d_t$, while each*

series in Xb satisfies dist$(Xb_i, S) \geq d_t$. The information gain at each split point is calculated as [AK21, p. 4f.]*:*

$$IG = H(X) - \left(\frac{|Xa|}{|X|} H(Xa) + \frac{|Xb|}{|X|} H(Xb) \right) \tag{2.6}$$

The IG value ranges between $0 \leq IG \leq 1$. Based on the distance from every shapelet candidate to every time series, shapelets with more discriminative power will generate small distances when compared to a time series of its own class and therefore will have a higher information gain. By ranking and selecting the shapelets, computational effort can be minimized in contrast to utilizing all extracted subsequences. Based on the IG of each individual split, a classifier divides the labeled anomalous time series with their associated shapelets using the transformed data space, while maximizing summarized IG as target function. With the rise of deep learning and automated feature extraction, shapelet identification and classification was turned into an end-to-end deep learning task, for example using deep support vector data description (SVDD) by [ZZSG22]. [ZZSG22] tried to learn a shapelet-based time series representation, where all anomalous time series samples lie outside a given hypersphere of radius R and normal class within. Advantageous is that the proposed methods learn multi-scale length shapelets, instead of using fixed length shapelets, but still rely on supervised datasets while making the shapelet identification a black-box model.

[BKS+18] showed a similar to PS-SVM [MP03] approach and demonstrated how SVDD can be extended for unsupervised time series for fixed length shapelets, using shapelet learning [GSWST14]. With a time series transformation $\Phi(X; S_1, ..., S_{f_S})$, calculated similar to [AA22],with f_S being the number of final shapelets $\mathbf{S}$, a given decision boundary R, a regularization parameter C and slack variable θ, [BKS+18] define a solution to the constrained optimization problem:

$$arg \min_{R,\mathbf{S},\theta} F(R, \mathbf{S}) = R^2 + C \cdot \sum_{i=1}^{N} \theta_i + \sum_{i=1}^{N} ||\phi_i||^2 \tag{2.7}$$

$$s.t. ||\phi_i||^2 \leq R^2 + \theta_i,\ \theta_i \geq 0,\ \forall i = 1, ..., N$$

with $\phi_i = \Phi(X_i; S_1, ..., S_{f_S})$. Initially, all representation points ϕ_i lie within a scattered feature space. By learning best fitting normal class shapelets, normal class representation points are drawn towards the origin of the feature space, leaving anomalous time series as outliers. However, the proposed algorithm is only able to learn the normal class from the data and not anomalies or their types. Given the lack of existing literature supporting unsupervised anomaly detection methods capable of simultaneously identifying different types of anomalies and providing representative shapelets for both anomalous and normal classes, approaches have been drawn from the broader domains of shapelet-based time series classification and clustering.

2.4. ShapeNet

A lot of research in unsupervised time series clustering and classification exist [RFL+20][FFW+19][RPY+22][AAXJ21][yan18][ASW15], but the possibility of a modular workflow, which can combine the advantages of already known classification methods with the above mentioned anomaly detection algorithms, led to the final selection of the ShapeNet framework [LCX+21a]. ShapeNet was developed to provide more interpretability in the methods of end-to-end classification models. It consists of four individually configurable parts, namely shapelet discovery, shapelet selection, shapelet transformation and a final time series classification. The goal of shapelet discovery is the grouping of shapelets to find maximally representatives shapelets of a cluster in an unsupervised manner. The process of candidate creation, which serves as the starting point for shapelet discovery, bears similarities to the methods introduced in Section 2.3. Initial shapelet candidates are all time series subsequences of different lengths, found by using sliding windows of discrete sizes. Similar to, e.g. DBStream [HB16] or k-Means [YKH01], all shapelet candidates are clustered based on their distance to each other [LCX+21a, p. 8376f.].

Since shapelet discovery is unsupervised, clustered shapelets are sorted to obtain the final number of shapelets f_S, eliminating duplicates from dense, overlapping candidate clusters. At the same time, strong outliers must be preserved as probable representatives of rare classes [LCX+21a, p. 8378f.]. This is called the shapelet selection.

ShapeNet uses the already introduced time series transformation as a means of dimension reduction and time series representation [BB17][MP03]. Regions of the time series X that are close to a given shapelet S_{f_S} shall have small transformation values, simplifying the use of classification algorithms, assuming the shapelets are representative of a class. [LCX+21a, p. 8378f.]

Finally, the shapelet transformation is used by a simple, linear classification module learning the time series classes [LCX+21a, p. 8379]. It was shown, that the classification accuracy of the proposed ShapeNet framework is above the introduced DTW, WEASEL+MUSE or convolutional filter based methods while it is also capable of providing easy to understand, meaningful shapelets. In this work, the ShapeNet framework will be adapted to extend current anomaly detection and shapelet-based anomaly detection methods [LCX+21a].

3. Methodology

This chapter gives a short explanation about how state of the art shapelet based time series classification frameworks can be taken as inspiration, expanding the current anomaly detection algorithms to fit the defined goal of this thesis. The finalized workflow of this thesis is presented in this section. It is described how anomaly detection algorithms can be combined with the presented ShapeNet framework to obtain an easily configurable, interpretable unsupervised time series anomaly detection. For every part a short description is given and an explanation how the target of the part is embedded in the overall thesis target. After the overview has been given, the used algorithms are described in the next section.

The Workflow retains the described four-part modular structure, as seen in figure 3.1.

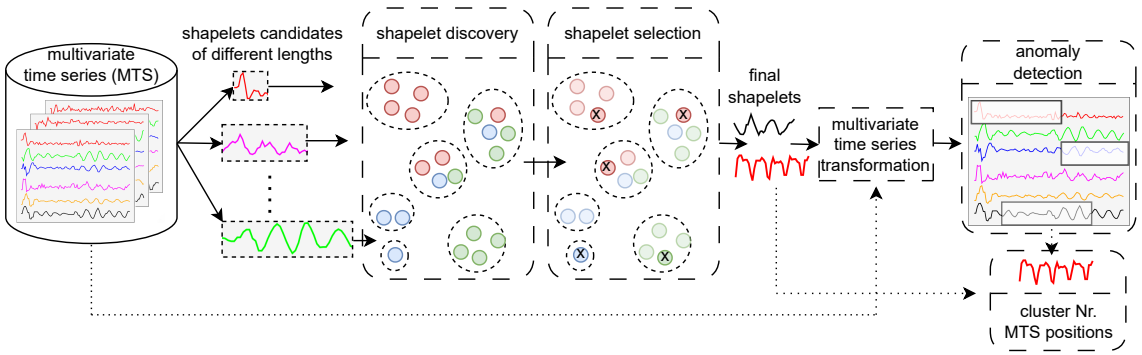


Figure 3.1. – The proposed workflow for unsupervised interpretable anomaly detection, in accordance with [LCX+21a]

Shapelet discovery initiates univariate shapelets as windowed subsequences of the time series and clusters them subsequently. By considering anomaly types as classes, clustering provides a grouping of all initial shapelets for further filtering. Since the

initial shapelets are of unknown type, the task of selection is to create a balance between shapelets from large clusters and shapelets that are outliers but can be counted as examples of rare anomaly types. By only choosing maximum representative final shapelets, the computational effort is reduced but nevertheless, high possible interpretability should be preserved. Shapelet multivariate time series transformation is also already widely recognized, as well for classification as anomaly detection and will be inherited by this thesis. Because labels for the whole time series, subsequence or time points are non existent, a representation is needed, that makes it possible to identify all anomaly types, point, collective or contextual outliers. Current anomaly detection literature does not provide a suitable solution with the desired representation properties, therefore a new transformation is proposed in this thesis. Lastly, anomaly detection receives the multivariate time series transformation and decides based on the distance values if a time point, sequence or time series can be identified as an anomaly or not. Anomaly detection will be unsupervised since no information about the dataset will be given. In addition to the anomaly tags of the time series, anomaly detection shall return additional information about the final shapelets, for example, the corresponding cluster or for which time points it influenced decision making about being detected as normal or anomalous.

The proposed method has two main advantages: firstly, an operator receives information where to find anomalies in the dataset. Different types of anomalies should be easily identifiable, using the cluster information of each final shapelet and where in the time series it was used for anomaly detection. Final shapelets of the same shape, in the same or close clusters, which have detected similar anomalies, form the anomaly prototypes. The previous selection should ensure that few best representative shapelets are used, which makes a possible later labelling much easier and more intuitive since the anomaly prototypes are more intuitive to distinguish and anomalies do not have to be searched manually in the dataset. Secondly, even though parts of the proposed workflow are already used in anomaly detection (e.g. subsequence clustering) or shapelet-based anomaly detection (e.g. time series transformation), it has not yet been tested as a whole in this form, so the white-box modular design allows improvements to be made more quickly or performance problems to be traced back to individual parts that can be changed in future work.

4. Basics

Based on the in chapter 3 introduced workflow, each part will be explained in detail in this section and information about the used algorithms and techniques is given.

4.1. Shapelet discovery

Shapelet discovery starts by extracting univariate shapelets from the dataset X using windowing [LKWL07]. The window size for initial univariate shapelets candidates **S** with length $|S|$ is contingent upon the time series length |X|, with tunable parameter α:

$$|S| = \alpha \cdot |X| \tag{4.1}$$

Each candidate is annotated with a variable label as preparation for the shapelet transformation in section 4.3. All initial shapelets are clustered.

A large number of clustering methods have been proposed over the past decades, including centroid-based clustering, density based clustering and deep learning based clustering [RPY+22]. Shapelets fall into the category of shape-based clustering approaches [ASW15, p. 19] and only methods suitable for anomaly detection or shapelet-based time series analysis were implemented in this work.

4.1.1. k-Means

Within the centroid-based clustering methods, k-Means is a common one used for both classification and distance based anomaly detection, as shown in section 2.3. The k-Means clustering [AV07] problem aims to partition a given dataset into k

distinct clusters, such that the within-cluster sum of squares (WCSS) is minimized. WCSS represents the sum of squared distances between each data point and its assigned centroid. For the k-Means problem, an integer k and a set of n data points $\mathbf{X} \subseteq \mathbb{R}^d$ are given. The objective is to select k centers $\mathbf{C}$ to minimize the potential function:

$$\phi = \sum_{\mathbf{x} \in \mathbf{X}} \min_{\mathbf{c} \in \mathbf{C}} ||\mathbf{x} - \mathbf{c}||^2 \tag{4.2}$$

The choice of these centres implicitly defines clustering, where each centre represents a cluster consisting of data points closer to that centre than to any other. Datapoints closest to the cluster centre are called centroids. The k-Means problem is known to be NP-hard, making it difficult to find an exact solution.

Optimal clustering is denoted as COPT and its corresponding potential as ϕ_{COPT}. For a given clustering $\mathbf{C}$ with potential ϕ, $\phi(A)$ represents the contribution of subset $A \subseteq \mathbf{X}$ to the potential, defined as $\phi(A) = \sum_{\mathbf{x} \in A} \min_{\mathbf{c} \in \mathbf{C}} ||\mathbf{x} - \mathbf{c}||^2$.

The k-Means algorithm is a simple and efficient method that aims to improve an arbitrary k-Means clustering locally. The algorithm follows these steps:

1. Arbitrarily choose k initial centers $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\}$.

2. For each $i \in \{1, \ldots, k\}$, assign the points in $\mathbf{X}$ closer to $\mathbf{c}_i$ than to any other center $\mathbf{c}_j$ $(j \neq i)$ to cluster C_i.

3. Update each centre $\mathbf{c}_i$ to be the centre of mass of all points in cluster C_i: $\mathbf{c}i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$.

Repeat Steps 2 and 3 until the clustering $\mathbf{C}$ no longer changes.

It is common practice to initialize the centres randomly from $\mathbf{X}$. In Step 2, ties can be broken arbitrarily as long as the method is consistent. Both Steps 2 and 3 guarantee a decrease in the potential function, leading to local improvements in the clustering. The decrease in potential in Step 3 can be demonstrated with the following:

Let O be a set of points with a centre of mass $\mathbf{c}(O)$, and let $\mathbf{z}$ be an arbitrary point. It follows that,

$$\sum_{\mathbf{x}\in O} ||\mathbf{x}-\mathbf{z}||^2 - \sum_{\mathbf{x}\in O} ||\mathbf{x}-\mathbf{c}(O)||^2 = |S| \cdot ||\mathbf{c}(O)-\mathbf{z}||^2 \tag{4.3}$$

Monotonicity in step 3 follows by considering O as a single cluster and $\mathbf{z}$ as its initial centre.

The k-Means algorithm is appealing in practice due to its simplicity and efficiency. However, it is only guaranteed to find a local optimum, which can often be suboptimal for the clustering problem and is highly dependent on the initialization of the centroids. To mitigate this problem, the computation is typically repeated multiple times, each time initializing the centroids differently. One approach that aims to alleviate this issue is the k-Means++ initialization scheme. At any given time, let $D(x)$ denote the shortest distance from a data point x to the closest centre that has already been chosen. k-Means++ is defined as follows:

1a. Choose an initial center c_1 uniformly at random from X.

1b. Choose the next center c_i by selecting $c_i = x_0 \in X$ with probability $\frac{D(x_0)^2}{\sum_{x\in X} D(x)^2}$.
1c. Repeat Step 1b until a total of k centers have been chosen.

2-3. Proceed as with the standard k-Means algorithm.

The weighting used in Step 1b, where centres are chosen with probabilities proportional to $D(x_0)^2$, is referred to as "D2 weighting".

4.1.2. DBSCAN

Density-Based Spatial Clustering of Applications with Noise, DBSCAN, [EKSX96, p. 2-3] [SEKX98, p. 170-181] [SSE+17, p. 2-5] is a density based clustering method, already used for anomaly detection, e.g. in [HB16], that groups together data points based on their density in the feature space. It does not require the user to specify the number of clusters beforehand.

Let X be the set of data points to be clustered. The fundamental idea is that each point in a cluster should have a neighbourhood with a minimum number of points within a given radius, indicating a higher density in that region, as described in equation (4.4). The shape of the neighbourhood $N_{eps}(x_i)$ of point x_i is determined by the chosen distance function, denoted as $dist(x_i, x_j)$, between two points x_i and x_j.

$$N_{eps}(x_i) = \{x_j \in X | dist(x_i, x_j) \leq eps\} \tag{4.4}$$

To determine clusters, a naive approach would require a minimum number ($minpts$) of points in the *eps*-neighbourhood of each point. However, this approach fails due to the presence of two types of points within a cluster: points inside the cluster (core points) and points on the cluster's border (border points).

Definition 4.1.1. *Core point: A data point $x_i \in X$ is a core point if there are at least $minpts$ data points within a distance of eps from it, including itself.*

Definition 4.1.2. *Border point: A data point $x_j \in X$ is a border point if it has fewer than $minpts$ data points within a distance of eps from it, but it is within the eps-neighbourhood of a core point.*

Figure 4.1 shows an example with core points in red and border points in yellow, $minpts$ is set to 4. Since the *eps*-neighbourhood of a border point generally contains significantly fewer points compared to the *eps*-neighbourhood of a core point it would be necessary to set the minimum number of points $minpts$ to a relatively low value to include all points within the same cluster. However, this value would not be representative of the cluster, especially in the presence of unwanted noise (blue point N). To address this, it is required for every point x_i in a cluster C that there is a point x_j in C so that x_i is inside the *eps*-neighbourhood of x_j and $N_{eps}(x_j)$ contains at least $minpts$:

$$x_i \in N_{eps}(x_j) \; and \; |N_{eps}(x_j)| \geq minpts$$

This is called directly density-reachable, which is symmetric for pairs of core points. The density postulation by DBSCAN allows to canonical extend-density reachable

for chains of points. A point x_j is density-rachable from point x_i w.r.t. *eps* and *minpts* if there exists a chain of points $x_1, ..., x_n$ where $x_1 = x_j$ and $x_n = x_i$, such that x_{i+1} is directly density-reachable from x_i. In figure 4.1 the red coloured points A are core points and directly density-reachable. For example the leftmost and rightmost points A in figure 4.1 are density reachable. The relation is transitive but not symmetric in general, except for core points.

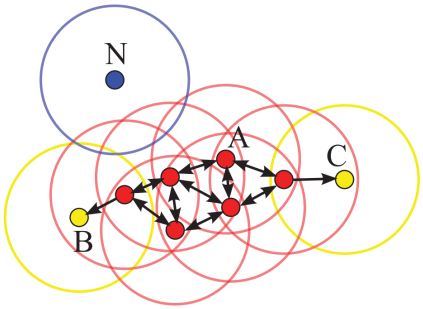


Figure 4.1. – Clustering using DBSCAN. Red points A are core points, B and C are border points, N denotes an outlier with a distance $>$ *eps* to the next core point. Figure in accordance with [SSE+17, p. 3]

While it is possible that two border points within the same cluster C may not be density-reachable from each other due to the core point condition not being satisfied for both of them, it is necessary to have a core point in C from which both border points are density-reachable. To address this situation, the concept of density-connectivity is introduced, which encompasses the relationship between border points. A point x_j is density-reachable from point x_i w.r.t. *eps* and *minpts* if there is a point x_o such that x_i and x_j are density reachable from x_o

Using these definitions, clusters can be defined as sets of density-connected points that are maximal in terms of density-reachability. Noise refers to points in X that do not belong to any cluster. A cluster C with respect to *eps* and *minpts* is a non-empty subset of X satisfying the following conditions:

1. $\forall$ x_i, x_j: if $x_i \in C$ and x_j is density-reachable from x_i with respect to *eps* and *minpts*, then $x_j \in C$ (Maximality).

2. $\forall\ x_i, x_j \in C$: x_i is density-connected to x_j with respect to *eps* and *minpts* (Connectivity).

One major advantage of DBSCAN is it ability to sort out noise points in clustering.

Definition 4.1.3. *Noise point: Noise is defined as the set of points in the database X that do not belong to any cluster C_i, i.e., noise $= \{x_i \in X | \forall i : x_i \notin Ci\}$ for all clusters $C_1, ..., C_k$ w.r.t. eps_i and $minpts_i$, $i = 1, ..., k$.*

A noise point is shown as a blue outlier in figure 4.1.

4.1.3. Mdc-CNN

k-Means and DBSCAN utilizing the Euclidean distance, as described above, can not be applied to time series classification if candidate shapelets are of different lengths. For multi-length shapelet clustering, ShapeNet adopts a few existing studies as building blocks to embed all the shapelet candidates from the original space into a new unified space [LCX+21a, p. 8375f.].

As basis a Convolutional Neural Network (CNN) [LHBB99] is used. Deep CNNs have demonstrated remarkable performance in image recognition tasks, achieving human-level accuracy, as well as in natural language processing tasks. Inspired by the success of CNN architectures in these domains, researchers have started to explore their application in time series analysis [FFW+19, p. 924f.][CYPY21, p. 120054f.]. In the context of time series analysis, a convolution for two real-valued time series $f, g : \mathbb{R}^n \to \mathbb{R}$ operation can be viewed as the application of a sliding filter over the time series. The filter f can be seen as a generic non-linear transformation applied to the time series g [GBC16, p. 327f.]:

$$[f * g](t) = \int_{R^n} f(u)g(t-u)du = \sum_{u=-\infty}^{\infty} f(u)g(t-u) \tag{4.5}$$

where the second part of the formula denotes the discrete convolution under the assumption, that f and g are defined only on integer t. The general formulation

for applying one-dimensional continuous convolution F at a centred time stamp t is given by the following equation [FFW+19, p. 924f.]:

$$F_t = a(f * X_{t-l/2:t+l/2} + b) \mid \forall t \in [1, T] \tag{4.6}$$

where F_t denotes the result of a convolution (dot product $*$) applied on a univariate time series X of length T with a filter f of length l, a bias parameter b and a final non-linear activation function a. As can be noted in the calculation of F_t, the original convolution violates the temporal causality, because values after a given time point t are included $(t + l/2)$. [BKK18, p. 3f.] presented the dilated causal convolutional network (dc-CNN) which is designed such that the future data does not "leak" to the past. Secondly, dc-CNN can take sequences of any length and map them to an output sequence of equal length. The dilated convolution operation f on element t of a sequence X is defined as:

$$F(t) = (X *_d f)(t) = \sum_{i=0}^{fs-1} f(i) \cdot X_{t-d \cdot i} \tag{4.7}$$

with dilation factor d, filter size fs and $t - d \cdot i$ accounting for the direction of the past. For $d = 1$, dilated convolution equals conventional convolution. For every layer, the number of accessible input points for one filter, the receptive field, can be increased by a larger filter size or dilation factor, where the effective history of one such layer is $(fs - 1)d$. Commonly, d is exponentially increased with the depth of the network, i.e. $d = O(2^i)$ at level i of the network. Although the dc-CNN output can be of the same length as the input, it can not handle inputs of various lengths simultaneously. ShapeNet [LCX+21a, p. 8377] modified the dc-CNN by stacking a max pooling layer and a linear layer on top of the last dc-CNN layer, to embed different sized shapelets into a unified space. The output of the new multi-length-input dilated causal CNN (Mdc-CNN) is called the shapelet candidate embedding. A further illustration of the Mdc-CNN can be seen in figure A.1.

The Mdc-CNN can be trained in an unsupervised manner. The objective of learning/training is to ensure, that similar shapelets obtain similar representations and vice versa. The cluster-wise triplet loss [LCX+21a, p. 8377f.] takes multiple posi-

tive and negative samples and the distances among them as input. One candidate shapelet is randomly selected as anchor sample s. Then from the same cluster, top S^+ other shapelet candidates nearest to the anchor are chosen as positive samples $\mathbf{s}^+$. For negative samples $\mathbf{s}^-$ candidates are randomly picked from other clusters. Various triplet tuples $(s, \mathbf{s}^+, \mathbf{s}^-)$ are constructed, namely $(s, \cup_{i\in[1,S^+]} s_i^+, \cup_{i\in[1,S^-]} s_i^-)$ from shapelet candidates at the beginning of each iteration. For simplicity, in the following, the loss is demonstrated for two clusters. First, the normalized distance of the positive (negative) samples from the anchor is denoted as $\mathcal{D}_{AP}$, $\mathcal{D}_{AN}$ and a margin t_{margin} is enforced as regulation parameter:

$$\mathcal{D}_{AP} + t_{margin} < \mathcal{D}_{AN} \tag{4.8}$$

With implementation of squared mean Euclidean distance, $\mathcal{D}_{AP}$ and $\mathcal{D}_{AN}$ can be defined as

$$\mathcal{D}_{AP} = \frac{1}{S^+} \sum_{i=1}^{S^+} ||f(s) - f(s_i^+)||_2^2 \tag{4.9}$$

and

$$\mathcal{D}_{AN} = \frac{1}{S^-} \sum_{i=1}^{S^-} ||f(s) - f(s_i^-)||_2^2 \tag{4.10}$$

where $f(\cdot) \in \mathbb{R}^z$ is the representation embedded by Mdc-CNN of length z. Additionally to the distances between the anchor and the positive (negative) samples, the distance among the positive $\mathcal{D}_{pos}$ (negative $\mathcal{D}_{neg}$) samples are included and should be small (large):

$$\mathcal{D}_{pos} = \max_{i,j\in(1,S^+)\wedge i<j} \{||f(s_i^+) - f(s_j^+)||_2^2\} \tag{4.11}$$

and

$$\mathcal{D}_{neg} = \max_{i,j\in(1,S^-)\wedge i<j} \{||f(s_i^-) - f(s_j^-)||_2^2\} \tag{4.12}$$

The intra-sample loss is defined as:

$$\mathcal{D}_{intra} = \mathcal{D}_{pos} + \mathcal{D}_{neg} \tag{4.13}$$

Putting all together, the cluster-wise triplet loss function for training the Mdc-CNN unsupervised is given as:

$$\mathcal{L}(f(x), f(\mathbf{s}^+), f(\mathbf{s}^-)) = log\frac{\mathcal{D}_{AP} + t_{margin}}{\mathcal{D}_{AN}} + t_{intra} \cdot \mathcal{D}_{intra} \tag{4.14}$$

with hyperparameter t_{intra}. Following the standard practice [GBC16, p. 330], Mdc-CNNs use shared weights for training models of shapelet candidates of different lengths and variables. The derivation of the loss function for the network training is given in A.

4.2. Shapelet selection

Shapelet selection determines the final shapelets S_{f_S}. Methods are the in section 2.3 introduced information gain [AA22] or a utility function by ShapeNet [LCX+21a, p. 8379]. Since information gain, as introduced in anomaly detection, rely on labelled data, the utility function will be used within this work. Shapelet selection ranks the candidate shapelets and selects the most representative ones by a utility measure denoted by $\mathcal{U}(s_i)$. The first term in $\mathcal{U}(s_i)$ represents the size of the candidate's cluster. A larger cluster indicates that the candidate represents a greater number of candidates. The second term measures the candidate's distance to other candidates in different clusters. A larger distance suggests that the candidate is distinct from the other candidates. For k-Means, a simplification can be made by considering only centroids as shaplelet candidates, since these correspond to the average of a cluster and the computational effort is thus reduced:

$$\begin{aligned} \mathcal{U}(s_i) = \gamma \cdot \frac{\log(\text{size}(\mathbf{C}(s_i)))}{\log\left(\max_{i=1}^{Y}(\text{size}(\mathbf{C}(s_i)))\right)} + \\ (1-\gamma)\frac{\log \sum_{j=1}^{Y} ||\mathbf{C}(s_i) - \mathbf{C}(s_j)||_2^2}{\log(\max_{i=1}^{Y}(\sum_{j=1}^{P} ||\mathbf{C}(s_i) - \mathbf{C}(s_j)||_2^2))} \end{aligned} \tag{4.15}$$

where s_i with $i = 1, ..., Y$ is the currently evaluated shapelet of all Y initial shapelets. $\mathbf{C}(s_i)$ denotes the cluster of shapelet s_i and $\gamma \in [0, 1]$ controls the trade-off between cluster size and distance to other candidates. To further simplify the calculations,

for k-Means only shapelets that are centroids or closest to them are considered for evaluation.

The top-f_S candidates among all Y clusters based on the ranking provided by equation (4.15) are selected. These selected candidates correspond to the original time series subsequences and are denoted as S_{f_S}, representing the final shapelets.

4.3. Shapelet transformation

Shapelet transformation was proposed for the Shapelet Tree algorithm. Shapelets were used to form rules within a decision tree. It was later shown, that using shapelets in data transformation produces significantly better accuracy for complex classifiers combined with faster enumeration [BB17, p. 1]. Hence, by interpreting anomaly types as classes, shapelet transform achieved comparable to above-average anomaly detection scores for supervised studies [BKS+18][AA22]. Shapelet transformation, for a chosen set of shapelets $\mathbf{S} = (S_1, ..., S_{f_S})$, is a method to transform a multivariate time series dataset $\mathcal{D}$ of dimension $M \times V \times N$ into a new data space, by calculating the distance between each shapelet in $\mathbf{S}$ and each multivariate time series instance X_m in $\mathcal{D}$ [LCX+21a, p. 8379][ASW15, p. 23]:

$$\Phi(X_m; S_{f_S}) = dist(X_m^v; S_{f_S}) \tag{4.16}$$

where the variable $v \in V$ of X_m^v and S_{f_S} is the same. The $M \times k$-dimensional feature representation of the MTS instance X_m is then given as:

$$\Phi(X_m; \mathbf{S}) = \Phi(X_m; S_1, ..., S_{f_S}) \tag{4.17}$$

After the transformation of the multivariate time series is completed, some standard classifiers can be applied to learn a classification model from the representation [LCX+21a, p. 8379] [BKS+18, p. 951-953].

The originally proposed method for obtaining $dist(X_m^v; S_{f_S})$ conducts a striding euclidean distance calculation between a given shapelet S_{f_S} of length $|S_{f_S}|$ and a

time series X_m^v with length $|X_m^v| > |S_{f_S}|$ for all subsequences $(x_{m,j}^v, ..., x_{m,j+|S_{f_S}|-1}^v)$ in X_m^v. As the final representation value is the minimum along all initial time points j returned, as seen in formula (4.18) [LCX+21a, p. 8379] [BKS+18, p. 950]:

$$dist(X_m; S_{f_S}) = \min_{j=1,...|X_m^v|-|S_{f_S}|+1} \frac{1}{|S_{f_S}|} \sum_{l=1}^{|S_{f_S}|} (x_{m,j+l-1}^v - x_{S_{f_S},l})^2 \tag{4.18}$$

Intuitively, $\Phi(X_m; S_{f_S})$ is the distance of the shapelet to the most similar subsequence in X_m^v. If time series X_m^v consist of more than one class, it needs to be split in sub parts of length larger than $|S_{f_S}|$, each representing one class.

Using the minimum Euclidean distance as an anomaly score metric has a few major drawbacks. The minimum Euclidean distance was admittedly successfully used to determine if a given time series, or sub part of it, is anomalous or not. But only for supervised experiments, where the time series, or sub parts, can be assigned to one class and is labelled as such [BKS+18, p 951-959][BB17, p. 3-4][AA22, p. 5-12]. For example, [AA22] manually labelled every time series part as normal class or as a member of different anomaly types. This is not applicable for truly unsupervised environments, where the number of anomaly types is unknown and little to no information is present to their exact position in the dataset, making it impossible to draw clear class boundaries. The decision, whether a data point or a subsequence is anomalous or not and a subcategorization of anomaly types, needs to be possible without further information, but the raw dataset.

Regarding those problems, an adapted distance metric for time series transformation is proposed. Instead of using the minimum Euclidean distance, the squared distance was used. For all subsequences $(x_{m,j}^v, ..., x_{m,j+|S_{f_S}|-1}^v)$ in X_m^v starting at initial time point j, it is defined as:

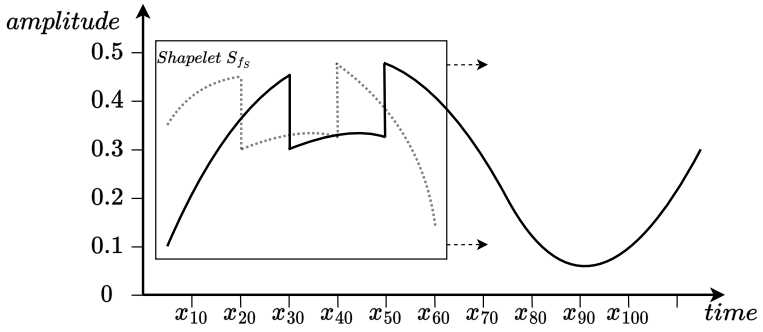
$$dist_j(X_m^v; S_{f_S}) = (x_{m,j}^v - x_{S_{f_S},j})^2;\ j = 1, ..., |S_{f_S}| \tag{4.19}$$

resulting in $dist_j(X_m^v; S_{f_S})$ to be of length $|S_{f_S}|$. The overall distance $dist(X_m; S_{f_S})$ between the whole time series instance and shapelet S_{f_S} is then found by striding the shapelet over all starting points $j = 1, ..., |X_m^v| - |S_{f_S}| + 1$ and only updating

$dist(X_m; S_{f_S})$ if the current distance $dist_{j+1}(X_m^v; S_{f_S}) = dist_{j+1}$ is smaller for every value than $dist_j(X_m^v; S_{f_S}) = dist_j$, for all matching x:

$$dist(X_m; S_{f_S}) = \begin{cases} dist_j(X_m^v; S_{f_S});\ dist_{j+1} > dist_j, \forall x\ \in\ dist_j \\ dist_{j+1}(X_m^v; S_{f_S});\ dist_{j+1} < dist_j, \forall x\ \in\ dist_j \end{cases} \tag{4.20}$$

Figure 4.2 shows the distance calculation and the final distance vector $dist(X; S_{f_S})$ between a generic anomalous time series X and a shapelet S_{f_S}, where the distance values are given for every tenth $x \in X$.



$dist(X_{x_{10}-x_{60}}, S_{f_S})$	0.18 0.06 0.03 0.01 0.08 0.28
$dist(X_{x_{20}-x_{70}}, S_{f_S})$	0.03 0.0 0.0 0.0 0.03 0.02
$dist(X, S_{f_S})$	0.18 0.03 0.0 0.0 0.0 0.03 0.02 ⋯

Figure 4.2. – Time series transformation for a generic time series X and a exemplary shapelet S_{f_S} with the proposed distance calculation

The new time series representation $\Phi(X_m; \mathbf{S})$ is now of shape $V \times N \times f_S$, instead of a vector, depicted in figure 4.3. One major advantage of this calculation is the possibility to implement anomaly detection methods on time point level, while the time dependency within the shapelet sequence is not lost. Most importantly it allows completely unsupervised anomaly detection since no prior time series labelling is needed. With a large enough amount of shapelets S_{f_S}, anomalous points, or subsequences, should manifest by having large distances to many and small distances to few shapelets, assuming normal class majority in the dataset. An unsupervised clustering method should be able to identify those points and associated shapelets,

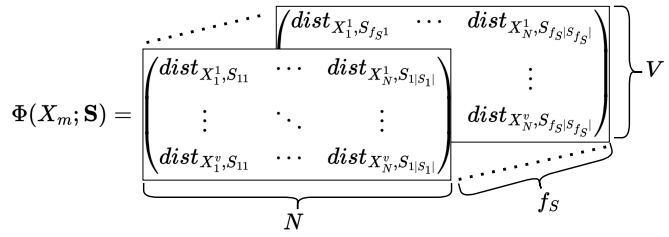


Figure 4.3. – Multivariate time series transformation for a time series instance X_m and f_S shapelets S_{f_S}

allowing to trace back these shapelets within the shapelet clusters, indicating if a given shapelet cluster accounts for a unique type of anomaly or normal class.

4.4. Shapelet-based anomaly detection

In this work, the proposed classification case of the transformed time series in ShapeNet is transitioned to an unsupervised anomaly detection task by implementing already examined algorithms. Current shapelet-based anomaly detection does not support unsupervised anomaly detection utilizing the representation $\Phi(X_m; \mathbf{S})$ [AA22, BKS+18]. As a result, algorithms from the broader field of time series clustering and anomaly detection are employed. Given the absence of labelled data, the approach essentially resorts to clustering techniques, which are already approved for anomaly detection [BGCML21, p. 26]. Among the various clustering algorithms, k-Means has proven to be effective and extensively researched [ASW15, p. 26-28][YKH01]. Therefore, k-Means was adopted for the task.

The primary objective is to cluster time points or sequences into anomalous or non-anomalous categories. Due to the unknown number of anomaly types, the clustering process often aims to identify patterns and groupings that distinguish anomalies from normal instances, a binary decision making [AAXJ21, p. 8]. By leveraging k-Means, the only parameters to discover meaningful clusters that capture the underlying structure of the representation are the cluster number k set to 2.

As mentioned in the previous chapter, the closest final shapelet to an anomalous point or sequence is easily identifiable, by searching for the index k of the lowest distance in $\Phi(X_m; \mathbf{S})$. Following the denotation in chapter 3 those shapelets are called the anomaly prototypes. Tracing back the anomaly prototypes to its adjunct cluster in shapelet discovery gives interpretable information about the estimated number of anomaly types and their distance to other types in the shapelet space. With described shapelet-based anomaly detection, an operator receives information, where possible anomalies are and if the detection was based on normal class or by finding an anomaly prototype. As shapelet discovery and selection reduced the size of the shapelet space while maintaining validity, anomaly prototypes account for a significantly smaller number of shapelets than initial candidates accelerating anomaly type labelling.

5. Implementation

In this chapter, the fundamentals of the implementation are described. The first part describes both the measured dataset in section 5.1 and subsequently the synthetic dataset in section 5.2, while evaluation measures are explained in section 5.3. Lastly, in section 5.4 the experimental design is shown.

5.1. Eden ISS FEG Dataset

In this work, the dataset from the Future Exploration Greenhouse (FEG) within the EDEN ISS Project [ZBV+15, Dan17, BV17] was used. The FEG is part of the mobile test facility with two major objectives [ZBV+15, p. 8]:

"Design of a space analogue mobile test facility for a 12+ month mission in Antarctica to provide representative mass flows and proper test environments for plant cultivation technologies as an essential on-ground preparatory activity for future space exploration."

and

"Integration and test of key elements for plant cultivation in 1) an ISPR-like system (International Standard Payload Rack) for future tests on-board ISS and 2) a Future Exploration Greenhouse (FEG) to prepare for closed-loop bio-regenerative life support systems."

A schematic view of the mobile test facility can be seen in figure 5.1. It is divided into two parts, the FEG and a service section, containing all monitoring and controlling systems. The FEG is the main plant growth area of the mobile test facility, including

multilevel plant growth racks operating in a controlled environment [ZBV+15, p. 12]. Cultivated crops include lettuce, spinach, tomatoes, cucumbers, radish and more [1] [ZBV+15, p. 152].

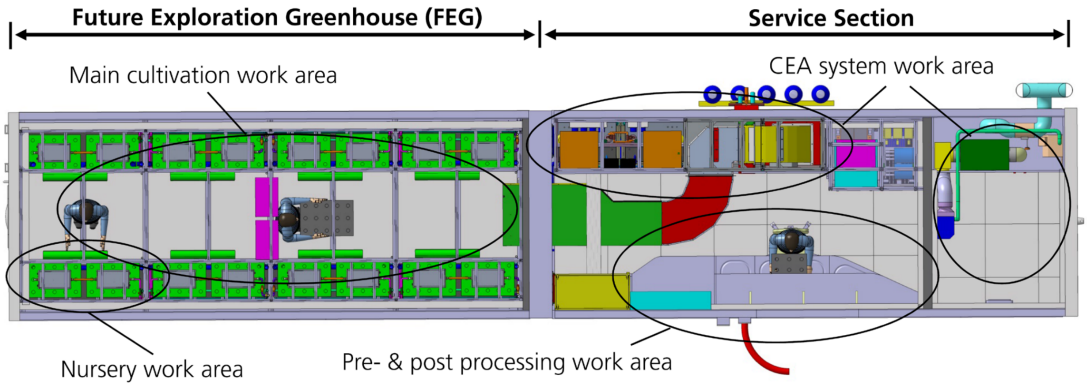


Figure 5.1. – The schematic structure of the FEG and affiliated service station [Sch17]

Monitoring inside the FEG started on 11.01.2018 and ended on 31.12.2020, with a sample every five minutes, resulting in 312642 samples. The used dataset consists of measurements taken from nine different sensors. Sensors include: temperature, CO_2 concentration, relative humidity and photosynthetic active radiation. Photosynthetic active radiation is the radiation emitted by the lighting system and accounts for the range of light usable for photosynthesis processes [ZBV+15, p. 141]. The dataset is entirely unlabeled, i.e. unsupervised. Exemplary anomalies were found by visual inspection and a sparse logbook, more information is given in section 5.2.3. Before the proposed methods could be applied, data preprocessing was conducted. Two sensor time series were disregarded because the values for the entire first year were missing. The remaining sensors had between 52 and 58 missing samples during recording. Interpolation was done by overriding with the last existing value. Finally, the multivariate time series dataset consists of 7 variables with 312643 samples each. The characteristics and modalities of the dataset will not be discussed further here, as they will be discussed in detail in the next section on generating a synthetic dataset. An excerpt of the dataset can be seen in Appendix A.2.

[1] for more information about the EDEN ISS Project, visit: https://eden-iss.net/

5.2. Synthetic dataset

Before applying the described methods to the EDEN ISS FEG dataset, experiments will be conducted on a synthetically generated dataset. This allows for better control of anomaly types, their clear properties and the use of labels. Generating a synthetic data set with similar properties and identical anomalies is a common practice for evaluating model capabilities [GCC+23, p. 4]. Various studies have introduced synthetic strategies, but they focus on their specific application or miss a sufficient type diversity and can not serve as a general synthetic criterion for anomaly clustering benchmarking. The synthetic dataset serves as a blueprint to optimize the proposed methods, while ensuring the transferability of findings to the FEG dataset, since they are approximately equal in statistic characteristics and anomaly types [LZX+21, p. 6][LIPJ21, p.12][BRGD19, p. 9].

The synthetic dataset was built according to the principle design of the EDEN ISS project design report and related documents [ZBV+15, Dan17, BV17]. Values or dynamics which are not explained or defined within this project are taken from the literature about similar greenhouse environments. The dataset is divided into four independent and two dependent variables.

5.2.1. Independent variables

Concerning anomaly generation, independent variables are those that are modelled from the EDEN ISS FEG documentation, whereas dependent variables are calculated. Independent variables are plantmass, photosynthetic active radiation, temperature and relative humidity.

The main goal of the FEG is a constant supply of fresh food for eventual future space stations and explorations. One crucial metric for project success was the constant rate of fresh food yield. Therefore plantmass [g] was included as one of the independent variables. In addition to the data gathered inside the FEG, growth modelling was supported by findings from similar closed, controlled environment experiments for example the controlled ecological life support system conducted by the National

Aeronautics and Space Administration (NASA) in the 1990s [WMS+94]. plantmass, or its derivative: growth, are key factors that describe the efficiency and success of the FEG. Inside FEG different harvesting techniques are used, from steadily harvest to clear growth and harvest periodic cycles [ZBV+15, p. 146][MDSK17, p. 18, 20, 24]. Plant growth modelling was done exemplarily on lettuce, one of the successfully tested crops, which showed extraordinary usability for closed, controlled environments due to its modest demands. The lettuce will be fully grown before being harvested. In general, lettuce growth in controlled, encapsulated environments can be divided into three stages: a logarithmic, an exponential and, after reaching near maximum mass, a stationary phase. Approximation of fresh weight per time period is often done by a fitted sigmoid curve matching the three stages [van80, p. 6-8][SKF08, p. 2-7][WMS+94, p. 611-613][DPA05, p.310]. Vegetation periods range between 30 - 45 day periods with a final weight of approximately 170g - 190g [KMRH19, p. 51-53][SKF08, p. 2-7][KKA+13, p. 502]. For synthetic data generation, the growth period was set to 40 day periods, as taken from the FEG documentation [MDSK17, p. 20], followed by a complete harvest with a final weight of 180g.

Photosynthetic active radiation is the radiation emitted by the lighting system and accounts for the range of light usable for photosynthesis processes [ZBV+15, p. 141]. It is measured in $\mu\text{molm}^{-2}\text{s}^{-1}$. The optimal PAR for all different types of lettuces was found to be around 300 $\mu\text{molm}^{-2}\text{s}^{-1}$ for a day period and 0 for the night period respectively [Dan17, p. 11]. One day period consists of 16h illumination time and 8h of darkness for all crops [ZBV+15, p. 139-140].

Temperature [°C] was modelled according to the project design document. The temperature inside the FEG needs to fit a general regime to be in the "sweet spot" for all plants and was therefore set to 20-22°C during day period and 16-18°C during night period [ZBV+15, p. 139-140]. The temperature control automatically follows this periodic cycle and makes it therefore unnecessary to include effects like heat exchange with the environment. The temperature control can only match the desired temperature within a range [BV17, p. 74-81]. Temperature mean, in the synthetic dataset, was set to 21°C and 17°C for day period and night period respectively, with a Gaussian standard deviation. Given a mean μ, σ can be chosen, so that 99.7% of the realisations lie within a given interval, $\mu \pm 3\sigma \leftrightarrow \sigma = \mu/3$. Setting μ to the maxi-

mum allowed divergence, sigma can be easily found by $\sigma = |21°C - 22°C|/3 \approx 0,33$, meaning that with calculated σ temperature divergences are with a probability of 99.7% within the range of ±1°C. The duration of heating and cooling of the FEG was not given and could only be approximated by visually inspecting the measured data. Heating and cooling were both simulated by an adapted sigmoid function:

$$T(x) = T_{dayperiod} + (T_{nightperiod} - T_{dayperiod}) \frac{T_{nightperiod} - T_{dayperiod}}{1 - e^{-a \cdot x}} \tag{5.1}$$

whereas $T_{dayperiod}$ conforms 21°C and $T_{nightperiod}$ conforms 17°C. a denotes a silhouette parameter, influencing the slope of the function and was set to $a = 5 \vee 1$ for heating or cooling, while x equals x or $-x$, mirroring the function axisymmetrical. The duration of heating was set to one hour and cooling to 20min. Cooling the FEG can be modelled faster than heating because the FEG was tested in Antarctica, where cooling can be conducted by inserting external air.

Relative humidity (RH) in [%] is an important factor for plant growth. Values below 50% leads to plants closing their stomata which affects the uptake of CO_2. On the other hand, a high relative humidity (>90%) can be a starting point for fungal diseases. For the duration of all FEG experiments, the relative humidity was set constant to 70% ± a maximum percentage deviation (mdp) of 5%. Modelled RH was set accordingly with an added sinus daily cycle reciprocal proportional to temperature and an amplitude of 0.5. Deviation for modelled RH is taken from Gaussian standard deviation. Similar to temperature, σ was determined by $\sigma = (mean_{RH} \cdot mpd - 1)/3 = (70 \cdot 0.05 - 1)/3 \approx 1$ [ZBV+15, p. 139-140].

5.2.2. Dependent variables

One major aim of this research is the examination of shapelet clustering for anomaly detection in complex environments. The EDEN ISS is such an environment, where causal connections appear between various variables. Still, it only includes one calculated variable within the dataset, the vapour pressure deficit. It defines the difference between actual vapour pressure and saturation vapour pressure and can be expressed

dependent on relative humidity and temperature. For a higher system complexity, another dependent variable is added to the dataset, the photosynthetic rate.

Vapour pressure is an important value for plant growth regulation. In contrast to relative humidity, vapour pressure has a more direct effect on the evaporation rates of leaves. To convert relative humidity to vapour pressure deficit [kPa] the Antoine equation is used [dLFRA21]. The Antoine equation represents the non-linear thermodynamic relationship between equilibrium vapour pressure P and temperature T [°C] [dLFRA21, p. 1-2] by:

$$P(T) = \eta(T) + \epsilon = 10^{a - \frac{b}{c+T}} + \epsilon \tag{5.2}$$

a, b and c are numerical constants related to the enthalpy and entropy of vaporization and vary for different pure substances. Within a temperature range $T \in [1, 100]$°C the best guesses are a = 8.07131, b = 1730.63 and c = 233.426 for water above sea level [dLFRA21, p. 6-7]. Because of the underlying normality assumption ϵ represents the constant error and is taken from the normal distribution with $Var(P(T)) = \sigma_0^2$ with $\epsilon \sim \mathcal{N}(0, \sigma_0^2)$ for the homoscedastic case and $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2 \eta(T)^2)$ for the heteroscedastic case, where normally a homoscedastic case can be assumed. By multiplying equilibrium vapour pressure and relative humidity RH the actual vapour pressure can be received $P_{act} = \frac{\text{RH}}{100} * P$ to obtain vapour pressure deficit VPD with:

$$VPD(T) = P(T) - P_{act}(T) \tag{5.3}$$

Photosynthetic rate, PR, determines how much CO_2 can be absorbed, measured in μmolm$^-$2s$^-$1, and has a direct influence on O_2 production, a critical resource for space exploration missions. With PR being a natural process, a lot of factors have an influence, resulting in a wide range of studies evaluating different photosynthetic rate models for lettuce in controlled, encapsulated environments [ZLW22, JKY+16, VBBD17, LKO+12, WMS+94, van81]. The EDEN ISS FEG does not provide an explicit photosynthetic model, CO_2 consumption for the whole FEG and different crop types were taken from a simple constant estimation [ZBV+15, p. 142-143, 151]. PR models forecast either for single leaves, whole plants or whole

canopy. For easier adaption, a single lettuce leaf model was chosen from literature, depending on temperature (T), photosynthetic active radiation (PAR) and CO_2 concentration [VBBD17, p. 129-130]:

$$PR = \frac{1}{2}\theta \cdot \left(q \cdot PAR + PR_{max} - \sqrt{(q \cdot PAR + PR_{max})^2 - 4\theta q \cdot PAR \cdot PR_{max}}\right) \tag{5.4}$$

where q is the quantum yield and θ is dependent on the leaf angle. θ can be set to 1 for simplicity since leaves of lettuce plants inside the FEG will be automatically growing to an optimal light absorbing position. PR_{max} denotes the theoretically maximum photosynthetic rate and is found experimentally by a fitted asymptotic curve parameterized by constants c_1, c_2 and c_3. Its main influence is CO_2 concentration as shown in Equation 5.5. CO_2 concentration was set to a constant 750ppm inside the FEG and will not be handled as a dependent variable due to its regulation by the environment control system [ZBV+15, p. 63][BV17]. Values for constants are $c_1 = -0,116,\ c_2 = 1,136$ and $c_3 = -0.002$ [VBBD17, p. 131].

$$PR_{max} = c_1 + c_2 \cdot (1 - e^{c_3 \cdot \mathrm{CO_2}}) = -0,116 + 1,136 \cdot (1 - e^{-0.002 \cdot \mathrm{CO_2}}) \tag{5.5}$$

The quantum yield q expresses the photochemical efficiency and is a temperature-dependent parameter [van81, p.3]. Temperature is measured in °C. As for PR_{max}, constants for the polynomial function of q were determined by curve fitting [VBBD17, p. 130].

$$q = c_4 - (c_5 T) - (c_6 T^2) = 0,0843 - (0,0003T) - (0,0000341T^2) \tag{5.6}$$

Documentation of the FEG suggests an estimated quantum yield of 4%, which would account for a temperature of around 30°C in equation 5.6, still, for a more precise model the quantum yield will not be taken as constant [ZBV+15, p. 142]. A limitation of equation 5.4 is the incorrect behaviour for periods involving turned-off lighting system. The model would assume a PR of zero, while in real world research a dark respiration rate can be observed, which describes the changing modality of lettuce being a sink of CO_2 to becoming a source during night periods [ZBV+15, p. 142]. Dark respiration rate R_d is an empirical value [van81, p. 14] and was estimated to be

5g/h for the whole FEG, values for different types of crops were not given [ZBV+15, p. 142]. In literature, under similar conditions, values for R_d for single leaf models differ, therefore a value from the most similar study was chosen, estimating R_d to be at $3\mu\text{molm}^{-2}\text{s}^{-1}$ [ZLW22, p. 4-5], extending equation 5.4 to:

$$PR = \frac{1}{2} \cdot \left(q \cdot PAR + PR_{max} - \sqrt{(q \cdot PAR + PR_{max})^2 - 4q \cdot PAR \cdot PR_{max}} \right) - R_d \tag{5.7}$$

Light use efficiency is known to be influenced by leaf size and growth stages. [JKY+16, LKO+12] propose a simple multiplication model. [LKO+12, p. 1258] suggest the deviation of plantmass, the growth rate in $[\frac{g}{d}]$, as multiplication factor, while [JKY+16, p. 488] adapt this to be not only accountable for whole plants but also single leaf models by making the deviation impartial of the weight and only dependent on time. Normally, the derivative of the sigmoid function is expressed as

$$\frac{d}{dt} sig(t) = \frac{d}{dt} \frac{1}{1 + e^{-t}} = \frac{1 + e^{-t}}{1 + e^{-2t}} \tag{5.8}$$

but because harvesting is directly followed by newly planting, the plantmass is not a steady function, hence using the simplification of $\frac{d}{dt} sig(t)$ [Cop04, p. 302]:

$$\frac{d}{dt} sig(t) = \frac{1 + e^{-t}}{1 + e^{-2t}} = sig(t)(1 - sig(t)) \tag{5.9}$$

By combining equation 5.7 and 5.9, the final equation for the photosynthetic rate of one lettuce leaf can be expressed by:

$$\begin{aligned} PR = &\left[\frac{1}{2} \cdot \left(q \cdot PAR + PR_{max} - \sqrt{(q \cdot PAR + PR_{max})^2 - 4q \cdot PAR \cdot PR_{max}} \right) - R_d \right] \\ &\cdot [sig(t/lcl)(1 - sig(t/lcl))] \end{aligned} \tag{5.10}$$

with life-cyclus-length, lcl, of $40 dayperiods \cdot \frac{samples}{dayperiod}$ and depending on temperature, photosynthetic active radiation, plant growth and CO_2 concentration.

For model discussion, the PR model was evaluated against literature and available information from the FEG. With a maximum of $4.4479\mu\text{molm}^{-2}\text{s}^{-1}$ it is in a typical

range for single leaf lettuce models [VBBD17, p. 132-133],[ZLW22, p. 178],[JKY+16, p. 489-490], [KKA+13, p. 506-507] [LKO+12, p. 1258 -1259], [WMS+94, p. 612]. The FEG data only provides a single estimation for CO_2 consumption at one timepoint for a whole crop of lettuce of $12\text{gm}^{-2}\text{h}^{-1}$ [ZBV+15, p. 151]. To convert $4.4479\mu\text{molm}^{-2}\text{s}^{-1}$ to $\text{gm}^{-2}\text{h}^{-1}$, only the molar mass of CO_2 (44.01gmol^{-1}) is needed:

$$4.4479\mu\text{molm}^{-2}\text{s}^{-1} \Rightarrow 4.4479\text{molm}^{-2}\text{s}^{-1} \cdot \frac{44.01\text{gmol}^{-1} \cdot 3600\text{s}}{10^6\text{m}^{-2}\text{h}^{-1}} = 0.704707\text{gm}^{-2}\text{h}^{-1} \tag{5.11}$$

For an actual comparison, the number of leaves per crop is missing. In controlled greenhouse environments the number of leaves per plant, which accomplish a life cycle, is 18 on average [DPA05, p. 308-311]. This brings the final CO_2 consumption to $0.704707\text{gm}^{-2}\text{h}^{-1} \cdot 18 = 12.6847\text{gm}^{-2}\text{h}^{-1}$ per lettuce. Even though scaling up the CO_2 consumption rate of single leaves up to crop or canopy levels is not straightforward and may cause errors [LKO+12, p. 1259], with the proposed PR model the MAPE against the FEG estimations is at a low enough level of $|\frac{12\text{gm}^{-2}\text{h}^{-1} - 12{,}6847\text{gm}^{-2}\text{h}^{-1}}{12\text{gm}^{-2}\text{h}^{-1}}| \cdot 100\% = 5,71\%$.

5.2.3. Anomaly generation

In this section, the anomaly types and their generation are explained. First of are general design principles introduced, followed by the description and implementation of the anomalies themselves. Anomaly types were identified by visually inspecting the FEG dataset. Overall, seven types were found, spike, drop, zero, missing, noise, level shift and time shift anomalies. Each will be explained subsequently, while also giving an example from the dataset aided by a cause explanation, if available. Anomalies can be generated both multivariate and univariate, but are always independent from each other. Multivariate anomalies, manifesting simultaneously in all six variables, are counted as one since anomalies are counted regarding their origin. Anomalies are added using a Monte-Carlo simulation, with an independent and adjustable likelihood for every anomaly type to occur or start, for sequential anomalies, at a time step x_t. Anomalies can not overlap. Termination criteria are an

adjustable maximum anomalous rate $MAR = \frac{number\ of\ anomalous\ timesteps}{number\ of\ timesteps}$ or a prior defined maximum number of anomalies, whatever will be exceeded first. Anomalies in dependent variables can be handled in two ways. The first is that anomalies are added by error propagation, where the dependent variables are calculated based on the anomalous modelled data. This would account for real world scenarios, where flawed sensor data will produce flawed calculated data. The second way is to calculate dependent variables with correct recorded data and add anomalies on the full dataset without the dependence of the individual time series variables. For the whole thesis, the first case will be set as standard as it suits to be closer to the real process.

Shapelets are a length-sensitive clustering method, so the length of the sequential anomalies was computed correlating to the current time series length, by multiplying the current time series length with a variable factor. The multiplication factor was randomly drawn from an exponential distribution [EH04, p. 94-96]. The exponential distribution ensures that the probability of the anomaly length declines with a high probability of lower lengths and a low probability of high lengths. Hence it is often used in modelling failure rates and senescence. The probability density function is parameterized in terms of λ, the rate parameter, or $\beta = 1/\lambda$, which is called scale parameter:

$$f(x;\lambda) = \lambda \cdot e^{-\lambda \cdot x} = \frac{1}{\beta} \cdot e^{-\frac{1}{\beta} \cdot x};\ x \geq 0 \tag{5.12}$$

Given the expected value of the exponential distribution function $E(x) = \frac{1}{\lambda} = \beta$, the α-quantile ξ_α with $\alpha \in [0;1]$ can be expressed as:

$$\xi_\alpha = -\frac{1}{\lambda} \cdot \ln(1-\alpha) = -\beta \cdot ln(1-\alpha) \tag{5.13}$$

with a median of $\xi_{0.5} = \beta \cdot \ln(2)$ and the mean of β. The exponential distribution was also used to calculate the magnitude of spike, drop and level shift anomalies.

Spike and drop anomalies are part of the point anomalies or outliers. Such anomalies are often caused by sensor errors or abnormal system operations [CYPY21, p. 120044]. To account for the diverse variances of the time series variables, a draw from the exponential distribution was utilized. This draw, denoted as magnitude m,

was then multiplied with the mean value of the current variable $\mu(X_i)$ within the time series. This approach facilitated the computation of spikes and drops in the data. [LZX+21, p. 6].

$$\hat{x_t} = x_t \pm m \cdot \mu(X_i);\ m \sim exp(\frac{1}{\beta}) \tag{5.14}$$

Four for the dataset typical drop anomalies can be seen in figure 5.2. One form of

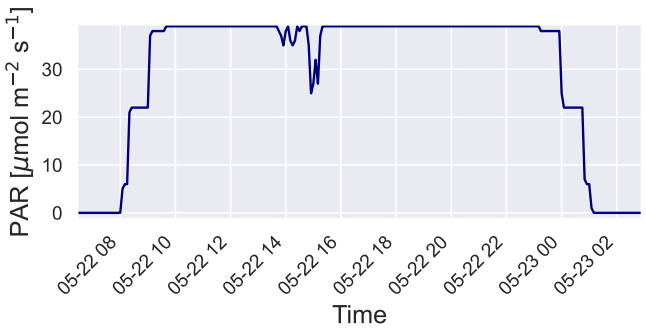


Figure 5.2. – Four drop anomalies with unknown reason occurring on the 22nd of may

shapelet outliers are zero-anomalies. Those can be observed in scenarios where the system or subsystem was shut off due to unforeseen reasons, while sensor data is still being tracked. During operation time of the FEG, at least three full environmental control system shutdowns are recorded in 2018 [ZZ19, p. 5]. zero-anomalies from timestep x_i to x_j with length $l = j - i$ can be synthesized by setting all values to zero. The length l was modulated according to the description above. Figure 5.3 shows a zero-anomaly recorded in the telemetries of a part of the lighting system, due to a shut-off caused by a failure in the temperature control module [BNZ19, 14].

$$\hat{x_t} = 0;\ \forall t \in [a, b],\ b = l + a,\ l \sim exp(\frac{1}{\beta}) \tag{5.15}$$

An enlargement of zero-anomalies are null or missing anomalies [CYPY21, p. 120045]. Missing anomalies are caused by sensor or system failures which lead to missing data recording. They manifest by an empty time value or, for easier programming im-

plementation, a *nan* value at the specific set of data points. The time/frequency response becomes 0. [BNZ19, p. 14].

$$\hat{x_t} = nan;\ \forall t \in [a, b],\ b = l + a,\ l \sim exp(\frac{1}{\beta}) \tag{5.16}$$

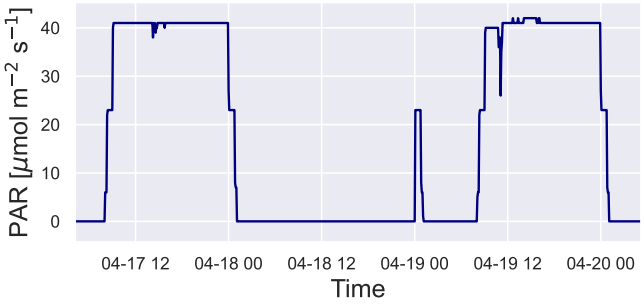


Figure 5.3. – A zero anomaly in parts of the lighting system due to a failure in its temperature controlling module

Noise is a general term for unwanted changes to signals during their capture, storage, transmission or conversion in signal processing. In many cases, noise is due to minor fluctuations in the sensor sensitivity [CYPY21, p. 120046]. For synthetic generation, every value in a sub-sequence of length l from point x_a to x_b is added with scaled white noise $s \sim \mathcal{N}(0, Var(X_i))$ of variance equal to the variance of the variable, ensuring that none additional information will be added to the time series:

$$\hat{x_t} = x_t + 0,1 \cdot s_t;\ \forall t \in [a, b],\ b = l + a,\ l \sim exp(\frac{1}{\beta}),\ s \sim \mathcal{N}(0, Var(X_i)) \tag{5.17}$$

Reasons for noise anomalies can be various. In one case noise was generated by a malfunctioning cooling loop, leading to faltering cooling circulation and consequently to glimmering lighting (figure 5.4) [BNZ19, p. 14].

The last two anomaly types are shift anomalies. Level shift refers to a subsequence of length l where a random, but constant magnitude m is added to or subtracted

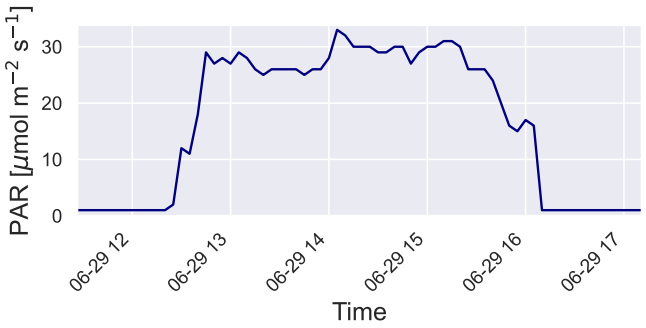


Figure 5.4. – Noise anomaly in a lighting module, caused by malfunctioning cooling system leading to glimmering

from the data [BRGD19, p. 9]. The calculation resembles spike and drop anomalies, but is extended for whole sequences:

$$\hat{x_t} = x_t \pm m \cdot \mu(X_i) \; ; m \sim exp(\frac{1}{\beta}), \; \forall t \in [a, b], \; b = l + a \tag{5.18}$$

Figure 5.5 shows a level shift, among other anomaly types. The reason was a partial failure of the lighting system which needed to be compensated [BNZ19, p. 14]. Time

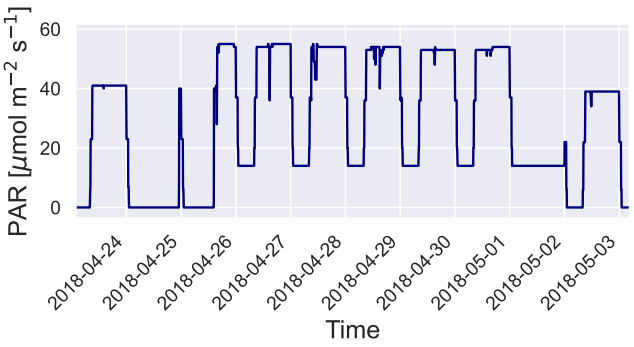


Figure 5.5. – A level shift anomaly. The failure of some lighting system modules had to be compensated by others

shifts are part of contextual anomalies, which do not deviate from the normal range of the time series. However, considering the given context, the data points are out of the expected pattern. Time shifts are frequency-based and therefore these anomalies are hard to detect [CYPY21, p. 120045][BGCML21, 17-19]. In this work, time shift anomalies are defined as anomalous finite frequency-changing behaviour in time

series variables with periodic cycles for whole patterns. For time shift anomalies the patterns were divided in two parts at breaking point p. For example a PAR pattern of length $b - a$, where a equals to the first x_t of the night period and b to day periods period last x_t, is divided by point p in a night period and a day period. The time shift applies on the full day period length of $b - p$ with a shift amount sa. Time shifts do mostly depend on human mistakes rather than on machine errors. A typical example would be a forgotten lighting system restart after maintenance operations. The FEG is operated mostly autonomously without workers, so such mistakes will not be noticed until a distinct period of time has elapsed. Therefore the exponential distribution can not be applied to determine sa for time shifts. Instead the continuous uniform distribution was chosen [EH04, p 92-94]. The boundaries are 1 as minimum shift amount and the length of $p - a$, or in the example of PAR a whole night period, as maximum, to guarantee that day periods cant interfere. With given boundaries, the probability density function of the continuous uniform distribution within boundaries is defined as $f(x) = \frac{1}{(p-a)-1}$. Time shift sequences of length $b-a$ with breaking point p and a shift amount of sa are accordingly expressed as:

$$\hat{x_t} = \begin{cases} x_t & ; a < t < p \\ x_{t-sa} & ; p < t < b + sa,\ sa \sim \mathcal{U}(1, p - a) \end{cases} \tag{5.19}$$

An example of a time shift anomaly can be found in figure 5.6. After an emergency shutdown the daily pattern can be seen to have shifted, visible through the increased gap between two daily periods [BNZ19, p. 15].

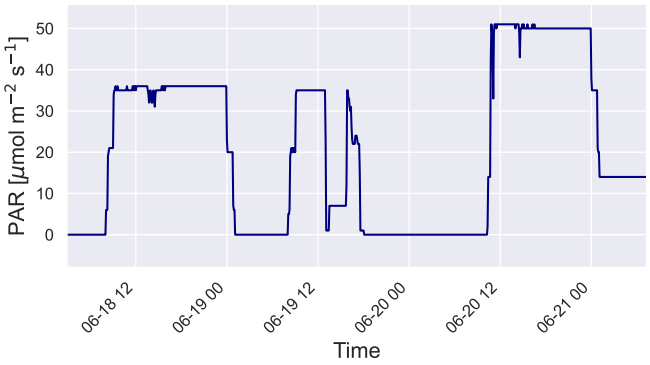


Figure 5.6. – A time shift anomaly among other anomaly types.

5.3. Evaluation measures

The evaluation of anomaly detection is divided in two measure categories, the internal and external measures.

5.3.1. Internal measures

Internal measures account for all evaluation metrics, that do not rely on supervised methods. Quantitative measures for interpreting and validating the results of cluster analysis are a success indicator in unsupervised anomaly detection. Most internal metrics rely on a variant of measuring cluster size or density against distances to other clusters.[ASW15, p 31f]

Silhouette score

A given set of data points $X = x_1, x_2, \ldots, x_n$ will be clustered into k clusters. Let $C = C_1, C_2, \ldots, C_k$ be the set of resulting clusters, where each C_i contains a subset of the data points. The silhouette score for a particular data point x_i can be calculated as follows and takes into account both the within-cluster and between-cluster distances of data points [Rou87, p 54-57]: The average distance a_i between x_i and all other data points in the same cluster x_i is computed as:

$$a_i = \frac{1}{|C_i| - 1} \sum_{x_j \in C_i, x_j \neq x_i} dist(x_i, x_j) \tag{5.20}$$

For each cluster C_j where $j \neq i$, the average distance between x_i and all data points in C_j is received. The cluster with the minimum average distance is chosen, and denote as b_i:

$$b_i = \min_{j \neq i} \frac{1}{|C_j|} \sum_{x_j \in C_j} dist(x_i, x_j) \tag{5.21}$$

For distance calculations, both in a_i and b_i the Silhouette Score mainly utilizes the Euclidean distance function. b_i and a_i finally compose the silhouette score for x_i as:

$$sc_i = \frac{b_i - a_i}{\max(a_i, b_i)} \tag{5.22}$$

The silhouette score ranges from -1 to 1, where a score of 1 indicates that x_i is very well matched to its cluster and poorly matched to neighbouring clusters, a score of -1 indicates the opposite. A score of 0 indicates that x_i is equally distant from both its own cluster and neighbouring clusters. The overall silhouette score for the entire clustering is the average of the silhouette scores for all data points in X. For outline Clusters with a number of $x_i = 1$, a_i can not be computed, therefore sc_i is simply set to 0.

The silhouette score is a measure of clustering quality that takes into account both the cohesion of data points within clusters (measured by a_i) and the separation between clusters (measured by b_i). A higher silhouette score indicates better clustering quality, and it can be used to compare different clustering algorithms or parameter settings.

Davies-Bouldin Index

The goal of the Davies-Bouldin-Score is to define a general cluster separation measure, which allows computation of the average similarity of each cluster with its most similar cluster. Let k again be the number of clusters in a clustering result, and let C_i be the set of data points in the i-th cluster. The Davies-Bouldin index is defined as follows [DB79, p. 224-226]: For each cluster in C_k, the centroid c_i, which is the mean of all data points in a specific C_i, is found by:

$$c_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j \tag{5.23}$$

For each cluster C_i, the average distance between each data point in C_i and the centroid c_i is denoted as d_i:

$$d_i = \frac{1}{|C_i|} \sum_{x_j \in C_i} dist(x_j, c_i) \tag{5.24}$$

Here, $dist(x_j, c_i)$ represents the distance between the data point x_j and the centroid c_i. For simplicity reasons, the computation of d_i is often done using the Euclidean distance.

Per each pair of clusters (C_i, C_j), the similarity between them is defined regarding the distance between the centroids $dist(c_i, c_j)$ of clusters C_i and C_j.

Each cluster maximum similarity $m_{i,j}$ between C_i and all other clusters C_j is found. As distance measure of choice is again mostly the Euclidean distance implemented in literature:

$$m_{i,j} = \frac{d_i + d_j}{dist(c_i, c_j)} \tag{5.25}$$

The value R_i represents the maximum similarity between cluster C_i and all other clusters. A low value of R_i indicates that cluster C_i is well separated from other clusters, while a high value of R_i indicates that cluster C_i is poorly separated from other clusters.

$$R_i = \max_{j \neq i} m_{i,j} \tag{5.26}$$

The final Davies-Bouldin index is the average over all R_i:

$$DB = \frac{1}{k} \sum_{i=1}^{k} R_i \tag{5.27}$$

The Davies-Bouldin index ranges from 0 to ∞, where a lower score indicates better clustering quality. A score of 0 indicates perfectly separated clusters, while a higher score indicates clusters that are less well separated.

The Davies-Bouldin index is useful for comparing different clustering algorithms or parameter settings. It provides a quantitative measure of clustering quality that takes into account both the within-cluster and between-cluster distances of data points, but always in consideration of cluster centres.

Calinski-Harabasz score

The Calinski-Harabasz (CH) score [CH74, p.3-12], also known as the Variance Ratio Criterion, is a metric used to evaluate the quality of a clustering solution based on its ability to form clusters that are well separated from each other. It is defined as the ratio of the between-cluster variance and the within-cluster variance, multiplied by a scaling factor that adjusts for the number of clusters and data points.

Let k be the number of clusters, n be the total number of data points, and C_i be the i-th cluster with n_i data points. Let X be the entire dataset, and $\bar{x}$ be the mean vector of all data points in X. The CH score can be computed as follows:

First, total sum of squares (TSS) is computed, which is a measure of the total variability in the dataset:

$$\text{TSS} = \sum_{x_j \in X} \|x_j - \bar{x}\|^2 \tag{5.28}$$

Here, $\|\cdot\|_2$ denotes the Euclidean norm. If $\bar{x}_i$ denotes the mean vector of all data points in cluster C_i the within-cluster sum of squares (WSS), which is a measure of the variability within each cluster, is calculated:

$$\text{WSS} = \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \bar{x}_i\|^2 \tag{5.29}$$

leading to the between-cluster sum of squares (BSS), a measure of the variability between the clusters:

$$\text{BSS} = \text{TSS} - \text{WSS} \tag{5.30}$$

The CH score is defined as the ratio of BSS and WSS, multiplied by a scaling factor that adjusts for the number of clusters and data points:

$$\text{CH} = \frac{BSS/(k-1)}{WSS/(n-k)} \tag{5.31}$$

The scaling factor in the CH score adjusts for the fact that increasing the number of clusters k tends to increase the BSS and decrease the WSS, which can lead to an inflated CH score. The scaling factor ensures that the CH score is maximized

when the clusters are well separated and the number of clusters is appropriate for the dataset.

In practice, CH score can be used to evaluate different clustering solutions and choose the one that maximizes the score. A higher CH score indicates a better clustering solution, with clusters that are well separated from each other and have low within-cluster variance.

5.3.2. External measures

The proposed internal measures are the main resources in evaluating shapelet clustering and subsequently anomaly detection for the unsupervised EDEN ISS FEG Dataset. They can be used as indications for well-suited hyperparameter optimisation. For more robust results, a more precise anomaly detection score is needed, which can be implemented for the fully labelled synthetic dataset. The external measures fill in that role and account for all metrics that rely on labelled data [ASW15, p 31f]. In the following paragraph are all used external measures explained for the simple binary case.

Accuracy

Let TP be the number of true positives, TN be the number of true negatives, FP be the number of false positives, and FN be the number of false negatives. Accuracy measures the proportion of correctly classified examples and is defined as [Met78, p. 283-286]:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{5.32}$$

Precision

Precision measures the proportion of correctly classified positive examples among all examples classified as positive, and is defined as[FK15, p. 3]:

$$Precision = \frac{TP}{TP + FP} \tag{5.33}$$

Recall

Recall measures the proportion of correctly classified positive examples among all actual positive examples, and is defined as[FK15, p. 3]:

$$Recall = \frac{TP}{TP + FN} \tag{5.34}$$

F1-Score

F1 score is a harmonic mean of precision and recall. Both recall and precision contribute equally [Sas07, p. 1-3]:

$$F1score = \frac{2 \cdot Precision \cdot Recall}{(Precision + Recall)} = \frac{TP}{TP + (FP + FN)/2} \tag{5.35}$$

Average Precision

The trade-off between precision and recall means both of them must be considered simultaneously when comparing and evaluating different detection algorithms. AP (average precision) is a measure of the area under the precision-recall curve and is often used in information retrieval. It is defined as the integral of the precision-recall curve over the range of possible recall values (i.e., from 0 to 1) [EGW+09, p. 314]:

$$AP = \int_0^1 P(R)dR \tag{5.36}$$

where P(R) is the precision at a given recall R. The AP ranges from 0 to 1, with higher values indicating better performance.

5.4. Experimental design

In this chapter, the experimental design is presented to evaluate the derived unsupervised, interpretable anomaly detection method. Two experiments were conducted, one for each dataset. Evaluation for the synthetic dataset is done via internal and external metrics while for the EDEN ISS dataset, only internal measures are applicable due to lack of labels.

5.4.1. General hyperparameters

Hyperparameters account for all parameters that are not changed during a testing session [YS20, p. 1-3]. For all experiments, constant hyperparameters are presented in this chapter. γ for shapelet selection in equation 4.14 was inherited from [LCX+21a, p. 8379] with 0.5, constituting an equal decision between clustering size and between cluster distance. The number of final shapelets for time series transformation can be seen as one of the most important hyperparameters and its hyperparameterspace was set to $f_S \in [5, 10, 20, 30, 40, 50]$ for every different shapelet length, as taken from [LCX+21a, p. 8381]. For hyperparameter optimization grid search [YS20, p. 4] was used. All experiments were repeated at least three times on a Ryzen 5 5600X with 64Gb of RAM.

5.4.2. Synthetic dataset

The synthetic dataset was generated over a span of two years, with data collected every five minutes, resulting in a total of 210240 time steps for the six variables considered. Anomalies for dependent variables were indirectly implemented, resulting

from the calculation with anomalous independent variables. The maximum anomalous rate (MAR) was set to 0.05, or 5%. Two values for the scale parameter of the exponential distribution (see chapter 5.2.3) for all anomalies were given, a height parameter, determining the magnitude of anomalies: $\beta_{height} = 0.4$ and the length parameter for sequential anomalies: $\beta_{length} = 0.00137$, resulting in an average anomaly length of 288 time steps, equivalent to one day period. The likelihood of each anomaly type to occur at a given time point x_t was set to 0.1, with a 30% chance of being a multivariate anomaly. Table 5.1 gives an overview of every variable and the whole dataset about the amount of each anomaly type, as well as the percentage of anomalous time steps. In general, 4.85% of all time steps can be counted as anomalous.

Synthetic dataset anomaly properties

anomalies nr. & perc.	plantmass		PAR		T		RH		VPD		PR		overall
spike	15	0.0071%	11	0.005%	11	0.0052%	10	0.0048%	12	0.0057%	18	0.0086%	**77**
drop	7	0.0033%	7	0.0033%	7	0.0033%	11	0.005%	14	0.0067%	13	0.0062%	**57**
zero	6	0.58%	5	0.64%	6	0.84%	6	0.7%	8	1.08%	9	1.13%	**40**
missing	1	0.08%	1	0,11%	1	0.28%	1	0.13%	2	0.41%	3	0.48%	**9**
noise	8	1.78%	7	1.43%	6	1.14%	6	1.25%	8	1.35%	12	2.25%	**47**
level shift	7	1.38	8	1.43%	6	1.23%	7	1.32%	8	1.36%	11	1.65%	**47**
time shift	0	0	11	1.24%	10	1.23%	0	0	10	1.11%	14	1.57%	**45**
overall	43	3.83%	50	4.86%	47	4.61%	41	3.41%	62	5.31%	80	7.09%	**4.85 %**

Table 5.1. – Number and percentage of anomaly types in the different variables, as well as for the whole synthetic dataset

The length of shapelets is calculated based on time series length. Previous literature suggests that the value of α in equation 4.1 varies between 5% and a maximum of 30%, depending on the specific dataset being analyzed [LCX+21a][BKS+18, p. 961]. In the case of the synthetic dataset, the presence of known periodicities provides valuable information for selecting appropriate window sizes to extract initial shapelets. Four different α values were employed: $\alpha \in [0.000685, 0.00137, 0.002055, 0.055]$, accounting for a timespan of half a day period, one day period, one and a half day period and 40 day periods, respectively. All variables, except plantmass, exhibit a periodicity of one day period, but can also be subcategorised in smaller subparts, for example, heating and cooling periods, therefore the periods of one half and one and a half day periods were also considered. For simplicity, instead of the numerical values, the alphas are named by the time span covered: $\alpha_{12h}, \alpha_{day}, \alpha_{36h}$ and α_{40days}.

5.4.3. EDEN ISS dataset

Even though periodicities for every variable in the FEG dataset are limited by set regulation values within the controlling system [ZBV+15], as explained in chapter 5.2, it is still suggested to perform a time series analysis to obtain meaningful values for α. Most monitored values inside the FEG should show a 24h-cycle, since no other cycle values are defined and the FEG was designed autonomously from environmental influences.

Simple methods to determine linear cycles are the autocorrelation [BJR08, p. 57f.] and partial autocorrelation [BJR08, p. 66f.] plot of the dataset. The autocorrelation function (ACF) is a statistical tool that evaluates the correlation between observations within a time series across different lags. It signifies the extent to which the data points in the time series are correlated with their past values, hence the term "lags". The ACF sequence for a time series X is defined as $\text{Corr}(x_t,\ x_{t-l})$, where $l = 1, 2, ...$

The partial autocorrelation function (PACF) is similar to the ACF, but it focuses solely on the correlation between two observations that cannot be explained by shorter lags. For instance, the partial autocorrelation at lag 3 represents the correlation that is not accounted for by lags 1 and 2. In essence, the partial autocorrelation at each lag captures the unique correlation between two observations after removing the influence of the intermediate correlations.

Figure 5.7 and 5.8 show the exemplary ACF and PACF for one lighting system. As expected, the ACF indicates a strong linear periodicity at multiplies of 288 lags, accounting for a daily cycle. This hypothesis is supported by the PACF, whose maxima are also to be found at these points.

The only variable indicating longer cycles was CO_2, with a higher amplitude at lag 2016, representing one week. Figure 5.8 shows also that some variables inherit correlations of smaller lag than day period length. Therefore, α values are, similar to the synthetic dataset, set so that corresponding shapelets obtain a length of half a day period, one day period, one and a half day period and finally one week: $\alpha \in [0.000691, 0.0009212, 0.001382, 0.00645]$

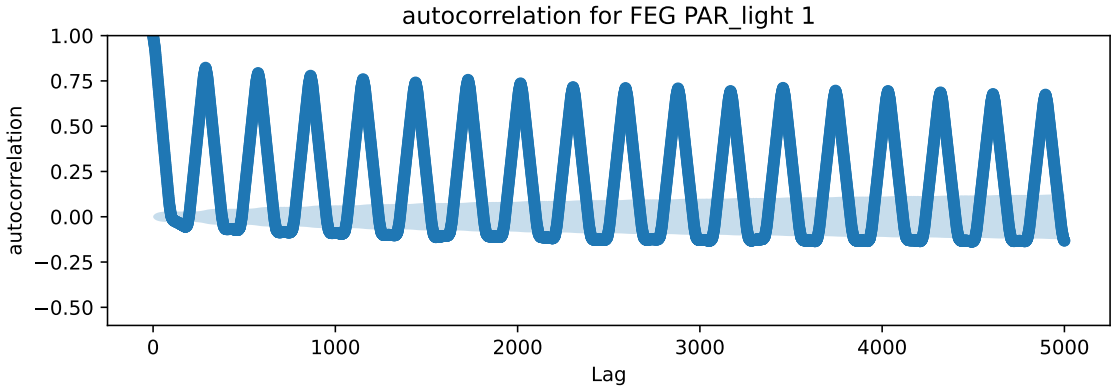


Figure 5.7. – Autocorrelation function of one lighting system: PAR 1. Visible are the maximum values at multiples of 288 lags. The light blue area marks the confidence interval above which values are significant

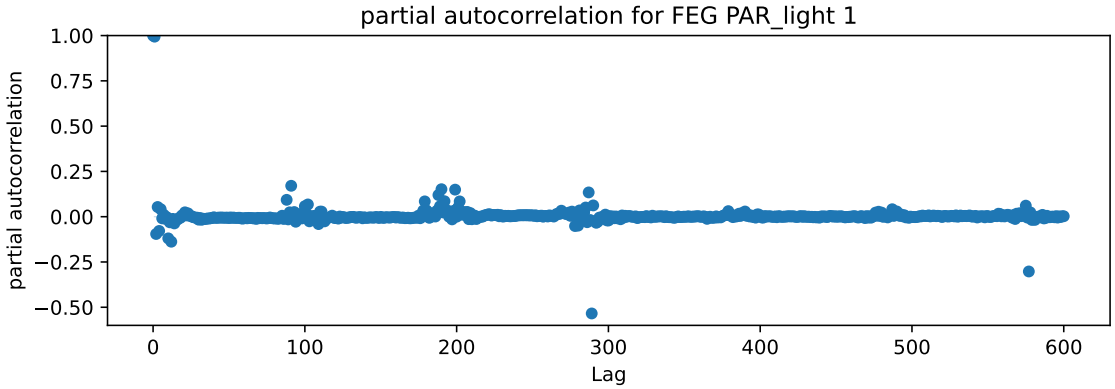


Figure 5.8. – Partial autocorrelation function of one lighting system: PAR 1. In addition to the ACF plot, the PACF shows maxima at multiples of 288 lags

5.4.4. k-Means

The only k-Means specific hyperparameter is the number of clusters k. Values for k were taken from the literature [LCX+21a, p. 8380f.] and expanded for a wider grid search optimisation, for example, $k = 8$ was included, hoping to find one normal class and the seven anomaly types. Values are: $k \in [8, 10, 25, 50, 100, 150, 200, 300]$

5.4.5. DBSCAN

Hyperparameters for DBSCAN clustering are *eps* and *minpts*. For both various heuristics exist [SSE+17, SEKX98]. For many datasets, *minpts* can be kept at the initial value of four, in general [SSE+17, p. 11] or [SEKX98, p. 182] suggest setting it to $minpts = 2 \cdot dataset\ dimension$. Results can improve for datasets that have a lot of noise or that are very large if *minpts* is increased further. Since the definition of "large" datasets is vague, for this work the datasets will be counted as such, to guarantee the best possible hyperparameter coverage. Consequently, the hyperparameterspace is set as $minpts \in [3, 4, 5, 6, 8, 10, 15, 20, 25]$. *eps* can be estimated by sorting the calculated average distances of a shapelet to its *minpts* nearest neighbours in ascending order [EKSX96, p. 5][SSE+17, p. 11]. The optimal value for *eps* can be found at the point of maximum curvature (i.e. where the graph has the greatest slope). This heuristic is called the elbow method, well illustrated in figure 5.9. Shown is the average distance of each shapelet with $\alpha = 0.000685$ to its $minpts = 25$ nearest neighbours in ascending order for the synthetic dataset. Between the shapelet indices 1800 and 2000, the curve exhibits a sudden change in slope, resembling the shape of an elbow, from which the method derives its name. In this case, an approximate value of 0.3 is inferred as the best estimate for *eps*. By visually inspecting all

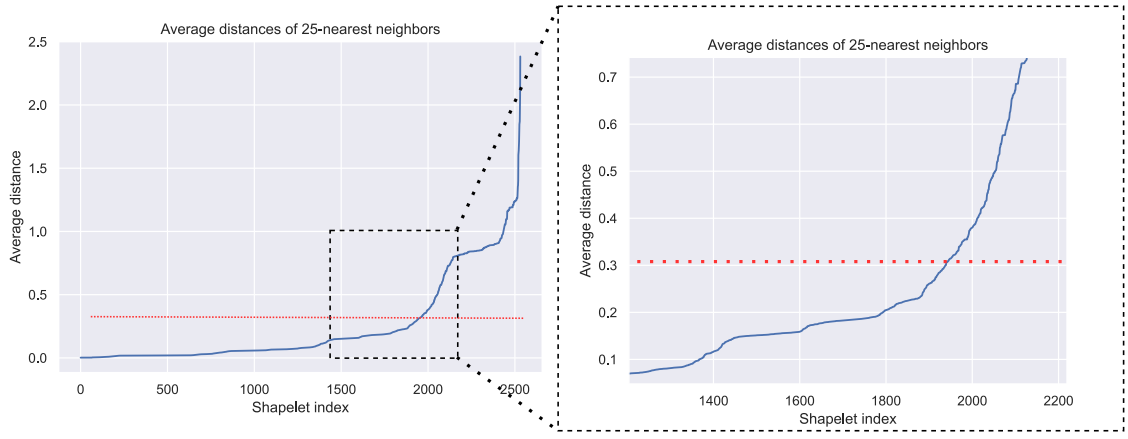


Figure 5.9. – An elbow plot for the synthetic dataset, $\alpha = 0.000685$ and $minpts = 25$, indicating the best possible value of *eps* to be approximately 0.3

elbow plots, for all combinations of α values and *minpts* for both datasets, the hyper-

parameter range of *eps* can be restricted to $eps \in [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8]$. Shapelets that DBSCAN filters as noise are consolidated in their own clusters since they can represent valuable rare anomalous shapelets.

5.4.6. Mdc-CNN

The Mdc-CNN framework was originally tested on the UEA archive [BDL+18], with an average time series length of 1073 time steps. Because examined datasets in this thesis, and therefore shapelet candidates, are significantly longer, the proposed ShapeNet network depth hyperparameter was reduced in size ($10 \rightarrow 5$) while also increasing batch size ($10 \rightarrow 256$) to achieve reasonable computation time[1]. For embedding optimisation, adopted triplet loss variables are $t_{margin} = 0.2$ and $t_{intra} = 1$ [LCX+21a, p. 8379]. The remaining hyperparameters follow the default of the network and network training from [BKK18] and [LCX+21a], for example with number of channels and kernels size of the convolutional network set to 40 and 3 respectively. The Hyperparameterspace for the Mdc-CNN is identical to k-Means experiments, with $k \in [8, 10, 25, 50, 100, 150, 200, 300]$. The number of final shapelets was prior defined depending on the number of alpha values. Since ShapeNet can cluster shapelets of various lengths, this definition is not necessary and is therefore changed to $f_S \in [8, 10, 20, 40, 80, 120, 160, 200]$

[1] This resulted in a reduction of the calculation time of a complete iteration from three weeks to one week

6. Results

In this chapter, the results of the conducted experiments are presented. The results are split into internal and external metrics for the synthetic and EDEN ISS FEG dataset, to asses shapelet discovery and finally the anomaly detection ability. The internal metrics are used to identify meaningful intervals for the hyperparameters used, which are specified to concrete values using F1 score and average precision. Finally, there is an evaluation of all metrics, including the ability to find individual anomaly types. Since the anomaly detection takes place at time point level, all evaluations were calculated likewise. The determination of the F1 score and the average precision is not possible without a label, hence for the EDEN ISS FEG dataset, a more intuitive metric was used for the external assessment, the number of visually identifiable anomaly prototypes. For k-Means and DBSCAN, no deviation was found for any of the internal metrics. The standard deviation for the external metrics, averaged over all iterations, are $1.123e^{-10}$ for k-Means and 0 for DBSCAN. The Mdc-CNN was the only clustering method with noteworthy deviation, therefore mean values across the iterations are presented with associated standard deviations.

6.1. Synthetic dataset

6.1.1. Evaluating shapelet discovery

Internal metrics can give an intuition which of the clustering algorithms can find more representative shapelets, by comparing their ability to define dense and well separated clusters. In general, comparing internal metrics can be seen as the evaluation of the shapelet discovery and shapelet selection, since the selection function is only dependent on clustering results.

k-Means k-Means distinct hyperparameter is the number of clusters k. Figure 6.1 shows the results of the grid search for internal metrics for the hyperparameter k across all α values. The figure is divided into three subplots, one for each metric. To assess shapelet length influence on clustering and representation capability, each subplot consists of a five-line plot, colour coded for the different *alpha* values: $(\alpha_{12h}; blue)$, $(\alpha_{day}; orange)$, $(\alpha_{36h}; green)$ and $(\alpha_{40days}; light\ red)$. For simpler comparison, the mean across all alpha values α_{mean} was included, encoded as dark red dotted line. As mentioned in section 5.3.1 best/worst measure values are Silhouette score: [(1,-1);0], Calinski-Harabsz score: [∞; 0] and for Davies-Bouldin index: [0; ∞].

As one could expect, shapelet clustering is length sensitive, with best results for the shortest shapelets, but figure 6.1 shows no linear correlation between length and clustering metrics because even though the length difference from α_{12h} to α_{day} and α_{day} to α_{36h} remains equal, difference between metric values are not. Over all number of clusters, α_{day} and α_{36h} are much denser and more correlated than α_{12h} and α_{day}, especially visible at the Silhouette score for cluster number $k = 200$. The Silhouette score, as well as the Calinski-Harabsz score, indicate two possible regions for the best number of clusters, $k \leq 50$ and $k \geq 200$. Adding the Davies-Bouldin index puts the statement into perspective, showing, on average, the worst results for $k \leq 50$. Reasons in curve course discrepancy can arise from various influences,

for example, noise, different density regions or close subclusters, that distort the validation [LLX+10].

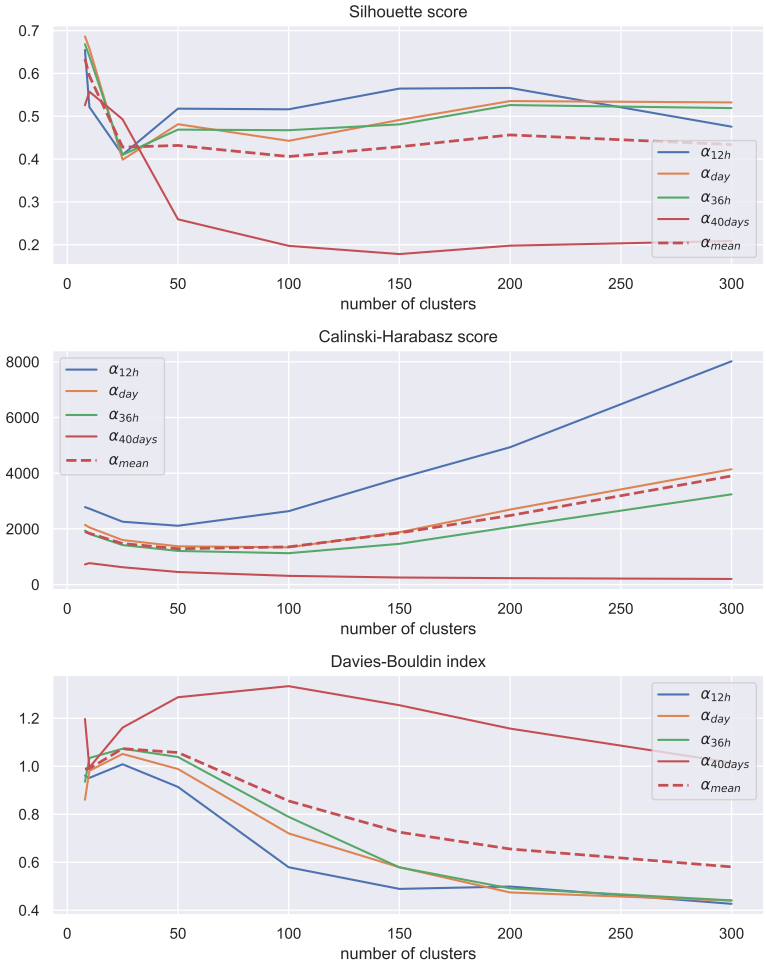


Figure 6.1. – Internal metrics for k-Means shapelet discovery across all α values and averaged. Best/Worst values for metrics are: Silhouette score [(1,-1);0], Calinski-Harabsz [∞; 0] and Davies-Bouldin index [0; ∞]

DBSCAN DBSCAN results for internal metrics are shown in figure 6.2, divided in one column per metric and one row per α value, while the last row shows averaged results. *eps* and *minpts* combinations that lead to just one cluster or otherwise

useless clustering are left blank, apparent by the visible steps. The colours of the rectangles encode the achieved values, ranging from dark blue for lower values to higher values shown in yellow. As can be seen for nearly all α values, for some hyperparameter subspaces *eps* and *minpts* can be increased and decreased correlated and uncorrelated in a certain range without affecting the outcome. For defining hyperparameter space the elbow criterion was used in the experimental design, suggesting small *eps* values for smaller and higher values for longer shapelets. The results confirm the assumption because an expected slight right shift for best hyperparameter regions is visible across ascending shapelet length. Evaluating DBSCAN supports the findings in k-Means, showing better results for shorter shapelets, although differences for the three smallest α are less distinctive. Interestingly, in some cases, the Davies-Bouldin contradicts statements that can be derived from the Silhouette score or Calinski-Harabasz score. This accounts for single rectangles, for example, $eps, minpts = (0.3;25)$, or that α_{40days} is the worst performing shapelet length for the first two. In contrast to k-Means, even the Silhouette and Calinski-Harabasz score can diverge for some hyperparameters, for example, $eps \in [0.2; 0.5]$ and $minpts \in [3; 6]$ for a shapelet length of one day. However, without a more in-depth cluster analysis, which is only possible to a limited extent, no statement can be made as to where this deviation could come from. By focusing on the average best performing regions, assumptions can be made about the hyperparameters *eps* and *minpts*. In general, it seems to be a good choice for setting $minpts \in [20, 25]$ and $eps \in [0.3; 0.6]$, which contradicts previous proposed heuristic *minpts* values of $minpts = 2 \cdot dataset\ dimension$.

Mdc-CNN The main hyperparameter for deep learning based clustering is the number of clusters k, resulting from Mdc-CNN employing k-Means for embedding clustering. Results are shown in figure 6.3. In contrast to k-Means and DBSCAN, deep learning based clustering showed a noticeable variance in results, even with a fixed random initiation both for the network and clustering algorithm. The coloured opaque lines represent the mean across all experiments, transparent shaded areas account for standard deviation. Although the Mdc-CNN relies on k-Means, little similarity can be found in the results and curve progressions. The Silhouette score

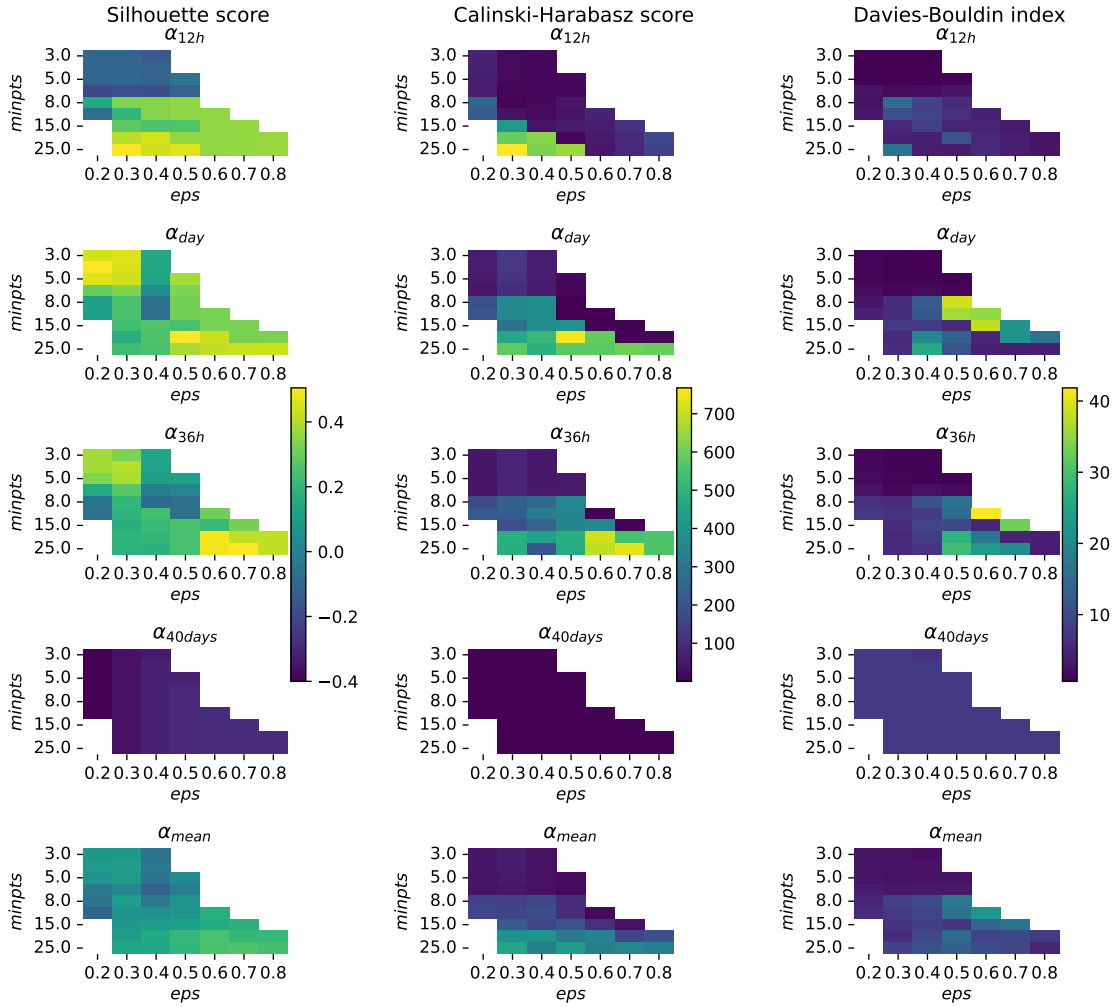


Figure 6.2. – Comparison of internal metrics for all alpha values for DBSCAN. Excluded are all combinations that resulted in only one cluster being formed.

and Davies-Bouldin index show a clear favour of higher cluster numbers. In this regard, the Calinski-Harabasz score falls out of line, with a strong preference for small cluster numbers, which is evident from the almost exponentially, asymptotically decreasing curve strong. Considering that two metrics generally favor higher values and only one lower, the number of clusters was chosen so that the Silhouette score and Davies-Bouldin index are near their maximum with decreasing slope but

before the Calinski-Harabasz score reaches its minimum, which corresponds to a final interval $k \in [100; 200]$.

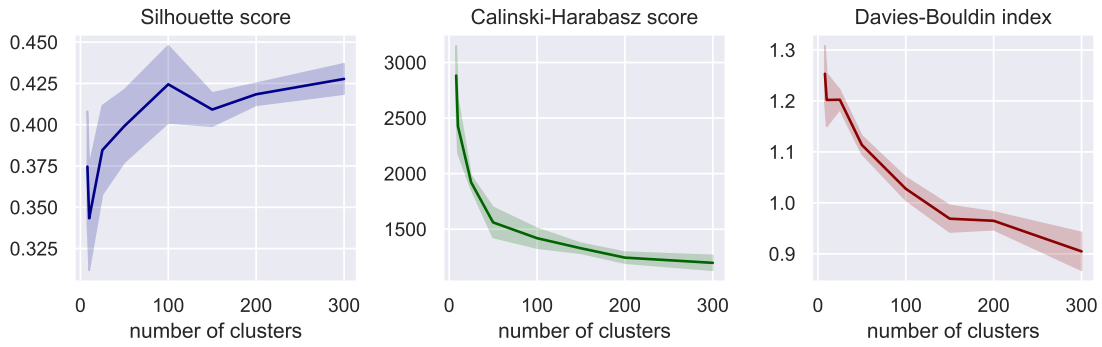


Figure 6.3. – Internal metrics for ShapeNet clustering. Transparent areas represent standard deviation from the mean value shown as opaque lines

For a more comprehensive comparison, an additional survey was conducted, where the Mdc-CNN was restricted to one shapelet length per iteration. This constraint aligns with sequential length-dependent shapelet clustering, used by k-Means. This approach allows for a direct and meaningful evaluation of the Mdc-CNN's performance against shapelet clustering using k-Means, considering the impact of different shapelet lengths on the clustering results. Figure 6.4 shows, similar to figure 6.1 in the k-Means evaluation, internal metrics for each alpha value individually and on average.

On the whole, the curves of both clustering methods are almost identical. Likewise, when comparing the best values for each internal metric, no significant difference is evident, except for the Calinski-Harabasz score. Best values for k-Means were found to be approximately 0.45 for Silhouette score, 4000 for Calinski-Harabasz score and 0.6 for Davies-Bouldin index. Therefore, no advantage of the use of deep learning-based clustering can be determined with a fixed shapelet length. An improved anomaly detection can only result from the fact that Mdc-CNN can process shapelets of different lengths, which can be examined using external metrics.

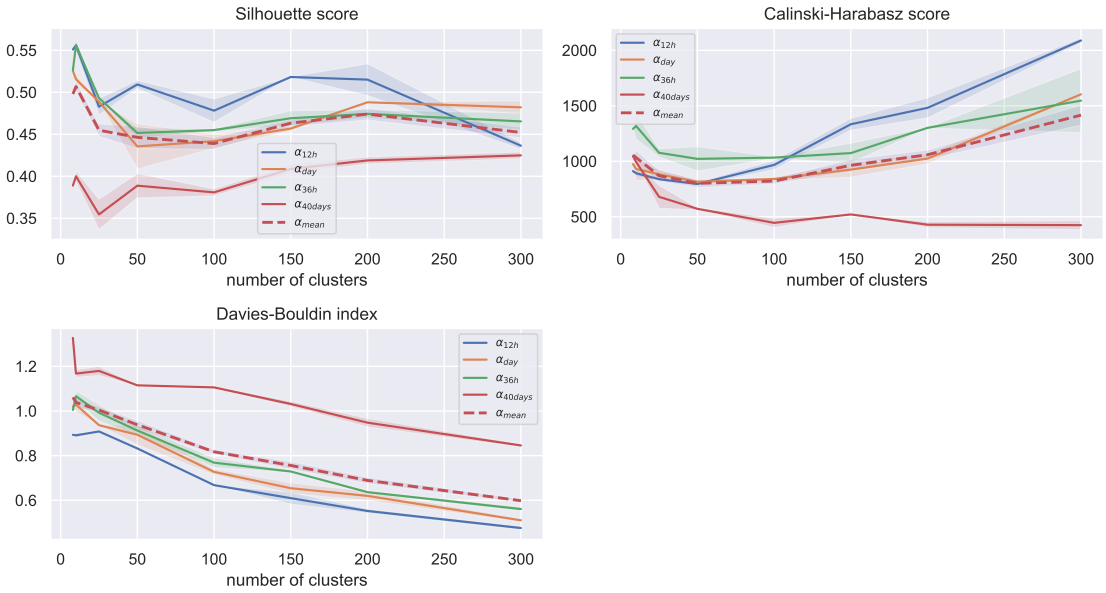


Figure 6.4. – Internal metrics for ShapeNet clustering, for each alpha individually. Transparent areas represent standard deviation from mean value shown as opaque lines

In general, no clustering method showed a clear preference for certain hyperparameter values, as in each study one metric conflicted with the others, highlighting the importance of using different metrics to increase the meaningfulness of the results. In summary, the intervals of the hyperparameters for the clustering methods are $k \geq 200$ for k-Means, $eps \in [0.2; 0.5]$ and $minpts \in [20, 25]$ for DBSCAN and $k \in [100; 200]$ for Mdc-CNN, presented in table 6.1. Before the final comparison

clustering method	preliminary best hyperparameter interval
k-Means	$k \geq 200$
DBSCAN	$eps \in [0.2; 0.5]$; $minpts \in [20, 25]$
Mdc-CNN	$k \in [100; 200]$

Table 6.1. – Preliminary best hyperparameter intervals for all examined clustering methods, based on the findings from internal metrics

regarding the ability to find anomalies, the intervals are further refined using the external metrics to reduce the selection to one combination per method if possible.

6.1.2. Anomaly detection ability

In this section, found hyperparameter intervals for each clustering method are further investigated using the external metrics and finally best found results are compared against each other w.r.t. their anomaly detection ability.

k-Means To finalize hyperparameter choice and evaluate anomaly detection ability, hyperparameters were plotted against the pointwise F1 score and average precision, shown as mean for k-Means in figure 6.5. Findings are colourcoded, ranging from high (blue) to low (yellow), for both figures. In addition to the number of clusters, the number of final shapelets is included in the study space in order to make a final evaluation. Since a higher number of final shapelets than number of clusters is pointless, those combinations were left out of the experiments. It is noticeable that for both values results are below expectation with a low maximum F1 score of approximately 0.19 and average precision of 0.148. The question of what could be the reasons and how anomaly detection can be improved in future work are discussed in section 7.

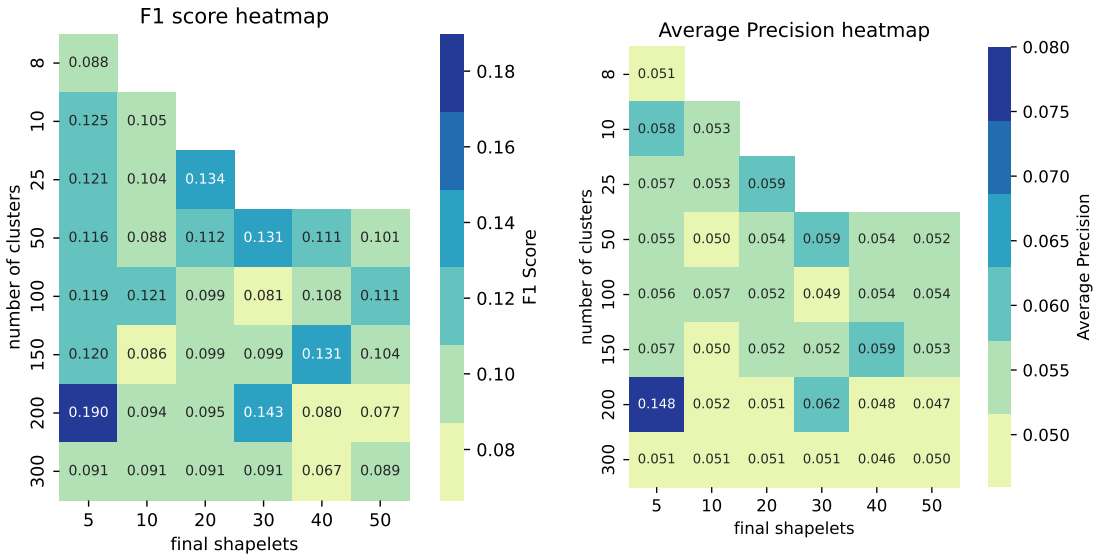


Figure 6.5. – k-Means F1 score and average precision for number of cluster k and number of finally selected shapelets f_S

As a supporting aspect for the validity and reliability of the best hyperparameters, it can be noted that both the F1 score and the average precision locate their optima at congruent positions. Although both metrics are based on recall and precision, both graphs also show that the results are not always congruent, for example for $k \in [10; 150]$ and $f_S = 5$ where F1 suggests local optima, but average precision does not. At first glance, $k = 200$ and $f_S = 5$ look like the optimal combination, which is also supported by the results from the previous chapter $k \geq 200$), but the view shifts on closer examination.

Table 6.2 presents accuracy (for anomalous and normal class combined) and precision, recall, F1 score and average precision for anomalous class.

no. cluster	no. final shapelets	accuracy	precision	recall	F1	average precision
200	5	0.9565	1	0.1049	0.1899	0.1483

Table 6.2. – External metrics for hyperparamters $k = 200$ and $f_S = 5$

The high accuracy and precision are striking. Taking the recall values for all anomaly types individually in table 6.3 into account, it becomes clear how these values are generated. The model has found anomalies either sparsely, or not at all, for example for missing, noise or time shift anomalous time steps. However, almost no false positives were detected. Therefore, it seems, the model was only able to detect few anomalous points and anomaly types, but with high precision. Thus, high F1 and AP results are generated, but the significance and usefulness of these models can be doubted. For further consideration, only hyperparameters are used, which can reliably find all anomalies. A main focus of this work is the question, of whether anomalies are not only found but also interpretable by assignable shapelets. Since shapelets can only be traced back for anomalies that have been identified, the values for the found maximum are not wrong, but give little scope for the investigation of interpretability and are therefore not considered further and are seen as outliers.

no. cluster	no. final shapelets	recall spike	recall drop	recall zero	recall missing	recall noise	recall level shift	recall time shift
200	5	0.0909	0.0677	0.3571	0	0	0.1519	0

Table 6.3. – Recall for all anomaly types for hyperparamters $k = 200$ and $f_S = 5$

As can be seen in both graphs, the next best hyperparameters are located at the same positions. Therefore, the hyperparameter combination of $k = 200$ and $f_S = 30$ with measured values 0.143 for F1 score and 0.062 for average precision is taken as final values, which also corresponds to the interval previously defined by means of internal metrics

DBSCAN Internal metrics for DBSCAN supported intervals of $eps \in [0.2; 0.5]$ and $minpts \in [20, 25]$. The F1 score supports these results only to a limited extent, because although it shows the best results in the range of the interval, other combinations also lead to a similar score, for example, $(eps,\ minpts,\ f_S) = (0.2,\ 10,\ 5)$ in figure 6.6. With the addition of the average precision, the dark rectangles become denser for the entire range of $eps = 0.5$ where also the best value, marked by red borders, can be found. A similar condensation can also be observed in the F1 heatmap, but less pronounced.

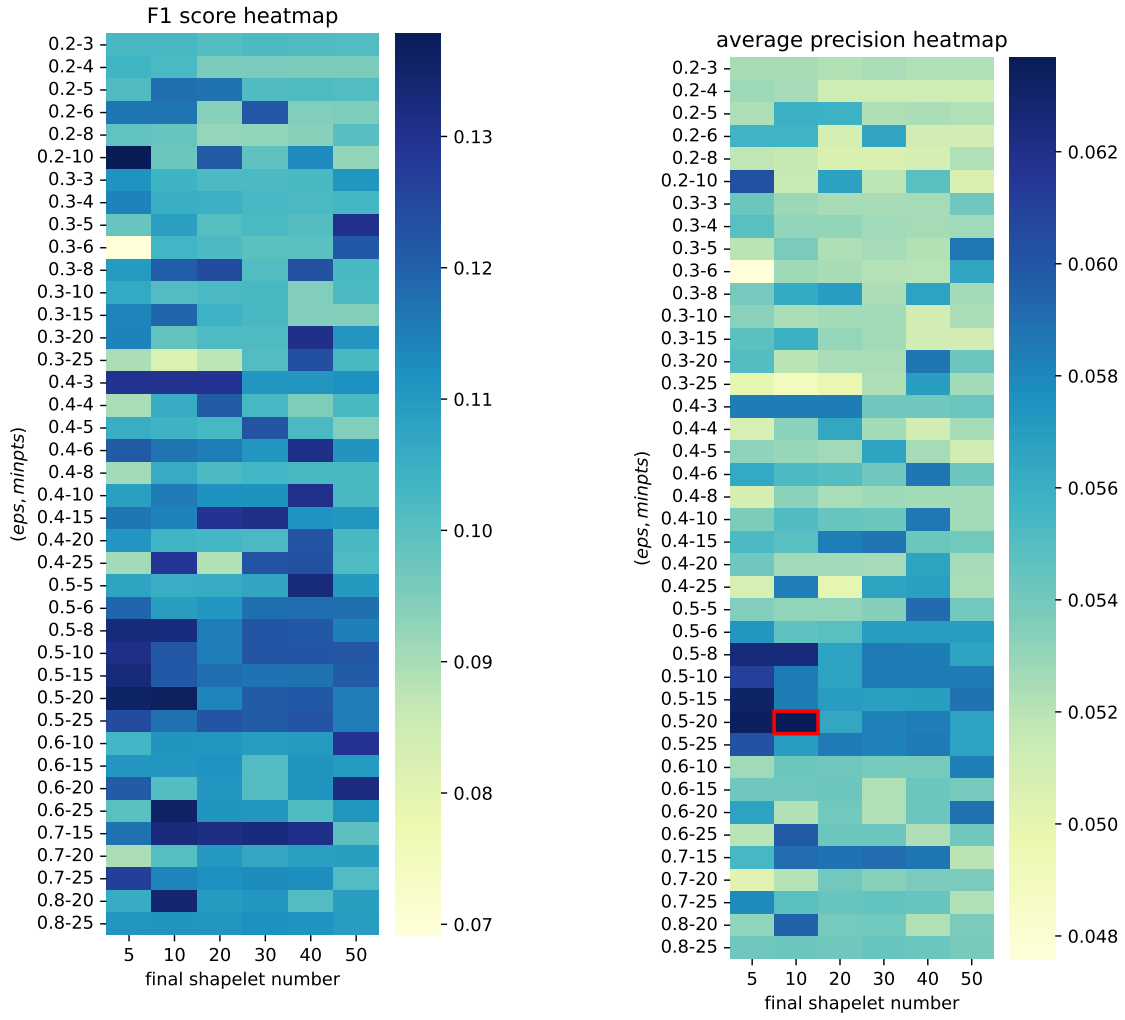


Figure 6.6. – F1 score and average precision for DBSCAN

As can be seen from both figures, there are several possible combinations of hyperparameters, with good results. Since the results for both metrics are not consistent, a high F1 score does not necessarily mean a high average precision and vice versa, the top 10 hyperparameters for both metrics were weighted and ranked. The combined rank consists of equal parts of the F1 score rank and average precision rank. Table 6.4 shows the top three hyperparameter combinations, along with the results and their combined rank. It was not necessary to discard combinations because they did not find all anomaly types, as with k-Means. For final comparison, $(eps,\ mintpst,\ f_S) = (0.5,\ 20,\ 10)$ will be chosen.

combined rank	eps	minpts	no. final shapelets	accuracy	precision	recall	F1	average precision
1	0.5	20	10	0.695	0.079	0.496	0.136	0.0637
2	0.5	20	5	0.693	0.078	0.496	0.135	0.0634
3	0.2	10	5	0.841	0.093	0.260	0.138	0.06

Table 6.4. – External metrics for best DBSCAN hyperparameters and their combined rank, consisting of equal shares of F1 score rank and average precision rank

Mdc-CNN As noted for figure 6.4, clustering using Mdc-CNN for fixed shapelet lengths is not superior to that of pure k-Means. However, there may be an advantage in embedding different lengths to make shapeled discovery and selection independent of alpha. This can be investigated by figure 6.7, showing the number of clusters against the number of final shapelets with associated mean F1 score and mean average precision over all experimental iterations. In contrast to k-Means, f_S refers to all shapelet lengths, so a final number of 8 means that a total of 8 shapelets were selected over all shapelets, whereas in k-Means this was done per alpha value.

Similar to k-Means, and in contrast to DBSCAN, the local and global maxima are consistently located at the same positions for both metrics. The range of the results is similar to that of k-Means, the F1 score ranges from 0.075 to 0.146 as well as the average precision from 0.049 to 0.062. However, it is noticeable that Mdc-

CNN produces significantly more results in the higher range, colourcoded green and cyan.

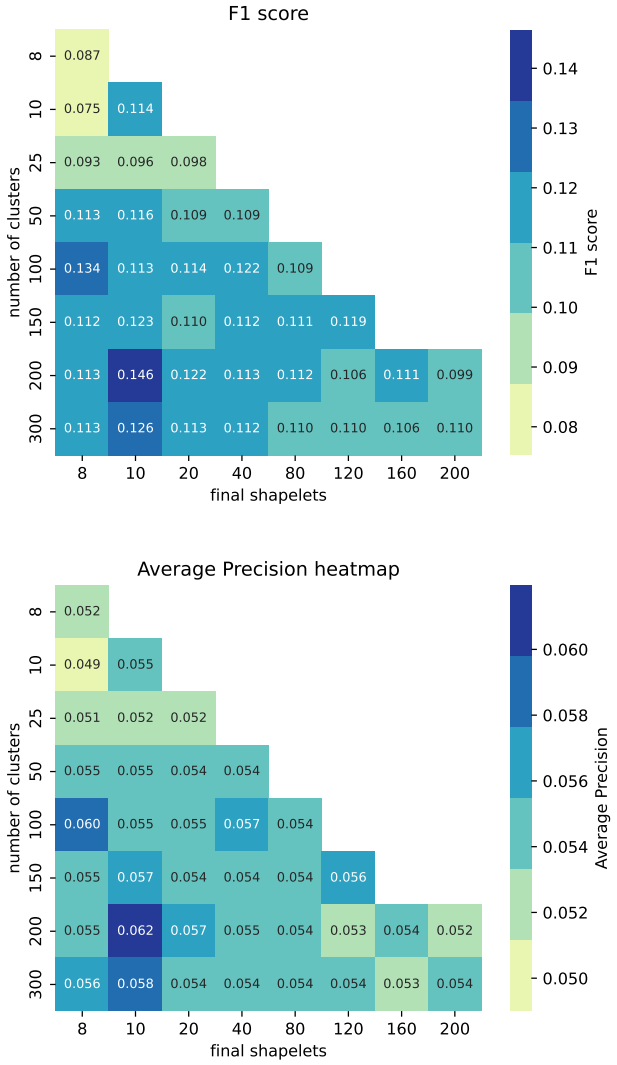


Figure 6.7. – F1 score and average precision for Mdc-CNN. Values represent the mean across all iterations.

This also becomes clear when comparing the mean values of both clustering methods. k-Means achieved on average 0.106 for F1 and 0.055 for average precision, Mdc-CNN on the other hand a 4.7% higher F1 score of 0.111 and equal average precision of

0.055 respectively. With k-Means, however, the considerably small peak at $k = 200$ and $f_S = 5$ should be taken into account, whereby, with elimination, the mean values drop to 0.104 and 0.0532. On average, the Mdc-CNN seems to produce more robust results [1]. As with the other methods, the results were examined in descending order to determine whether all anomaly types were found. The highest result with comprehensive type detection is 0.134±0.02 and 0.06±0.005 with hyperparameters $k = 100$ and $f_S = 8$ falling into the range defined through internal metrics of $k \in [100; 200]$.

Comparison for best hyperparameters After all investigated clustering methods have been examined concerning their results and hyperparameters, results for best found hyperparameters are compared. It is positive to note that in all cases internal and external metrics matched in the selection of hyperparameters, which supports the validity of the results. When focusing on the internal metrics in table 6.5, best performing shapelet discovery was achieved by k-Means, with a cluster number of $k = 200$. On average, k-Means exceeded DBSCAN by 987% and the Mdc-CNN by 46%. k-Means was also the only clustering method, that achieved near-optimal clustering of all shapelets, at least for one metric, the Davies-Bouldin index. Regarding the second metric with clear interval boundaries, the Silhouette score, no method managed to achieve near-optimal results, or at least within the top 50% of the interval, which would correspond to values $\in [\leq -0.5 \vee \geq 0.5]$. Worst performing clustering method, was by far DBSCAN. However, the results must be viewed with a certain degree of caution. Reasons for DBSCANS performance are discussed in chapter 7. Due to its exclusive utilization of shapelets of all lengths, the Mdc-CNN produces results that can only be compared to a limited extent with other methods. A final decision can only be made by comparing the external metrics.

Findings in the evaluation of anomaly detection do not align with the results for shapelet discovery. Table 6.6 summarizes results for external metrics and the recall for each anomaly type individually. The Mdc-CNN showed best results for accuracy of normal and anomalous class combined and precision, DBSCAN for recall

[1] A.6 shows a histogram comparison for both scores and both clustering methods. The shift to the right of the distribution curve for MDC-CNN is visible

	Hyper-parameter	Silhouette score	Calinski-Harabaszs score	Davies-Bouldin index
k-Means	$k = 200$	**0.456**	**2480**	**0.66**
DBSCAN	$eps = 0.5$ $minpts = 20$	0.202	313	14.81
Mdc-CNN	$k = 100$	0.424±0.023	1419±88	1.03±0.022

Table 6.5. – Comparison of internal metrics for best found hyperparameters for all clustering methods

and average precision and k-Means had the highest F1 score. Thus, it cannot be confirmed that a better clustering result, as measured by the internal metrics, necessarily leads to better anomaly detection. When looking at the individual types of anomalies, the situation is more uniform, with DBSCAN consistently achieving the highest values.

external metrics	hyper-parameter	Accuracy both classes	precision	recall	F1 score	average precision		
k-Means	$k = 200$ $f_S = 30$	0.849	0.0986	0.26	**0.143**	0.0615		
DBSCAN	$eps = 0.5$ $minpts = 20$ $f_S = 10$	0.695	0.0791	**0.496**	0.136	**0.0637**		
Mdc-CNN	$k = 100$ $f_S = 8$	**0.897 ±0.0097**	**0.115 ±0.023**	0.162 ±0.013	0.134 ±0.02	0.0596 ±0.005		
recall for anomalies		recal spike	recall drop	recall zero	recall missing	recall noise	recall level shift	recall time shift
k-Means	$k = 200$ $f_S = 30$	0.208	0.220	0.603	0.073	0.174	0.312	0.046
DBSCAN	$eps = 0.5$ $minpts = 20$ $f_S = 10$	**0.416**	**0.441**	**0.831**	**0.393**	**0.442**	**0.457**	**0.358**
Mdc-CNN	$k = 100$ $f_S = 8$	0.136 ±0.0065	0.136 ±0.017	0.448 ±0.021	0.037 ±0.037	0.098 ±0.045	0.168 ±0.064	0.023 ±0.023

Table 6.6. – Comparison of external metrics for best found hyperparameters for all clustering methods

Recall, a measure of the algorithm's ability to correctly identify positive instances (in this case, anomalies), implies that DBSCAN successfully captured a significant portion of the true anomalies present in the dataset. On the other hand, the accuracy score for both classes and precision for the anomalous class was the lowest, suggesting that the DBSCAN algorithm generated a considerable number of false positive predictions.

6.1.3. Interpretability and anomaly prototypes

A primarily motive of shapelet-based anomaly detection was the drastically improved interpretability. Unfortunately, interpretability can neither be assessed by qualitative or quantitative methods, since it is a sorely subjective perception. Therefore the idea of anomaly prototypes was introduced, shapelets, that are visually definable as anomalous, with lowest distance to anomalous points or sequences in the time series and which are the maximum representative of their cluster. Anomaly prototypes should enable the user to quickly classify anomaly types. For their identification, the following consideration is limited to the best clustering method.

No clustering method showed a distinct superior anomaly detection ability. However, the identification of the anomaly prototypes strongly depends on the results of the shapelet detection and selection, in which k-Means achieved the best results, as the internal metrics make a statement about the representational capability of a centroid. With the given hyperparameter combination, a number of 30 final shapelets per α were selected. These overall 120 final shapelets were further investigated.

The distance-based nature of the shapelet transformation allows for the detection of anomaly prototypes. These prototypes correspond to shapelets with the smallest distance to the time series points identified as anomalies. A shapelet closest to an anomalous time point or sequence is a potential prototype. The occurrence of a shapelet having the smallest distance to multiple anomalies diminishes the number of potential anomaly prototypes. Out of the final 120 shapelets, 46 were found to be closest to at least one anomalous location. All 46 shapelets can be found in Appendix A.7 and A.8. 50% of those shapelets had a length of 40 days, as can be seen from table 6.7, and 45.7% of all shapelets inherited an anomaly. However, none among them were shorter than 40 days. Below, three examples illustrate the process of determining anomaly prototypes. Initially, two examples of a normal class are presented, followed by an explanation of an anomalous shapelet. For a better overview, two variables are used whose detected anomalies can be completely assigned to the shown shapelets.

length	α_{12h}	α_{day}	α_{36h}	α_{40days}	percentage of all [%]
anomalous	0	0	0	21	45.7
normal	7	5	11	2	54.3
percentage of all [%]	15.2	10.9	23.9	50	100

Table 6.7. – Breakdown of final shapelets, with the least distance to time points and subsequences detected as anomalies, by length and anomalous or normal class

Normal class The selection of the two normal shapelets was based on the observation that, despite not being anomaly prototypes themselves, they can be distinctly assigned to one anomaly type, almost exclusively. Figure 6.8 shows the normalized relative humidity and all anomalous points, marked red, that were closest to shapelet no. 42 plotted in figure 6.9. The six anomalies just below the 0.2 line are zero anomalies, as shown in the enlarged section. This corresponds to an identification rate of 100% for this type of anomaly, as can be noted from table 5.1. The third anomaly marked from the right is a drop anomaly.

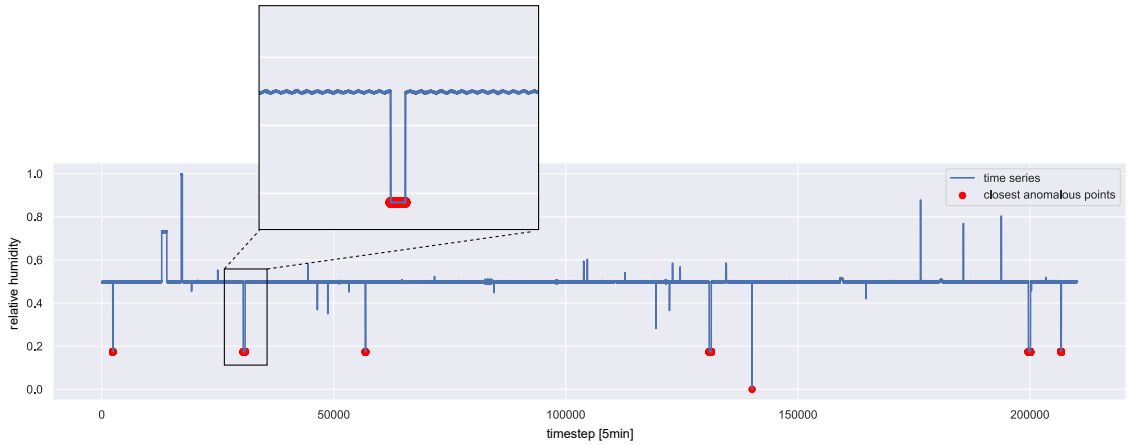


Figure 6.8. – Closest anomalous points to shapelet in figure 6.9. All six zero anomalies were identified, additionally to a drop

The shapelet itself can be classified as normal class, as evident from the comparison using A.3 and A.4. To get an overview of the associated cluster, the shapelets from

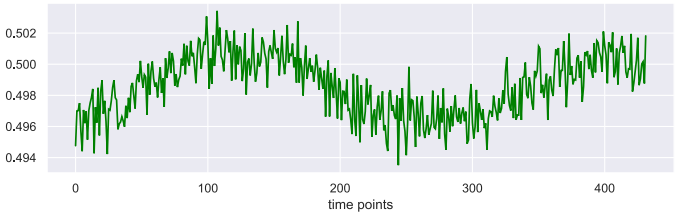


Figure 6.9. – Normal shapelet for relative humidity associated to zero anomalies, extracted using α_{36h}

the same cluster were reviewed. A section of this can be seen in figure 6.10, where the nearest four shapelets are shown. Every shapelet corresponds to a normal class, making it possible to assess the shapelet as a representative of that particular normal class. Thus, the identification of the anomalies was done by means of a large distance to the normal case. It is worth noting that a certain type of anomaly, except for a drop, can be assigned to a normal class of shapelets.

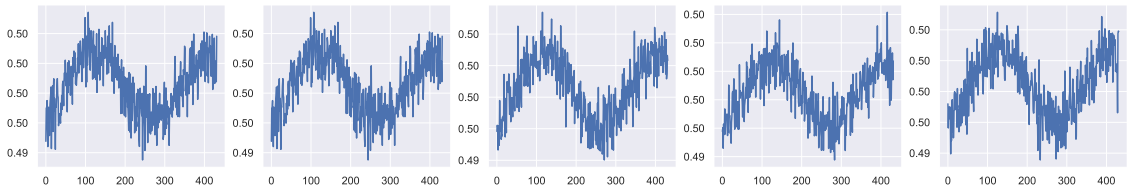


Figure 6.10. – The shapelet in figure 6.9 (leftmost) and four nearest shapelets within the same cluster

Another illustration of anomaly detection using only normal class shapelets can be observed for shapelet no. 43 in Figure 6.11. This time, the shapelet successfully detects six out of seven level shifts in the relative humidity time series. To classify the associated cluster, the associated shapelets were again examined. 6.13 shows a section of the cluster created identically to 6.10, with the final shapelet in the first position. Both the maximum representative shapelet and the cluster are to be classified as normal. The two shapelets above were the only ones returned as final shapelets for relative humidity and α_{36h}. The reason for this can be found in the shapelet discovery. Figure 6.14 is an estimation of a graph network for all shapelets for α_{36h} extracted from relative humidity. Every dot represents a shapelet and the estimated visual distance represents the actual distance of the shapelets

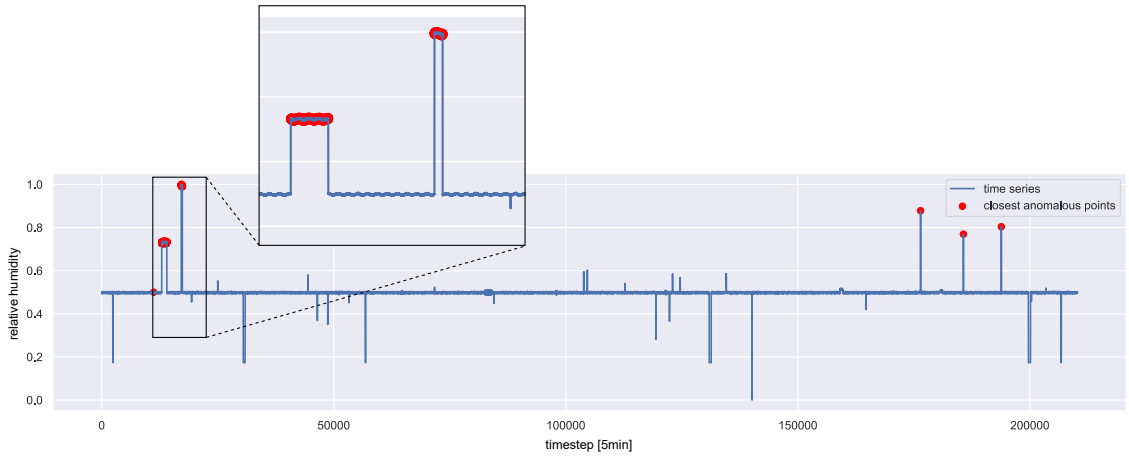


Figure 6.11. – Closest anomalous points to shapelet in figure 6.12. Six out of seven level shift anomalies were identified in the relative humidity data

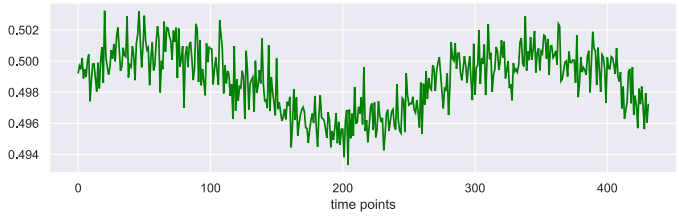


Figure 6.12. – Normal shapelet for relative humidity associated to level shift anomalies, extracted using α_{36h}

using Euclidean distance in shapelet discovery. As visible, no single shapelet or group can be identified as an outlier, resulting in only two clusters, colour coded in green and blue. The reason for two clusters with no discernible cluster boundary comes from the given number of clusters k for k-Means. Although no anomaly prototype emerged for relative humidity from the final shapelets, since they consist only of normal class, it is nevertheless possible within certain limits to assign two types of anomalies to one shapelet each.

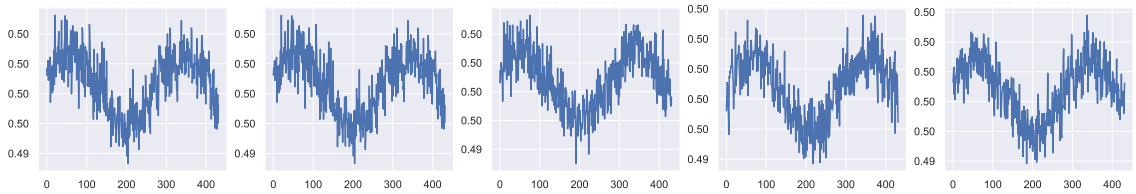


Figure 6.13. – The shapelet in figure 6.12 (leftmost) and four nearest shapelets within the same cluster

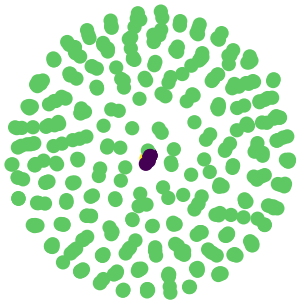


Figure 6.14. – Graphnetwork estimation for all shapelets for α_{36h} extracted from relative humidity. Dots represent the shapelets, visual distance estimates Euclidean distance and colour code accounts for corresponding cluster

Anomalous class Of all the found anomalous final shapelets, shapelet no. 46 shown in figure 6.15 will be used. It was chosen based on its ability to demonstrate the interpretability of shapelets in the context of the study. The shapelet was extracted from the photosynthetic rate, using α_{40days}. Visible is the anomaly around timestep 2000 for a period of over 500 timesteps. The marked closest anomaly is visible in figure 6.16. In contrast to above, only one anomaly was identified in the entire PR time series, but this one is of particular interest.

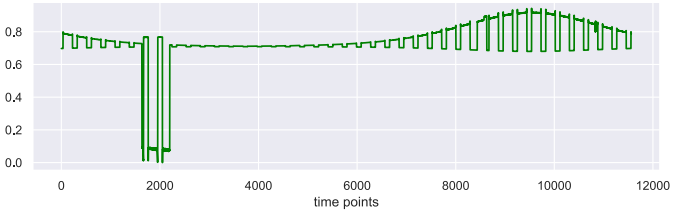


Figure 6.15. – Anomalous shapelet for photosynthetic rate associated to level shift anomalies, extracted using α_{40days}

PR is a dependent variable and the found shapelet allows an unambiguous identification, which of the influencing variables, plantmass, temperature and photosynthetic active radiation PAR, initially inherited an anomaly and of what type. The anomaly seems to be a level shift, since no other sequential anomaly would allow a similar behaviour. Noise is not visible at the subsequence and a zero anomaly cannot simultaneously increase the PR for the day and decrease it for the night. A zero anomaly in the temperature or PAR would automatically set the PR, according to formula 5.10, to zero, or negative, since only the dark respiration R_d remains. Also, without a known formula, a failure of cooling or illumination systems, in reality, would not increase the PR, as described in section 5.2.2. A zero anomaly for the plantmass would set the total subsequence to zero. By exclusion, only a level shift remains as possibility.

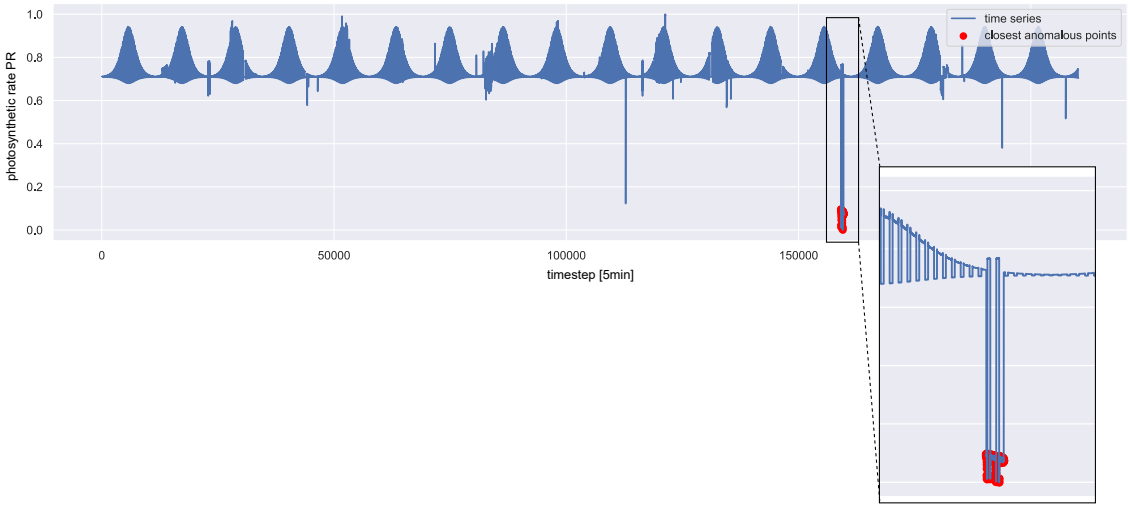


Figure 6.16. – Closest anomalous points to shapelet in figure 6.15. Shown is the only found level shift for the photosynthetic rate

The identification, in which of the three variables the anomaly occurred, is also possible. Temperature only affects the photosynthetic rate at day, R_d is not affected, so temperature alone cant explain the anomaly. PAR does also not effect R_d. The only variable that affects PR both for day period and night period is plant mass. Yet, the plant mass, or more precisely the growth, is a linear factor, it cannot explain the different amplitude increases for both periods. Thus, monocausal explanations

are out of the question. It therefore seems to be a multivariate level shift anomaly. In fact, there is a multivariate level shift anomaly at this time steps. Thus, the interpretability of the shapelet allows to make statements about the rest of the data set and to increase the number of identified anomalies from one to at least 5, since plantmass, PAR, temperature and therefore also VPD must be affected.

The shapelet is not an anomaly prototype since the cluster consists of various shapelets with different types of anomalies, as shown in figure 6.17. Still, being the only shapelet for this time series, the cluster contains valuable information, because it inherits all types of anomalies.

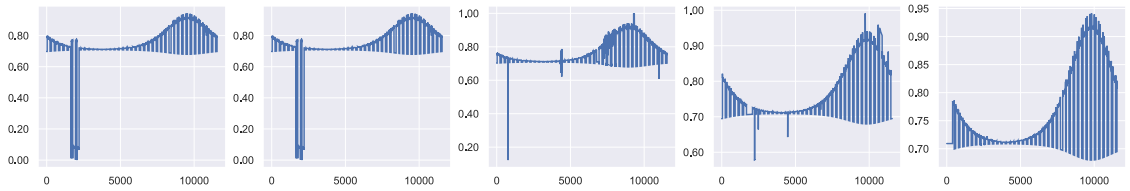


Figure 6.17. – The shapelet in figure 6.15 (leftmost) and four nearest shapelets within the same cluster

6.2. Eden ISS FEG dataset

The subsequent chapter provides a summary of the outcomes obtained from the EDEN ISS FEG dataset. Unlike the artificial dataset, this particular dataset lacks labels. Therefore, when choosing the best clustering technique and searching for anomaly prototypes, additionally to internal ones, a different metric is employed compared to the external metrics used before.

6.2.1. Hyperparameter optimisation

k-Means By inspecting the internal metrics for the FEG dataset in figure 6.18, colourcoded for each alpha value and on average, it becomes clear that k-Means as clustering method deteriorates for larger k. The only metric favouring larger cluster

numbers is the centroid-based Davies-Bouldin index, but only by a small extent of 1.16 against 1.02 for $k = 300$.

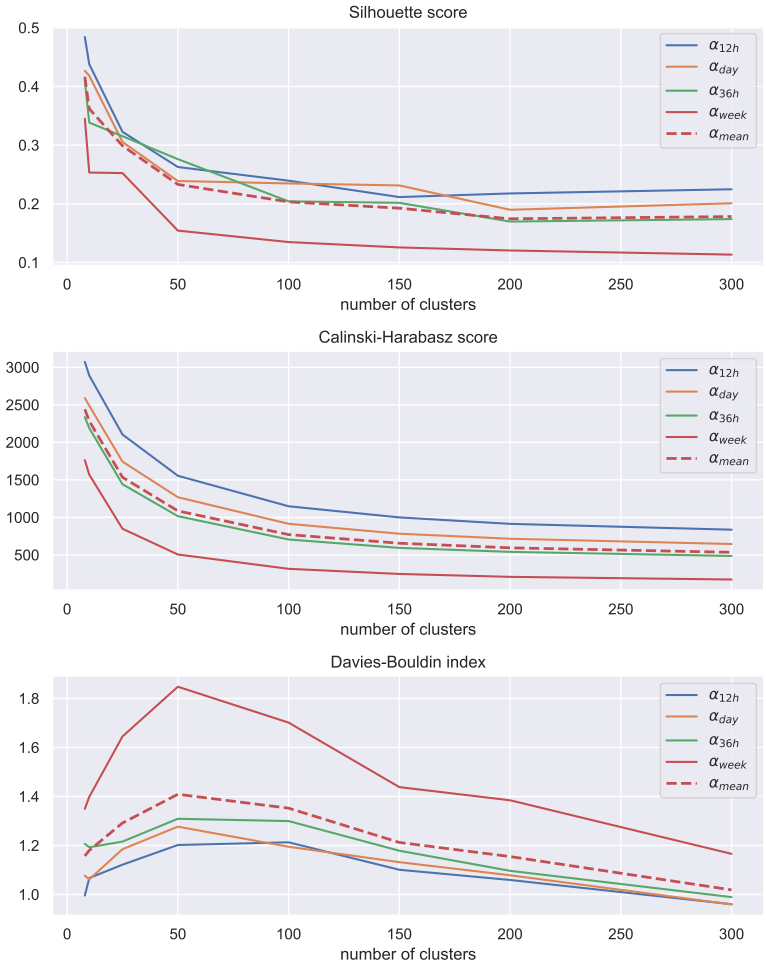


Figure 6.18. – Internal metrics for k-Means shapelet discovery across all α values and averaged.

As final number of clusters is $k = 8$ chosen, with a Silhouette score of 0.42, a Calinski-Harabasz score of 2440 and the second smallest Davies-Bouldin index of 1.16 on average. This limits the number of possible anomalies that can be detected by direct anomaly prototypes since the number of selectable shapelets is restricted to a maximum of $4 \cdot 8 = 32$. When comparing results for the synthetic and FEG dataset, scores for $\alpha_{12h}, \alpha_{day} and \alpha_{36h}$ are denser, especially for the Calinski-Harabsz

score and Davies-Bouldin index, which indicates a smaller influence of the shapelet length, but only to a limited extent, as alpha week still has a much more distant course. Curve progress remains, except the Calinski-Harabasz, nearly identical.

DBSCAN DBSCAN shows clear favoured hyperparameter intervals for almost every metric and shapelet length, recognizable by the bright regions for the Silhouette score and Calinski-Harabasz score in figure 6.19 and no particular outliers can be identified in any of the metrics.

The Davies-Bouldin index is significantly less selective for the first three alpha values, which is reflected by the large dark blue coloured areas. Among all the shapelet lengths, α_{week} stands out as the only one with a shift of the optimal range towards smaller regions at the edges concerning $minpts$. It should also be noted that the areas of the best results are almost congruent for all metrics and all shapelet lengths, i.e. the metrics match, which is not the case with k-Means and Mdc-CNN. A movement of the optimal region through the hyperparameter space depending on the shapelet length, like for the synthetic dataset in figure 6.2, is only slightly noticeable for the first two metrics and first three alpha values. The determination of the final combination of hyperparameters is heavily influenced by which metric is given more emphasis, as the results exhibit ambiguity. Employing a simple majority decision, the best hyperparameters are $eps, minpts = 0.8, 20$, with the corresponding scores of 0.185(1. SH), 904(1. CH) and 3.48(7. DBI) for α_{mean}, since they deliver better results in 2 of the three metrics. The numbers in brackets represent the placement of the hyperparameters in the respective metric.

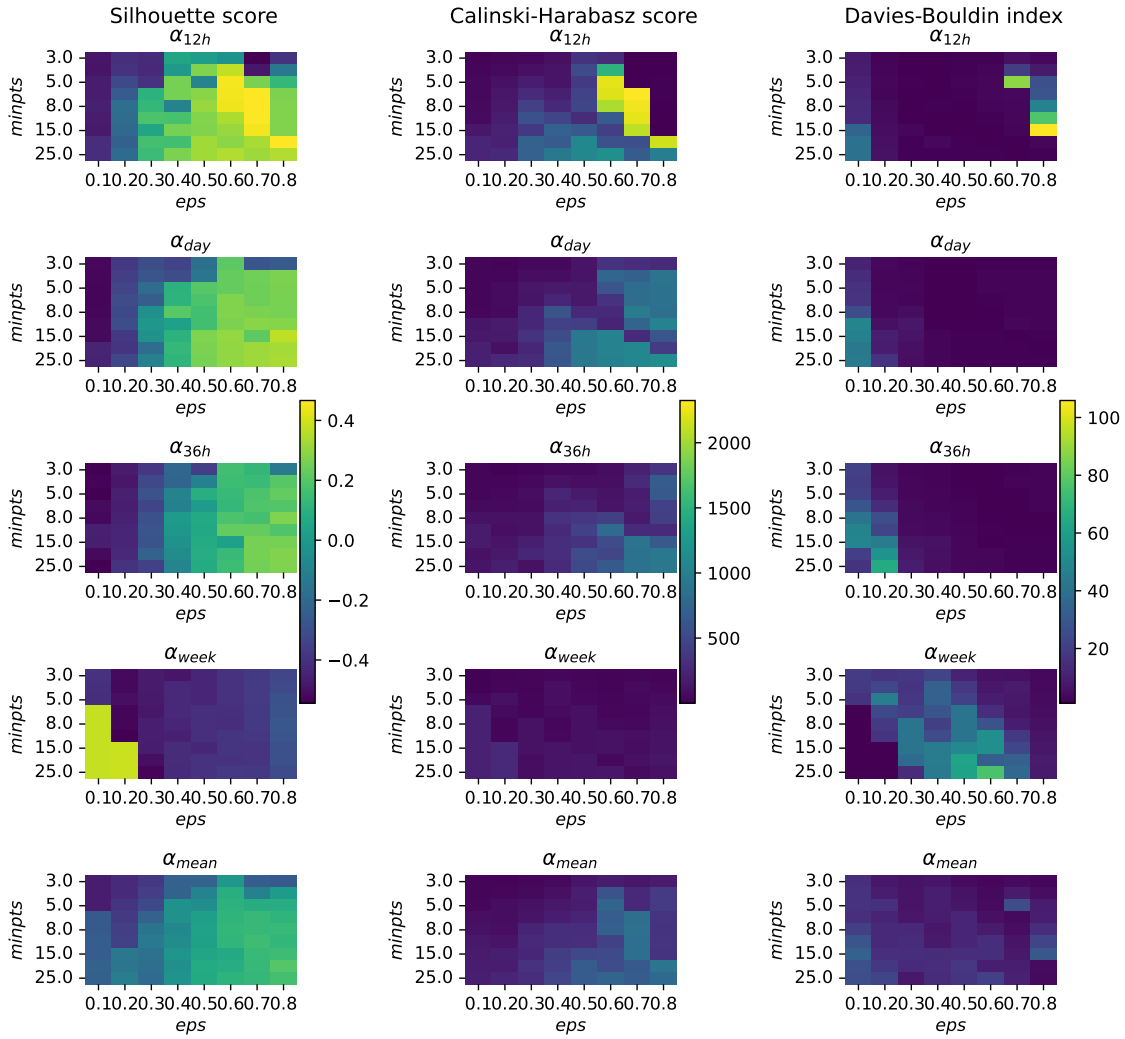


Figure 6.19. – Comparison of internal metrics for all alpha values for DBSCAN

Mdc-CNN For the Mdc-CNN all three internal metrics indicate a small number of clusters for best results, as can be seen in figure 6.20. Transparent areas represent the standard deviations of each metric. The Davies-Bouldin index is the only one with a significant improvement after deteriorating as the cluster count increases, however, not to a degree that would reach or surpass the previous optimum. Both the Silhouette score and the Calinski-Harabasz score start at or near their maximum

values and steadily decrease from there. With an increasing number of clusters, the clusters overlap, expressed by the silhouette score of almost 0. Values near 0 indicate, that a one-to-one assignment to a cluster is no longer possible for a single shapelet.

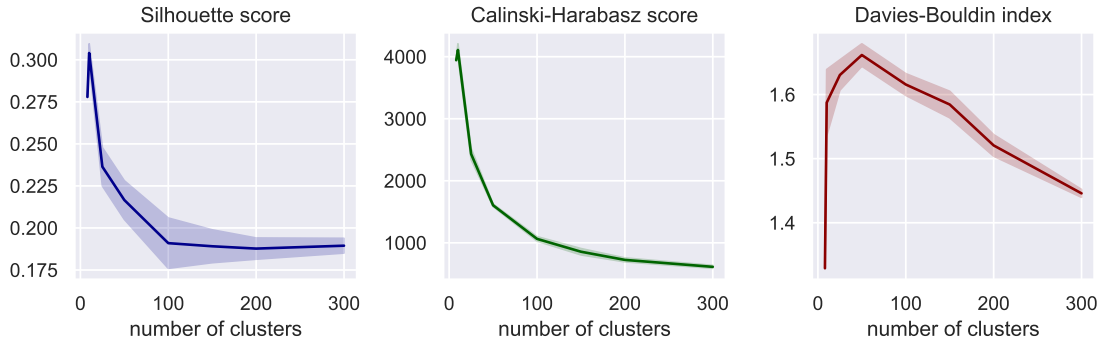


Figure 6.20. – Internal metrics for ShapeNet clustering. Transparent areas represent standard deviation from mean value shown as opaque lines

There are two possible best values for cluster number k, depending on which metric the emphasises lies, see table 6.8. This is problematic since it reduces the number

no. clusters k	Silhouette score	Calinski-Harabaszs score	Davies-Bouldin index
8	0.283±0.0057(2.)	4057±114 (2.)	1.39±0.067 (1.)
10	0.304±0.0055 (1.)	4109±99 (1.)	1.59±0.052 (2.)

Table 6.8. – Best two number of clusters k for Mdc-CNN. The number in brackets stands for the ranking in the individual metrics

of available shapelets for anomaly detection to maximally 10. Therefore, Mdc-CNN was evaluated for each alpha individually, under the assumption that the reduced complexity caused by the fixed shapelets length for each clustering iteration leads to better clustering results. Figure 6.21 shows the results for all alpha values and on average, again with standard deviation. Reducing complexity did not meet expectations, since the results remain nearly equal. The curve progression only displays changes for the Davies Bouldin index, surpassing its local optima at a lower level of cluster numbers. The silhouette score and Calinski-Harabasz score do not exhibit such variations. Both these metrics consistently yield the best results for the smallest possible number of clusters.

Even if the internal metrics are not compared with each other until later, it can be seen in comparison to k-Means that with the clustering of fixed shapelet lengths, the curves for individual alphas are significantly denser than with k-Means.

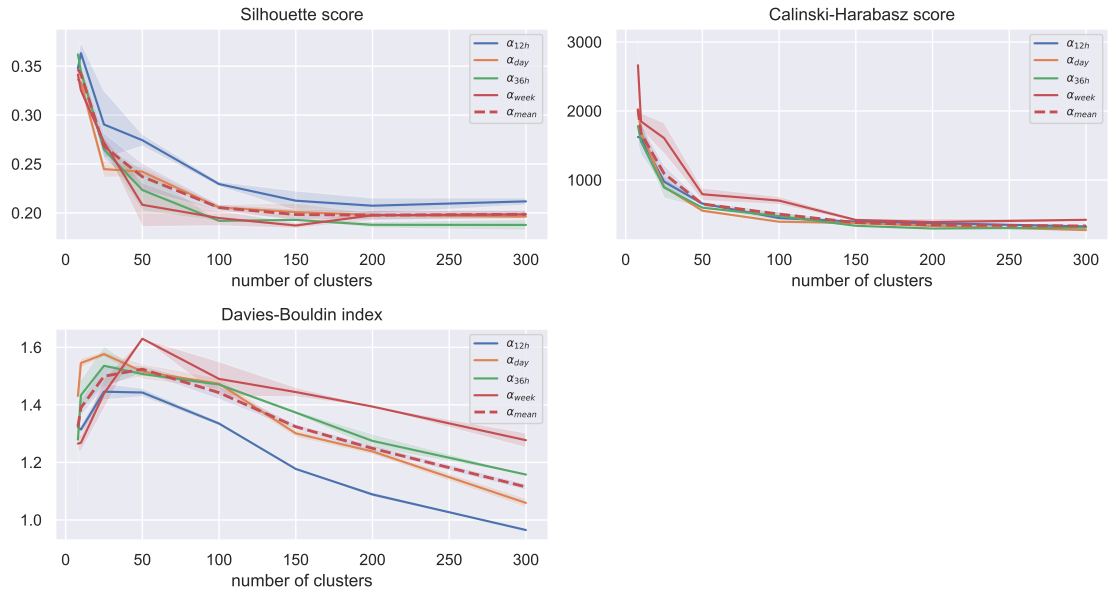


Figure 6.21. – Internal metrics for ShapeNet clustering, for each alpha individually. Transparent areas represent standard deviation from mean value shown as opaque lines

Comparison for best hyperparameters Table 6.9 shows the final comparison of all three clustering methods. Similar to the findings for the synthetic dataset, k-Means achieved the best results. The Mdc-CNN was the only clustering method to surpass it for one metric, namely for the Calisnki-Harabasz score. The internal metrics for DBSCAN do not necessarily indicate poorer clustering, as the discussion in chapter 7 will show, but has its reasons in the implementation.

In a direct comparison with the synthetic data set, it is noticeable that the curves are similar, but the results for individual hyperparameters are much less obvious, which makes it more difficult to define clear interval boundaries. Even though the EDEN ISS FEG dataset is much more complex than the synthetic one, the final internal

results are quite comparable, since the range of values has only shifted slightly. Some methods even seem to cluster more compactly and with clearer boundaries, e.g. DBSCAN, which achieved results of Silhouette = 0.202, Clainski-Harabasz = 313 and Davies-Bouldin = 14.81 in the synthetic experiments. All in all, the ranking of the methods has not changed, although the simultaneous clustering of all shapelets by Mdc-CNN should also be considered here.

	hyper-parameter	Silhouette score	Calinski-Harabaszs score	Davies-Bouldin index
k-Means	$k = 8$	**0.417**	2440	**1.16**
DBSCAN	$eps = 0.8$ $minpts = 20$	0.185	904	3.48
Mdc-CNN	$k = 10$	0.304±0.0055	**4109±99**	1.59±0.052

Table 6.9. – Comparison of internal metrics for best found hyperparameters for all clustering methods

6.2.2. Interpretability and anomaly protoypes

Since no clustering method was found to be the best and labels are missing to evaluate anomaly detection, the selection of the method to identify anomaly prototypes and evaluate their interpretability is based on the number of possible candidates for anomaly prototypes, more precisely the number of final shapelets. As centroid-based clustering methods, k-Means and Mdc-CNN explicitly specify this number as part of the clustering, namely 4*8=32 and 10, respectively. DBSCAN is the only clustering method that does not explicitly specify f_S. In order to keep the set of selectable final shapelets as large as possible, DBSCAN is examined with the hyperparameters $eps = 0.8$, $minpts = 20$ and $f_S = 50$, which theoretically corresponds to 200 possible anomaly prototypes. Still, the maximum number of final shapelets corresponds to the number of clusters.

A total of 108 potential anomaly prototypes were found, more than would be possible using k-Means or Mdc-CNN. All 108 shapelets are shown in A.9, A.10 and A.11. In total, 52.07% of the FEG dataset was labelled as anomalous. This value initially appears to be elevated, but the following examples show that it is quite realistic.

Each of the 108 shapelets was sorted according to its length and examined to determine whether it should be classified as normal or abnormal. The assessment was based on the normal case defined in chapter 5.1 for each variable. The boundary is not always clear-cut; small deviations (5%) are not directly classified as abnormal unless they are classifiable anomalies, such as a spike. Shapelets that nevertheless cannot be clearly assigned to a category are recorded as "undefinable". Table 6.10 shows the result of the analysis. The proportion of anomalous shapelets has remained

length	α_{12h}	α_{day}	α_{36h}	α_{week}	percentage of all [%]
anomalous	9	12	13	21	50.9
normal	15	10	7	4	33.3
undefinable	3	3	6	5	15.8
percentage of all [%]	25	23.15	24.07	27.77	100

Table 6.10. – Breakdown of final shapelets, with the least distance to time points and subsequences detected as anomalies, by length and anomalous or normal class for the FEG dataset

nearly equal, but in contrast to the synthetic data set, they are almost uniformly distributed across all shapelet lengths. Out of the 108 potential shapelets, only 17 (15.8%) could not be assigned to a class.

Normal class The first case demonstrates the identification of different anomalies as deviations from the normal pattern. In the excerpt from the second illumination time series in figure 6.22, it is evident that all illumination periods deviating from the normal pattern during the day have been labelled as anomalies. This classification holds true as long as these periods achieve the specified illuminance. Additionally, the course of an illumination period, as specified in the FEG documentation, is distinguishable and represented by the second rectangle from the left. This example also highlights the inherent ambiguity in classifying shapelets into distinct categories. Despite the shapelet, shown in figure 6.23, roughly aligning with the normal progression, a drop anomaly is apparent in the right part. It demonstrates that the classification of shapelets into two classes is not always straightforward or definitive.

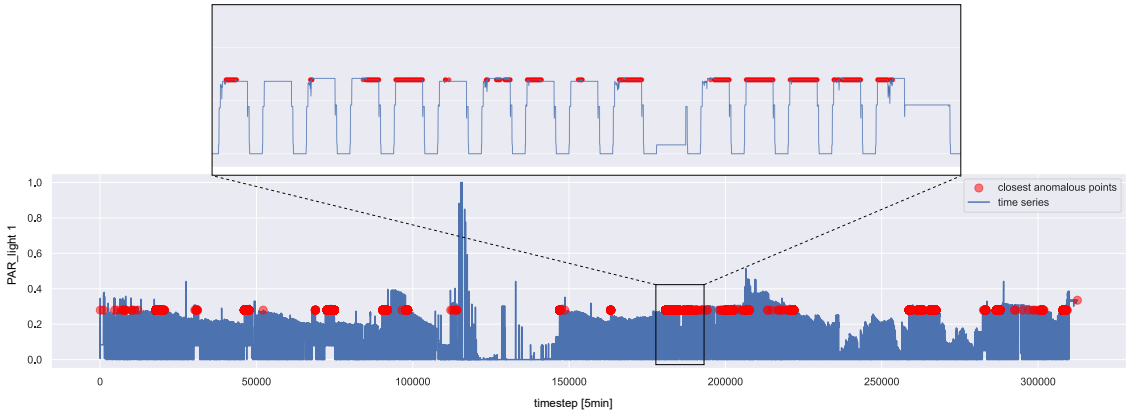


Figure 6.22. – Closest anomalous points to shapelet no. 31 in 6.23. Anomaly types identified are mainly small spikes, drops and level shifts

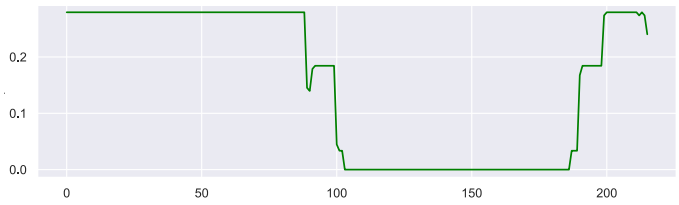


Figure 6.23. – Normal shapelet no. 31 for PAR light 1, extracted using α_{12h}

Even if the shapelet individually would be considered as normal class only without the right part, it was nevertheless assigned to this class, because first of all the part important for the anomaly detection corresponds to the normal class. Secondly, the cluster, excerptwise depicted in 6.24, contains mainly normal shapelets, which show the target course of the maximum illumination according to specification. A second example, for identified level shifts in temperature time series, can be found in A.12

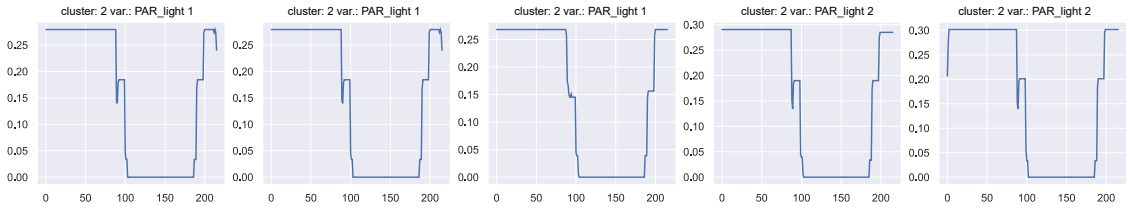


Figure 6.24. – The shapelet no. 31 in 6.9 (leftmost) and four nearest shapelets within the same cluster

Anomalous class The initial case example where an anomaly prototype could be identified, relates to level shift anomalies in the second illumination system. As depicted in the plot excerpt referenced as 6.25, these instances involve illumination durations that do not align with the specified intensity. In the majority of cases, the illumination fell short by less than 20% of the required level. Notably, the maximum lighting intensity during these periods exhibits flickering, indicating a system failure rather than a deliberate reduction. The anomaly prototype candidate is shown in figure 6.26. This image displays the normal case on the left, while the level shift candidate is evident on the right. This is particularly helpful for interpretability, as case discrimination for points marked as anomaly can be done with one image.

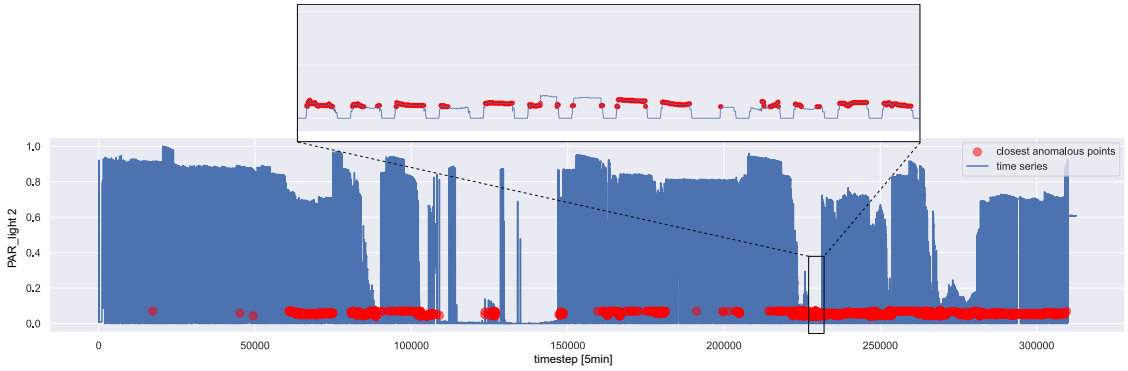


Figure 6.25. – Closest anomalous points to shapelet no. 108 in 6.26. Detected anomalies are mainly level shifts, as visible in the excerpt

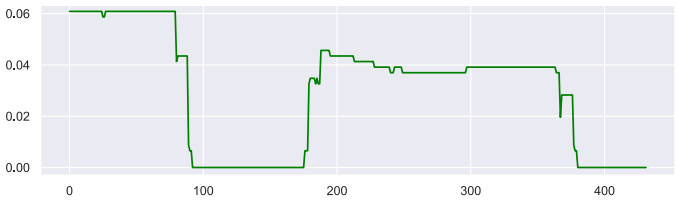


Figure 6.26. – Anomalous shapelet no. 108 for PAR light 2, extracted using α_{36h}

All shapelets contained in the associated cluster correspond to a level shift anomaly, although the manifestation in each case can have significant differences, as shown in figure 6.27. It is also apparent that the distinction between level shift and noise

anomaly is often ambiguous, which can be explained by flickering of the lighting system in the event of a power loss.

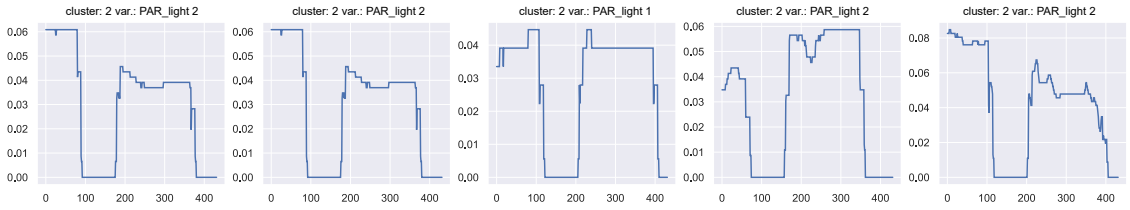


Figure 6.27. – The shapelet no. 108 in 6.26 (leftmost) and four nearest shapelets within the same cluster

The following example illustrates the challenge of precisely categorizing the specific type of anomaly in question. Within illumination system 2, multiple instances of spike anomalies were detected. These anomalies manifested as transient surges in illumination, followed by immediate cessation, as depicted in the magnified segment in plot 6.28 The spike anomalies were caused by the short illumination pulses. The short lighting pulses reached on average 10% to 20% of the actual illuminance. The normal intensity level can be seen in the snippet at the right edge.

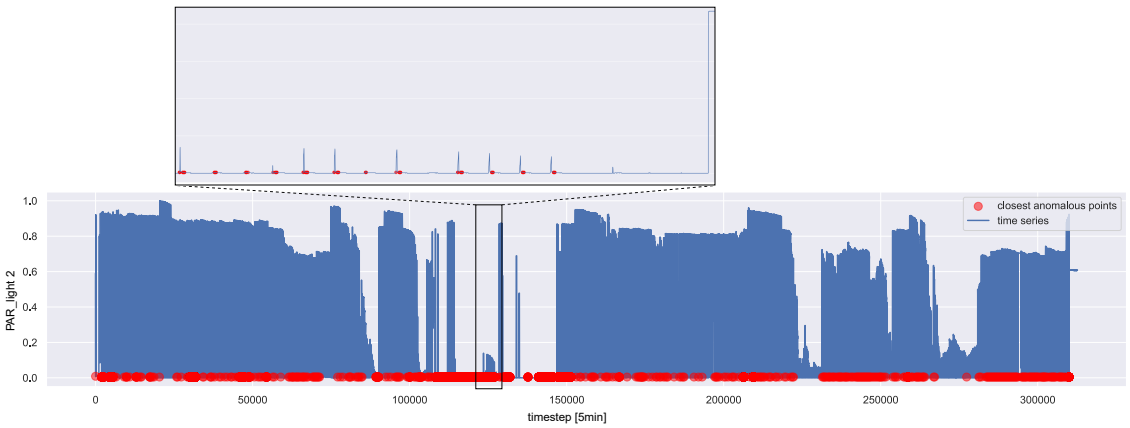


Figure 6.28. – Closest anomalous points to shapelet no. 106 in 6.26. Detected anomalies are mainly spikes, as visible in the excerpt

The associated closest shapelet is shown in figure 6.29. While visually resembling a spike, the anomaly needs to be formally evaluated as a level shift due to its dura-

tion spanning multiple sequence points. Similarly, the associated cluster, depicted in figure 6.30, exhibits a comparable behaviour. Consequently, it is reasonable to designate the anomaly prototype as an impulse anomaly, since these can extend over short periods of time, which corresponds to its behaviour in the time series. An impulse is characterized by a sudden rise, with impulse width $P_w \to 0$ and is thus very closely related to a spike.

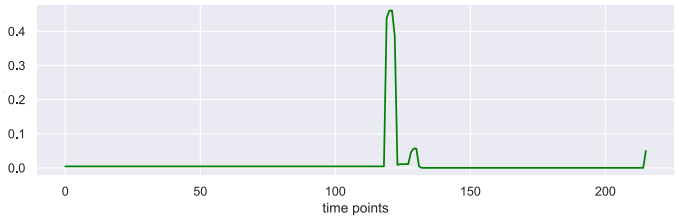


Figure 6.29. – Anomalous shapelet no. 106 for PAR light 2, extracted using α_{14h}

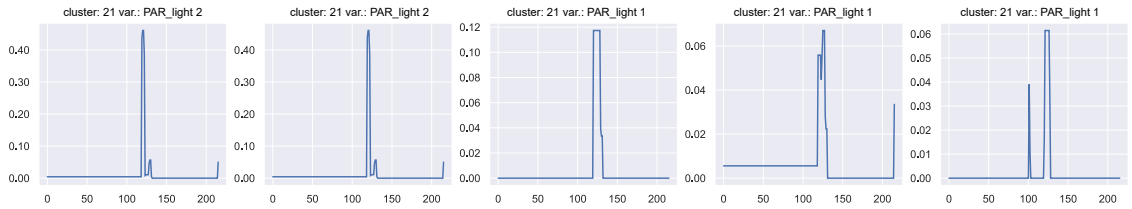


Figure 6.30. – The shapelet no. 106 in 6.26 (leftmost) and four nearest shapelets within the same cluster

7. Discussion

The following section delves into a comprehensive discussion of the results obtained from all conducted experiments, with a primary focus on evaluating the effectiveness and performance of the proposed workflow. The workflow, derived from existing time series classification frameworks, was specifically designed to detect anomalies and anomaly types in an unsupervised manner while providing interpretable anomaly prototypes.

One aspect was to evaluate different clustering methods and their use for shapelet selection. A surprising result was the worse performance of the Mdc-CNN in comparison to k-Means and DBSCAN, concerning both internal and external metrics. Although it was initially assumed that the simultaneous clustering of different shapelet lengths could have caused a negative effect, it was shown that clustering of individual alpha values did not yield any advantage over k-means. The only difference between k-means and the Mdc-CNN for a single α value is the embedding. Identifying the reasons behind this behaviour is a challenge due to the Mdc-CNN, which is the underperforming component, functioning as a black box model. The lack of transparency in its inner workings makes it difficult to gain a comprehensive understanding of its mechanisms and potential sources of underperformance. One aspect could be the missing hyperparameter optimizaiton because initial hyperparameter values were taken as is. Another aspect could be the length of examined shapelets. The Mdc-CNN was originally proposed for the UEA Archive for time series classification and it showed superior classification results for 14 out of 30 datasets [LCX+21a, p. 8381]. On closer inspection, it is noticeable that the average length of the time series, for which the Mdc-CNN yielded the best results, is 853, which is smaller than the average length of all time series in the dataset, of 1073. This leads to two assumptions: first, the length of the time series used was significantly greater than

that of all the data sets in the UEA Archive. At the same time, the length ratio of the used shapelets was significantly larger. For alpha values between 0.05 and 0.3, as used in literature [BKS+18, p.], the maximum length ratio is 1:6. The shapelets used in this work can reach a length ratio of 1:80, which could be problematic for the Mdc-CNN. Second, it can be assumed that the Mdc-CNN, in its current form, is only suitable for shorter time series, which is evident from the results by [LCX+21a] and since the results for individual α values were also below those of the other clustering methods.

By comparing k-means and DBSCAN it was observable that DBSCAN shows weaker and less distinctive clustering results than k-means. This can be explained by two possible reasons: Firstly, k-means favours convex clusters [MCT21, p. 176] that tend to have a smaller within-cluster distance, which is used for most internal metrics. Another factor is the influence of shapelets labelled as noise or outliers. DBSCAN was developed for more robust clustering by separating noise points that can not be allocated to a core point. As mentioned in section 4.1.2, all noise labelled shapelets are consolidated in a distinct noise cluster, to find far-distant, rare anomaly types. Density of this noise cluster can expected to be low since it is in the nature of noise points to be widely distributed and in less dense regions.

It was possible, with the presented workflow, to identify all anomaly types and, mainly for the FEG dataset, also to find anomaly prototypes, but the anomaly detection generally remained below expectations. Consequently, the results of the external metrics are further discussed to gain deeper insights into the reasons behind this discrepancy.

All external metrics were calculated on a point-by-point basis to evaluate the most accurate detection possible. In order to assess that this method of calculation does not have a significant impact on the error values and distort the result, the results were also calculated using less precise sequencewise metrics. A detection is considered a true positive if the detection occurred somewhere within the anomaly. Correctly identifying a single point of the anomaly is sufficient for marking the entire anomaly as found. False negatives are all anomalies (not their sequence points) that were not found. False positives are all labelled subsequences that do not overlap with an

anomaly. The results obtained using this approach for the best clustering method found, k-means, are presented in the table below. 7.1. The sequencewise calculation improves the results, but only to a certain level. Thus, the methodology of the metric calculation has an impact, but other sources of error still need to be identified for a final evaluation. Furthermore, the methodology of the calculation is always a question of trade-off, for the highest possible precision (pointwise) or whether it is sufficient, e.g. in application, to issue a warning as soon as an anomaly is detected, no matter how much of it (sequencewise).

method	recall	precision	F1 score	average precision
pointwise	0.26	0.0986	0.143	0.0615
sequencewise	0.24	0.159	0.192	0.129

Table 7.1. – Pointwise and sequencewise calculated external metrics for best k-means

For all three clustering methods, shapelet clustering itself can initially be ruled out as a source of error, justified by the good internal metrics, some of which came close to the best possible results, e.g. close to 0 for the Davies-Bouldin index. The next decisive step is the multivariate time series transformation, which was used in this form for the first time. The final anomaly detection is distance-based, which is why anomalies should show up accordingly in the MTST. To test this hypothesis, the MTST $\Phi(X; S_{f_S})$ for the k-means clustering was averaged over all final shapelets in order to reduce it from dimension $V \times N \times f_S$ to $V \times N$ and thus make it visually presentable. An excerpt for all variables for the first 75000 timesteps can be found in A.15, in the following are two representable examples shown. In figure 7.1 is the averaged MTST excerpt for relative humidity shown. Additionally, all true positives, false positives and false negatives are colourcoded in green, yellow and red respectively. Most anomalies within the relative humidity are visually easily identifiable, e.g. the first four anomalies that were identified as level shift and zero anomalies with a corresponding shapelet in section 6.1.3. A distance-based detection method should therefore be able to recognize points and sequences distant from the baseline as anomalies, even if this has not always worked. Thus, even though only

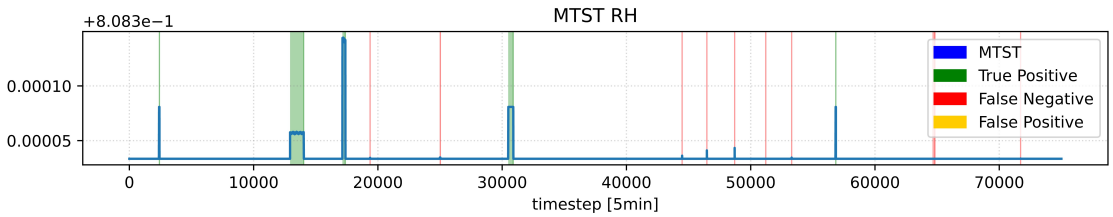


Figure 7.1. – The multivariate time series transformation for relative humidity for the first 75000 timesteps, with overlaying anomaly detection results

normal shapelets were found for relative humidity, they were sufficient to produce a MTST that could be meaningfully interpreted.

Figure 7.2 shows how the MTST behaves if no anomalous shapelets are found and the normal shapelets do not have enough variety to separate normal and anomalous behaviour. For the plantmass, only shapelets that were similar to the centrepiece of a growth period of 40 days were selected (shapelet no. 1 - 11 in A.7) For this area the MTST shows the lowest distance values. In the absence of a final shapelet that encompasses both the initial and final phases, the distance values in that region increase, resulting in frequent false positives and blurring the sharp delineation of anomalies based on their distance. Nevertheless, these anomalies remain visible in the MTST despite the challenges posed by the absence of a comprehensive shapelet that spans the complete anomaly.

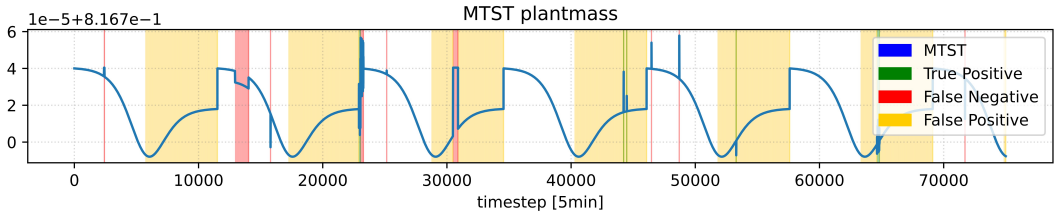


Figure 7.2. – Leftmost shapelet represents cluster centroid of cluster 0. The next four shapelets are the closest shapelets from the same cluster. Shapelets are derived from variables relative humidity, temperature, and VDP

In the case of k-means, two variables, plantmass and PAR, contain frequent, periodic false positives, from which the low results can be derived. For the other two clus-

tering methods, the behaviour of the MTST is similar with frequent false positives for individual variables, especially plantmass. Several factors can be considered as reasons for a low shapelet variance and a low occurrence of anomalous shapelets.

While reviewing the final shapelets to identify anomaly prototypes, the first factor was found that can negatively influence external metrics and shapelet variety. It was noted that point anomalies and short collective anomalies tend to vanish in a cluster consisting of normal class shapelets. This phenomenon will be called vanishing anomalies. Figure 7.3 shows the centroid (left) of a normal cluster extracted from the synthetic dataset experiments by k-means and its four closest shapelets of the same cluster. Even though shapelet no. 3 inherits a visible spike anomaly, it was assigned to the same cluster. All clustering methods utilized the Euclidean distance.

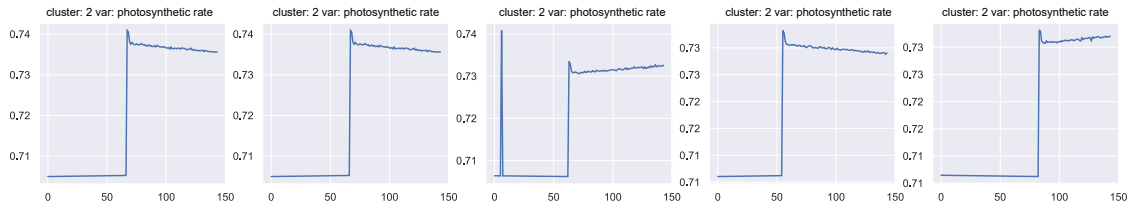


Figure 7.3. – The centroid shapelet and four closest shapelets from the same cluster from the synthetic dataset. Visible is the anomalous third shapelet, containing a spike anomaly.

For two shapelets S_1 and S_2, both of length $|S|$, where s denotes one timestep, the Euclidean distance is calculated as:

$$dist(S_1, S_2) = \sqrt{\sum_{i=1}^{|S|}(s_{1_i} - s_{2_i})^2} \tag{7.1}$$

Even for the smallest shapelet length α_{12h}, the shapelets consist of 144 timesteps. The influence of spikes, drops or other very short anomalies on clustering is therefore small. This probably has a direct influence on the anomaly detection. Anomalous points can only be clustered according to the fact that their MTST has a large distance to all shapelets. In the optimal case, there are also a few anomalous shapelets with a small distance to make the classification more accurate. Of course, the selec-

tion of anomaly prototypes is also restricted, since these vanishing anomalies do not play a role in the further interpretability and representation evaluation.

Another factor influencing shapelet selection is γ, the tradeoff between cluster size and cluster distance, in the utility measure for each shapelet. γ was not optimised but inherited from literature. This can explain why, for example, the experiments for the synthetic data set did not produce any anomaly prototypes for a shapelet length smaller than 40 days. The synthetic data set is homogeneous, normal class shapelets of one variable differ little or not at all. The probability of containing an anomaly decreases with shapelet length, which means that many normal shapelets exist, but only a few with anomalies. Since the normal shapelets are also extremely similar, the clusters without anomalies are significantly larger and denser than those with them. For short shapelets, this effect seems to have been so large that the equal weighting of cluster size and cluster distance resulted in the significant cluster size predominating and mainly nearly equal normal class shapelets being selected. For a more complex data set, such as the EDEN ISS FEG, this effect appears to cancel out; here the distribution of abnormal to normal shapelets was 50.9% to 33.3% across all lengths, with no precise identification possible for the remainder. For future research, γ should be implemented as an optimizable hyperparameter.

The phenomenon of false equal weighting has had a discernible impact on anomaly detection, especially for plantmass, where the majority of recurrent false positives occurred. Furthermore, the length of the shapelets exerted a significant influence on the dynamics. As mentioned, longer shapelets have a higher probability of containing anomalies and exhibiting greater variations, such as including the harvest period. However, an increase in shapelet length corresponded to a decrease in the cluster size of normal shapelets. Consequently, shorter normal shapelets are preferred over those that encompass a complete period of 40 days, resulting in a lower chance of them being included in the final set of shapelets, thereby yielding observable consequences for the MTST. All other variables had a factor that made the shapelet clusters much more fine-grained, even for short lengths, and thus shapelets of different lengths or with anomalies were selected, noise. The presence of noise leads to a broader dispersion of normal shapelets, which in turn prevents obtaining dominant normal clusters, as larger clusters have the potential to fragment.

This must be followed by a critical appraisal of the alpha values adopted. Since anomalous shapelets are of particular interest, the focus should be on extracting them through a well-adjusted shapelet length. In both data sets, the alpha values are based on periodicities within the data set, e.g. determined using autocorrelation and partial autocorrelation. Figure 7.4 shows that this approach alone may not be enough. Shown is a section of an anomalous shapelet cluster for the synthetic dataset, the centroid is at the first position, followed by the closest four shapelets. The centroid fulfils all requirements for an anomaly prototype, it is closest to as an anomaly detected location in the dataset and the cluster contains only shapelets of the same anomalous type from different variables, but the visualization is impaired by its length. Nevertheless, since the cluster belongs to level shift anomalies with a length greater than 1000 time steps, α_{40days} is the only way to detect anomalies of this length. For future research, it is therefore, necessary to also derive the shapelet length from manually recorded anomalies to obtain the shortest but visually most meaningful length.

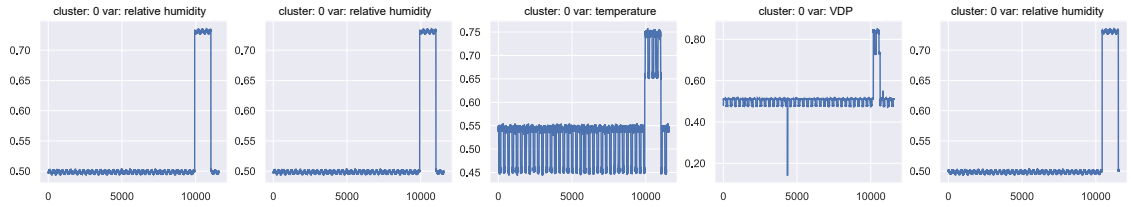


Figure 7.4. – Leftmost shapelet represents cluster centroid of cluster 0. The next four shapelets are the closest shapelets from the same cluster. Shapelets are derived from variables relative humidity, temperature, and VPD

8. Conclusion & Outlook

In this thesis, a novel workflow was proposed for shapelet-based unsupervised multivariate time series anomaly detection and unsupervised anomaly protoype identification. The workflow includes shapelet discovery and selection, multivariate time series transformation and unsupervised anomaly detection. The shapelet discovery process identifies windowed subsequences and filters outliers while preserving representative shapelets, utilizing distance, density and deep learning clustering methods. The multivariate transformation, based on a point-by-point squared distance calculation, tackles the challenge of representing different anomaly types without labels. The anomaly detection uses the transformed time series in a k-Means clustering to identify anomalies in time points, sequences, or entire time series. Also, a multivariate time series anomaly generator was created, based on four independent and two dependent variables derived from the EDEN ISS FEG dataset and implemented seven different anomaly types, in a univariate or multivariate manner to provide a more adjustable environment for future anomaly detection studies.

Experiments were conducted on a labelled synthetic and unlabeled real-world telemetry dataset to compare the different clustering methods and to verify the general suitability and functionality of the workflow. With regard to the internal metrics for evaluating the clustering and thus the meaningfulness and representability of the shapelets, values were achieved for both data sets that in some cases approached the best possible. k-Means consistently asserted itself as the best clustering method, although it must be taken into account that in the case of DBSCAN, shapelets are also included that DBSCAN would normally remove as noise, which negatively influences the evaluation. As a deep learning method, Mdc-CNN fell significantly short of expectations. For parallel clustering over all shapelet lengths as well as for single shapelets, Mdc-CNN could never reach beyond k-Means, although it is based

on it. Possible explanations are, on the one hand, that Mdc-CNN has only been used successfully for shapelets that were significantly shorter and that the model was adopted unchanged, i.e. hyperparameter optimisation did not take place.

It was observed that the application of various clustering methods enabled the detection of diverse anomalies. Nevertheless, the outcomes were found to be inconclusive and did not exhibit a distinct preference for any particular clustering technique, as indicated by the quantitative internal evaluation metrics. Reasons for this are, on the one hand, that the multivariate time series transformation is not always reliably able to clearly separate anomalies from the normal class. On the other hand, that k-Means incorrectly classified easy-to-detect anomalies as normal and many false positives due to the ambiguous transformation. Several factors that negatively influence the predictive power of the MTST were detected. Among them, the not optimized weighting of cluster size and cluster distance during shapelet selection is a significant issue. Additionally, the problem of vanishing anomalies caused by the Euclidean distance used for clustering also impacts the performance of the method. These factors contribute to suboptimal anomaly detection results and merit further investigation for potential improvements in the MTST approach.

The shapelets that could be used as anomaly prototypes were, without exception, of the greatest possible length for the synthetic dataset. This contradicts the definition of shapelets as short subsequences and shows the importance of the alpha values that corresponded to the periodicities in the individual time series. For future investigations, the implementation of a noise factor for the synthetic data set can lead to improved results for certain variables. In the EDEN ISS dataset, the anomaly prototypes were more clearly scattered over different lengths, which was probably also related to the greater complexity of the dataset.

All in all, the goal of finding different types of anomalies unsupervised in a multivariate time series was achieved. However, it was also shown that there is still room for improvement and the need for further research. The alpha values used are critical factors for achieving accurate anomaly detection. Nonetheless, finding the best possible shapelet length can be challenging, even when utilizing methods like autocorrelation or partial autocorrelation. An extension could be to include alpha

values based on individual hand-identified anomalies to obtain more samples for a better length distribution. The shapelet selection can also be improved or extended, for example by taking the density of the clusters into account, or by replacing the distance calculation between the clustered shapelets with the scatter-sensitive Mahalanobis distance instead of the Euclidean distance. The proposed multivariate time series transformation and unsupervised anomaly detection in this study have shown promise, but further investigations and enhancements are needed to optimize the performance. The transformation should be enhanced to better distinguish between normal and abnormal classes, ensuring a more precise identification of anomalies. Additionally, anomaly detection needs refinement to avoid disregarding easily detectable anomalies within the transformed data.

The possibility of incorporating direct feedback into the workflow has not yet been addressed. The anomaly prototypes can be used to let the proposed framework learn better detection decisions using labelling. By manually labelling the related cluster or only subparts, more shapelets can be used as comparison patterns for the detection, thus spreading the probability of finding anomalies more widely. With each cluster that can be assigned to an anomaly type and with each iteration, the potential to improve the anomaly detection rate increases. Simultaneously, this iterative approach allows for more precise and well-founded statements about the dataset, leading to a deeper understanding of the anomalies present. In this way, not only individual parts would be able to learn from data, but the entire process would be able to optimize.

A. Appendix

Differentiation of the triplet loss function, [LCX+21b, p. 3]: The Definition for the differentiation is given for a generic anchor point x, positive samples x^+ and negative samples x^-, were K^+ denotes the top candidates from the same cluster and K^- the number of randomly picked candidates from other clusters in proportion. For the sake of clarity, $D^+_{i,j}$ represents:

$$D^+_{i,j} = \|f(x^+_i) - f(x^+_j)\|^2_2$$

and $D^-_{i,j}$ represents:

$$D^-_{i,j} = \|f(x^-_i) - f(x^-_j)\|^2_2$$

By using the estimations:

$$D_{\text{pos}} \approx \tilde{D}_{\text{pos}} = \frac{\sum_{i=1}^{K^+}\sum_{j=1}^{K^+} D^+_{i,j} \cdot e^{\alpha \cdot D^+_{i,j}}}{\sum_{i=1}^{K^+}\sum_{j=1}^{K^+} e^{\alpha \cdot D^+_{i,j}}}$$

$$D_{\text{neg}} \approx \tilde{D}_{\text{neg}} = \frac{\sum_{i=1}^{K^-}\sum_{j=1}^{K^-} D^-_{i,j} \cdot e^{\alpha \cdot D^-_{i,j}}}{\sum_{i=1}^{K^-}\sum_{j=1}^{K^-} e^{\alpha \cdot D^-_{i,j}}}$$

where $\alpha > 0$ yields a smooth maximum approximation, the gradients of overall maximum distance are:

$$\frac{\partial \tilde{D}_{\text{pos}}}{\partial D^+_{i,j}} = \frac{e^{\alpha \cdot D^+_{i,j}} \left(1 + \alpha(D^+_{i,j} - \tilde{D}_{\text{pos}})\right)}{\sum_{i=1}^{K^+}\sum_{j=1}^{K^+} e^{\alpha \cdot D^+_{i,j}}}$$

and

$$\frac{\partial \tilde{D}_{\text{neg}}}{\partial D^-_{i,j}} = \frac{e^{\alpha \cdot D^-_{i,j}} \left(1 + \alpha(D^-_{i,j} - \tilde{D}_{\text{neg}})\right)}{\sum_{i=1}^{K^-} \sum_{j=1}^{K^-} e^{\alpha \cdot D^-_{i,j}}}$$

Thus, the gradients of the full loss function with respect to $f(x)$, $f(x_i^+)$, and $f(x_i^-)$ are as follows:

$$\frac{\partial \mathcal{L}}{\partial f(x)} = \frac{2 \cdot \sum_{i=1}^{K^+} \|f(x) - f(x_i^+)\|_2}{\sum_{i=1}^{K^+} \|f(x) - f(x_i^+)\|_2^2} - \frac{2 \cdot \sum_{i=1}^{K^-} \|f(x) - f(x_i^-)\|_2}{\sum_{i=1}^{K^-} \|f(x) - f(x_i^+)\|_2^2}$$

$$\frac{\partial \mathcal{L}}{\partial f(x_i^+)} = \frac{2 \cdot \sum_{i=1}^{K^+} \|f(x) - f(x_i^+)\|_2}{\sum_{i=1}^{K^+} \|f(x) - f(x_i^+)\|_2^2} + \sum_{j=1}^{K^+} \frac{\partial \tilde{D}_{\text{pos}}}{\partial D^+_{i,j}} \cdot 4\|f(x_i^+) - f(x_j^+)\|_2$$

$$\frac{\partial \mathcal{L}}{\partial f(x_i^-)} = \frac{2 \cdot \sum_{i=1}^{K^-} \|f(x) - f(x_i^-)\|_2}{\sum_{i=1}^{K^-} \|f(x) - f(x_i^-)\|_2^2} + \sum_{j=1}^{K^-} \frac{\partial \tilde{D}_{\text{neg}}}{\partial D^-_{i,j}} \cdot 4\|f(x_i^-) - f(x_j^-)\|_2$$

Since the last two equations are differentiable, backpropagation can be used over the entire neural network based upon minibatch stochastic gradient descent together with Adam [KB15] optimizer to optimize the Mdc-CNN parameters.

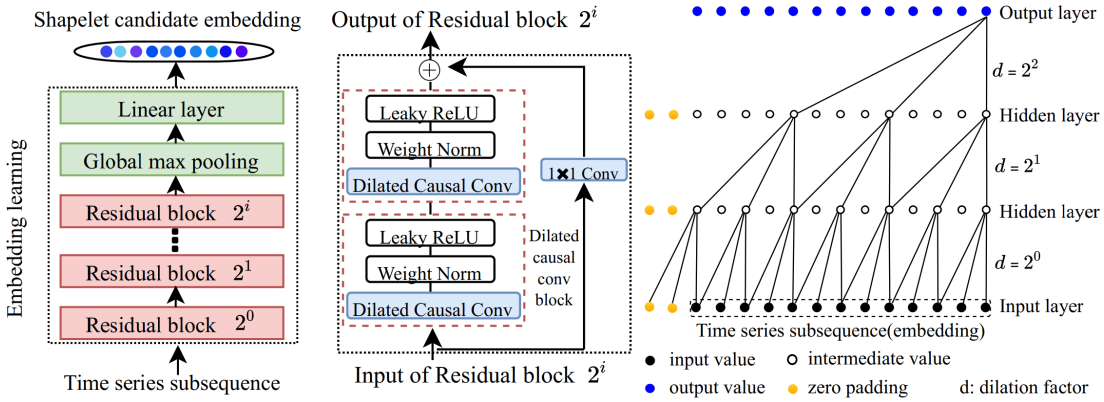


Figure A.1. – An elaboration of the Multi-length-input dilated causal Convolutional Neural Network (Mdc-CNN), showing the encoder architecture, one residual block and the dilated causal convolution layer, in accordance with [LCX+21a]

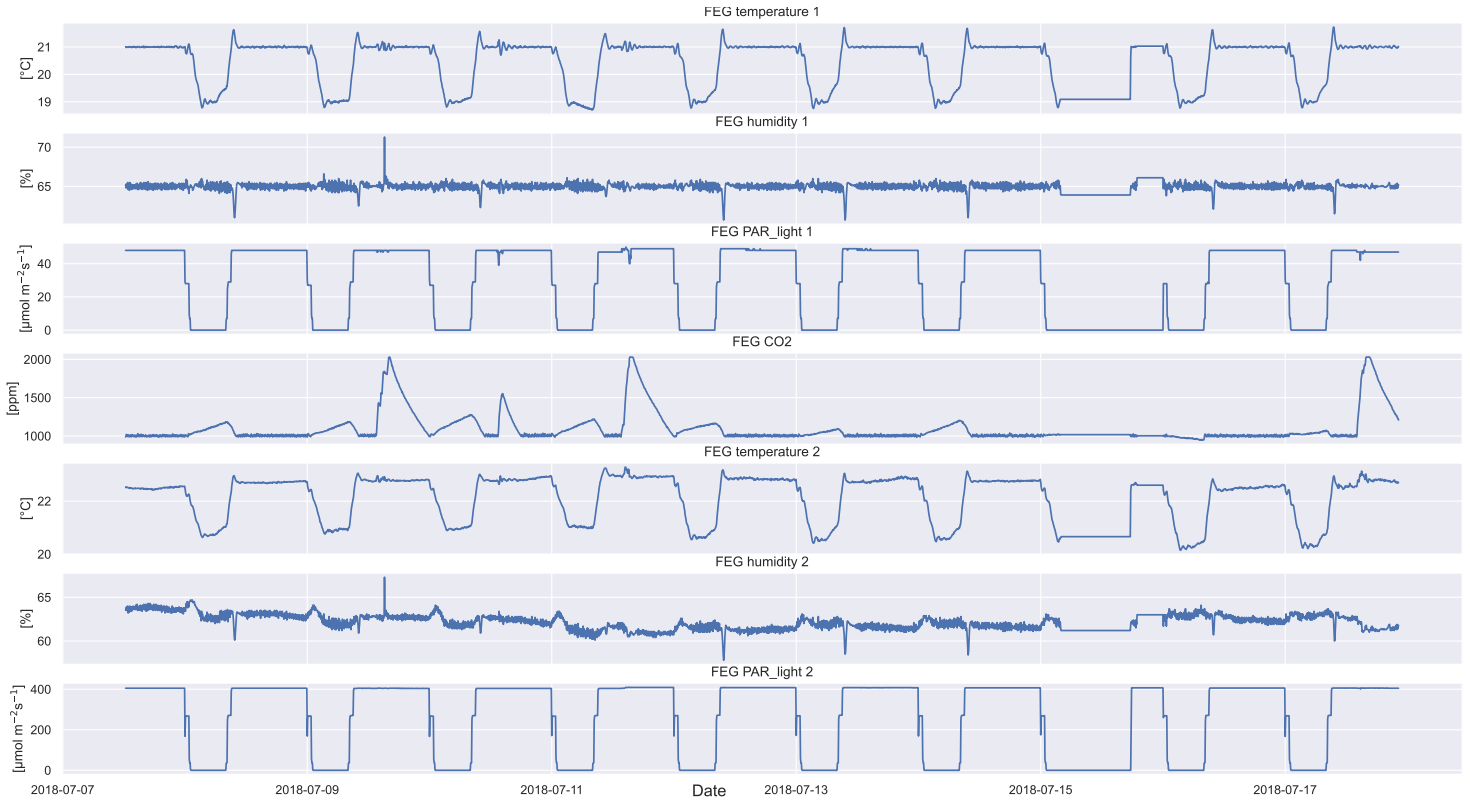


Figure A.2. – An extract of the feg dataset. Visible are the variable normal cases together with different anomalies

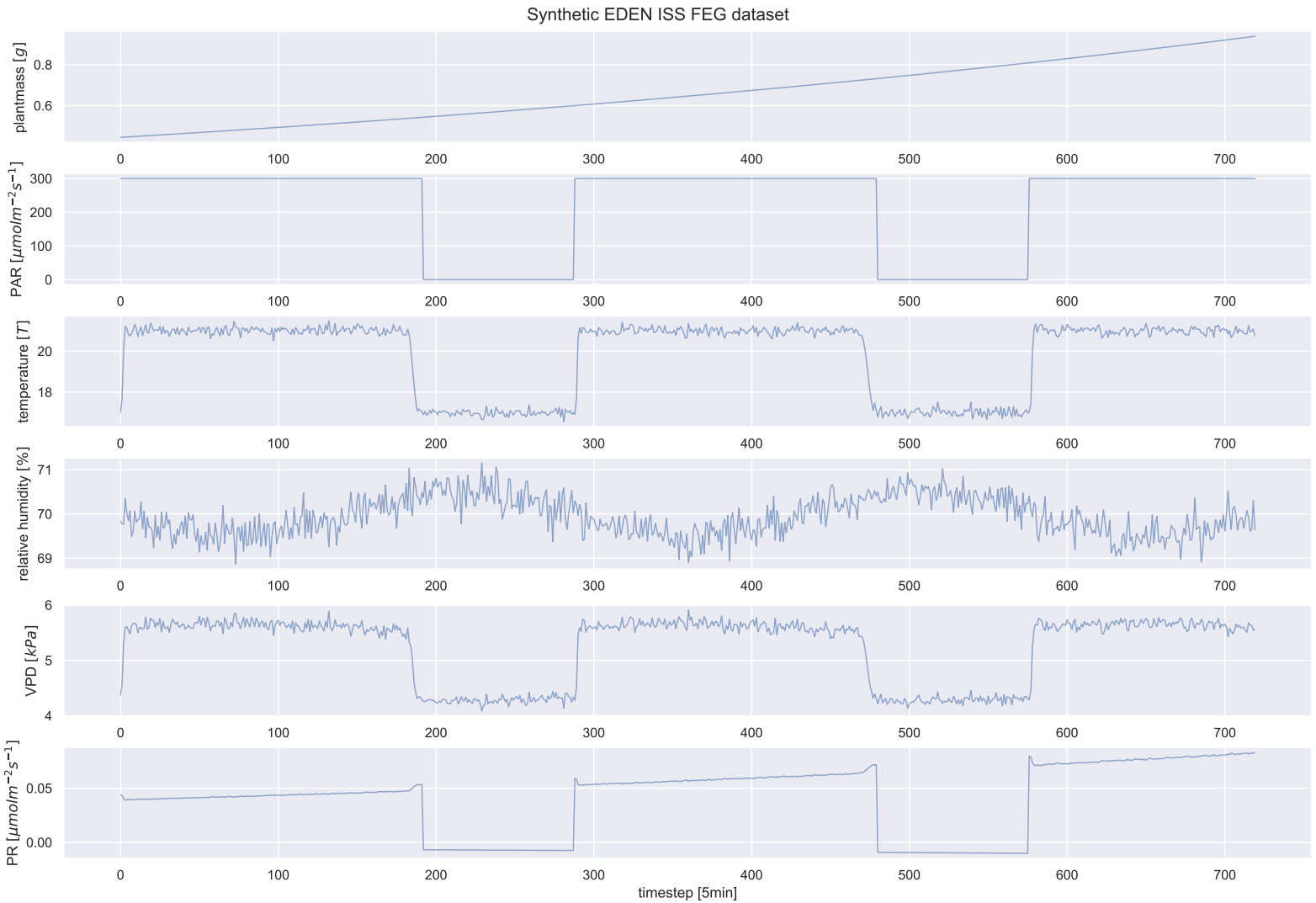


Figure A.3. – The first two and a half days of the synthetically generated dataset without anomalies

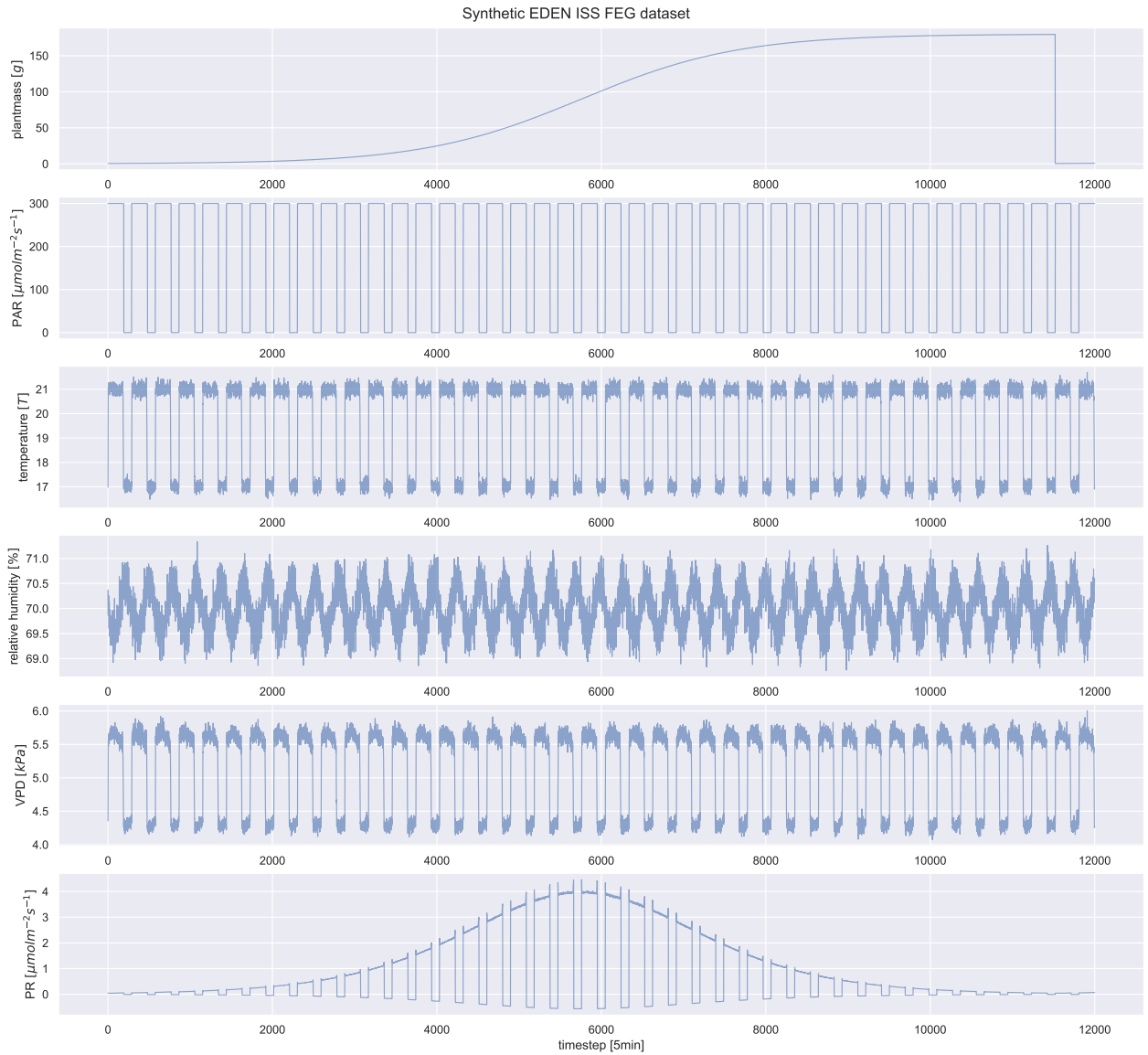


Figure A.4. – The first fourty days of the synthetically generated dataset without anomalies

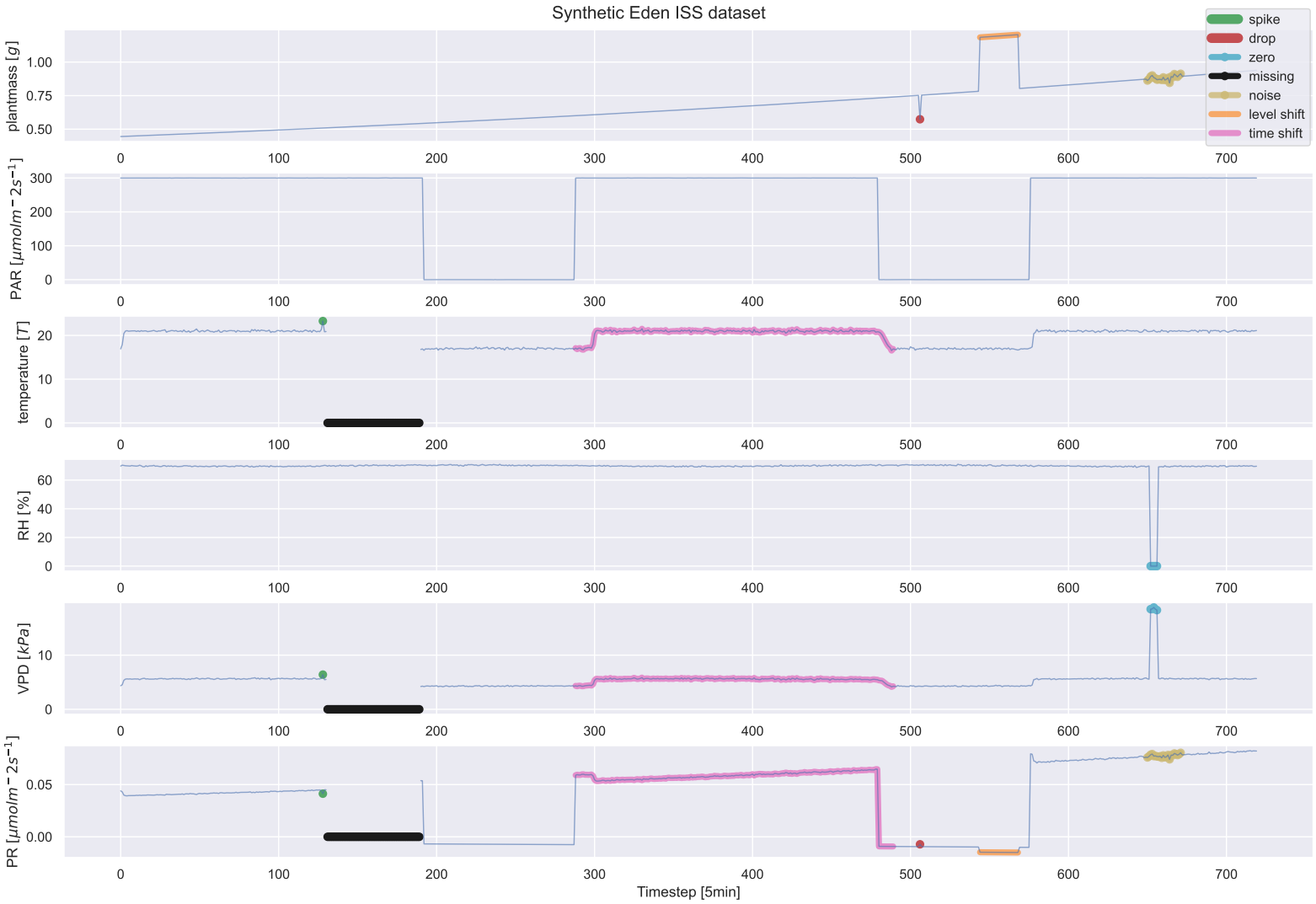


Figure A.5. – The first two and a half days of the synthetically generated dataset with all seven different anomalies. Dependent variables are calculated based on anomalous data.

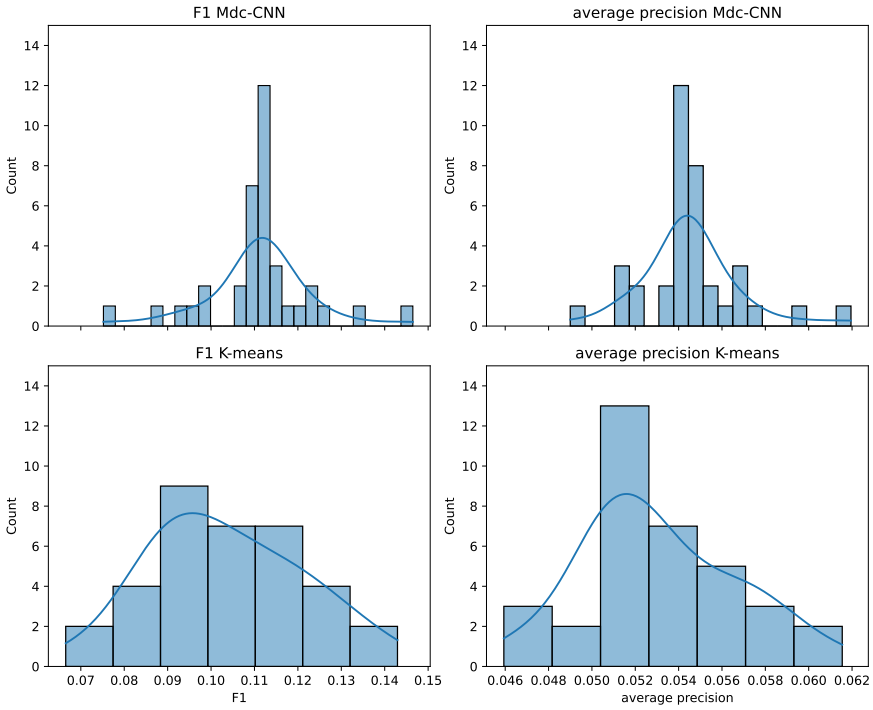


Figure A.6. – Histogramm for F1 score and average precision for Mdc-CNN and k-Means, without the k-Means outlier for a better comparability

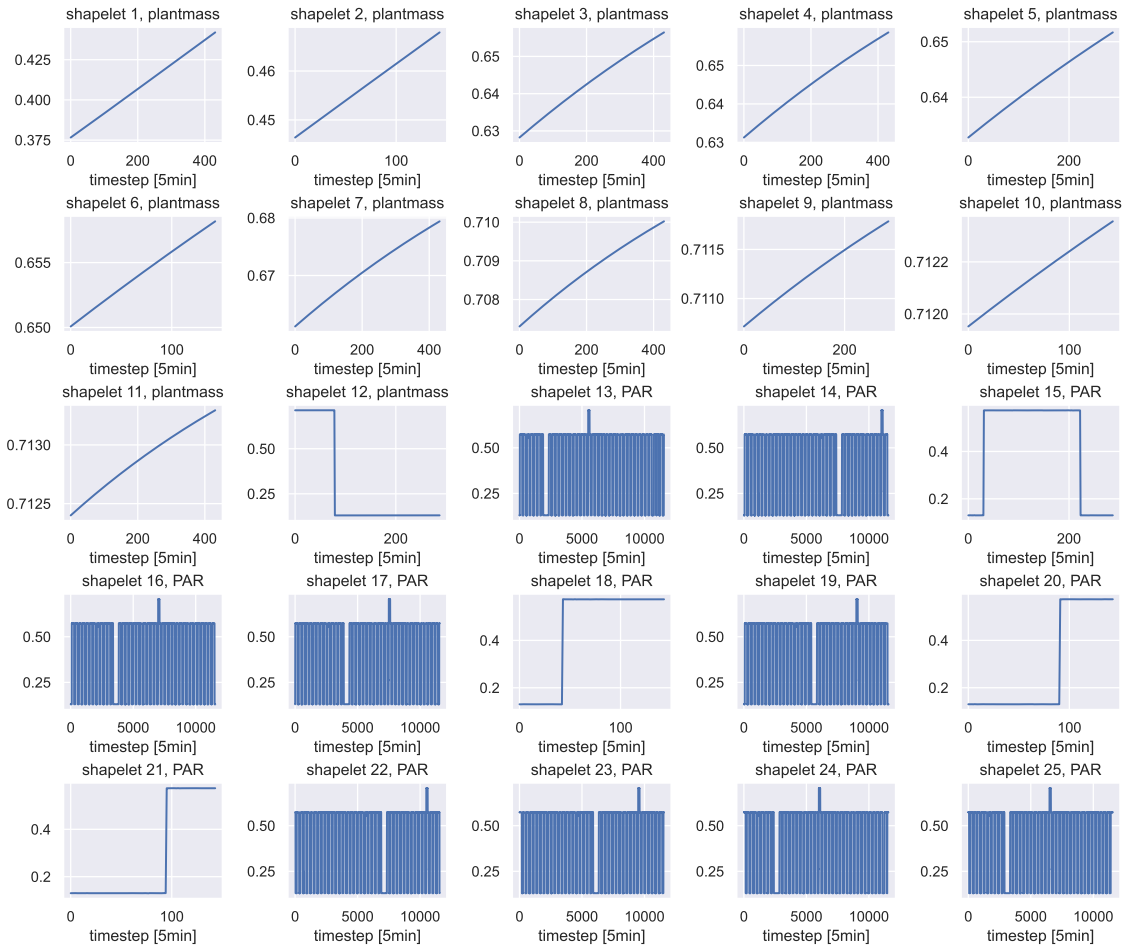


Figure A.7. – Part 1 of all 46 shapelets, that are closest to at least one as anomalous detected timestep or subsequence of the synthetic datdaset.

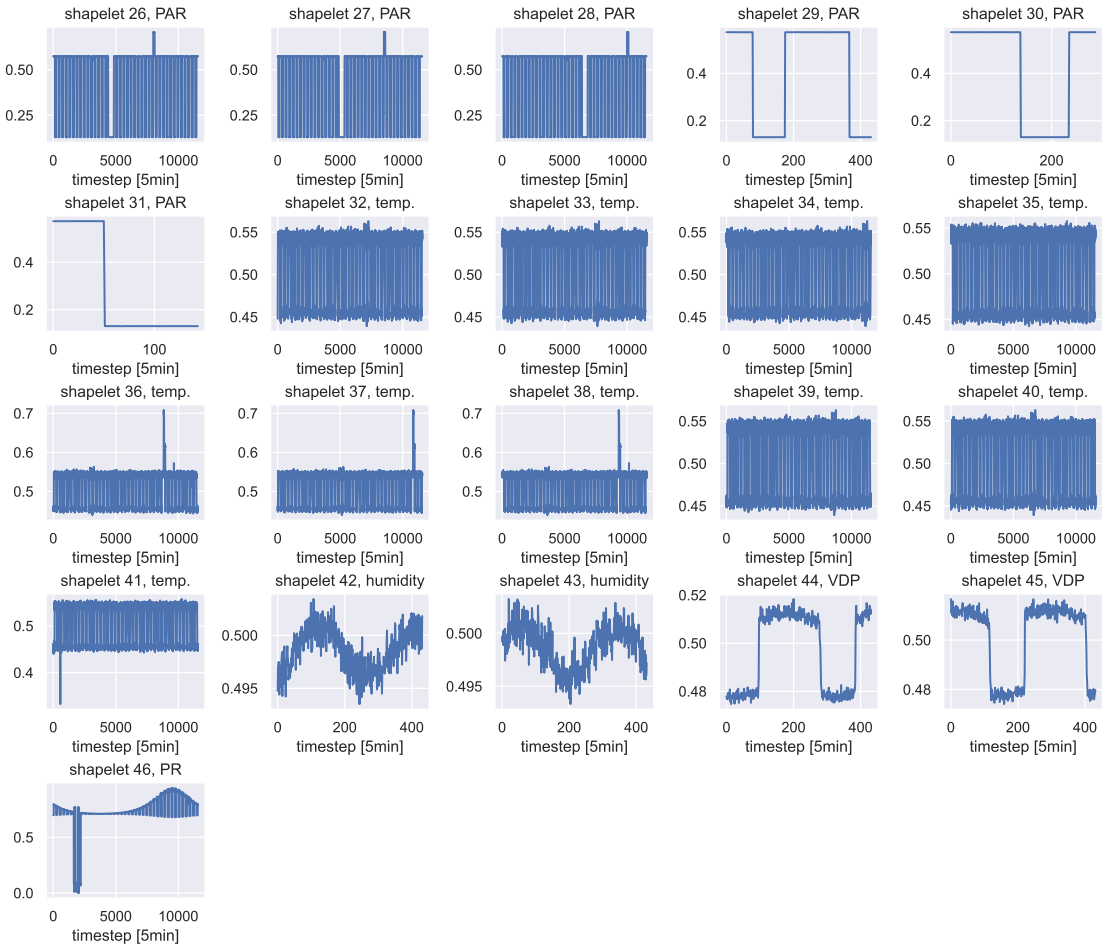


Figure A.8. – Part 2 of all 46 shapelets, that are closest to at least one as anomalous detected timestep or subsequence of the syntethic datdaset.

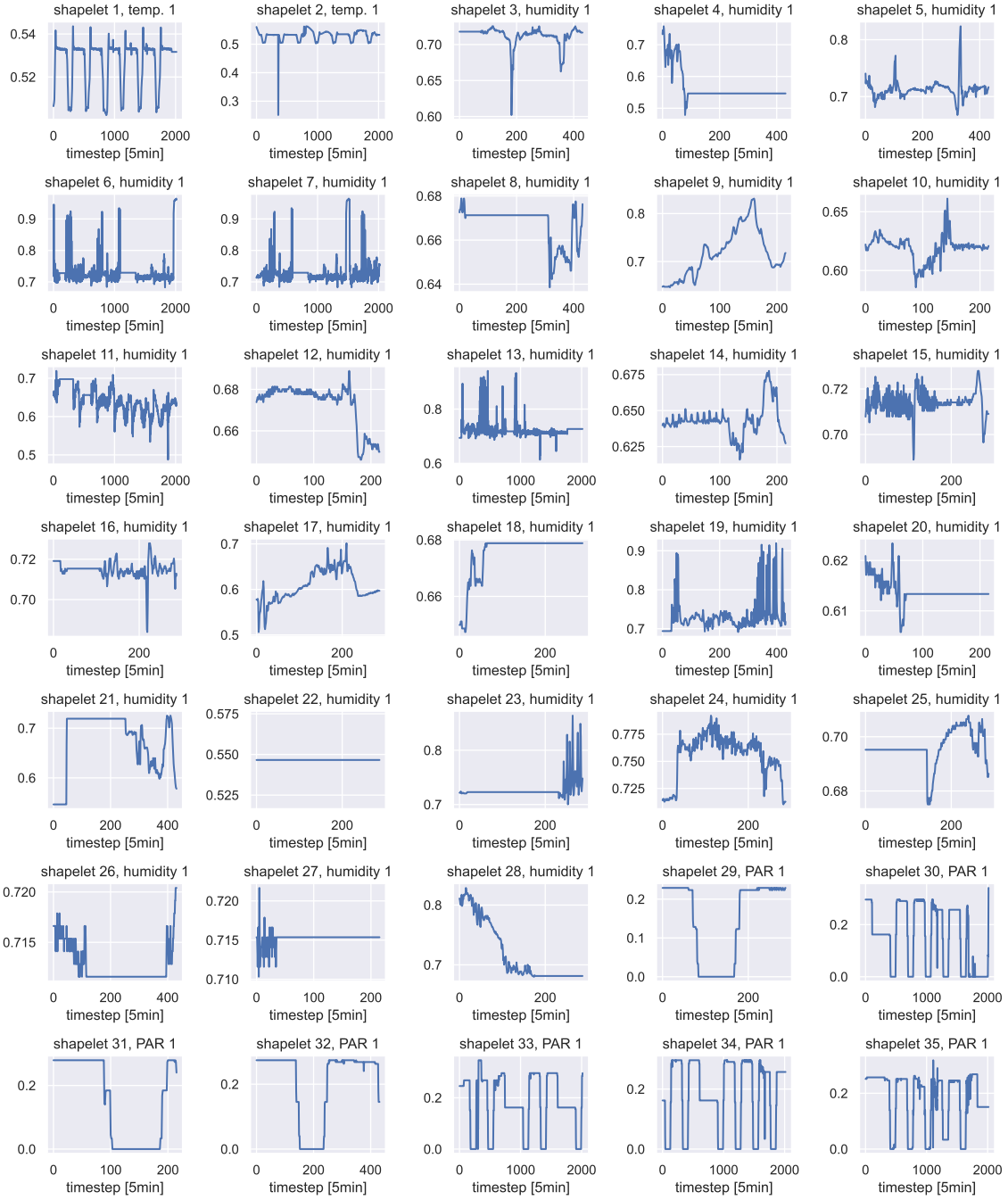


Figure A.9. – Part 1 of all 108 shapelets, that are closest to at least one as anomalous detected timestep or subsequence of the EDEN ISS FEG datdaset.

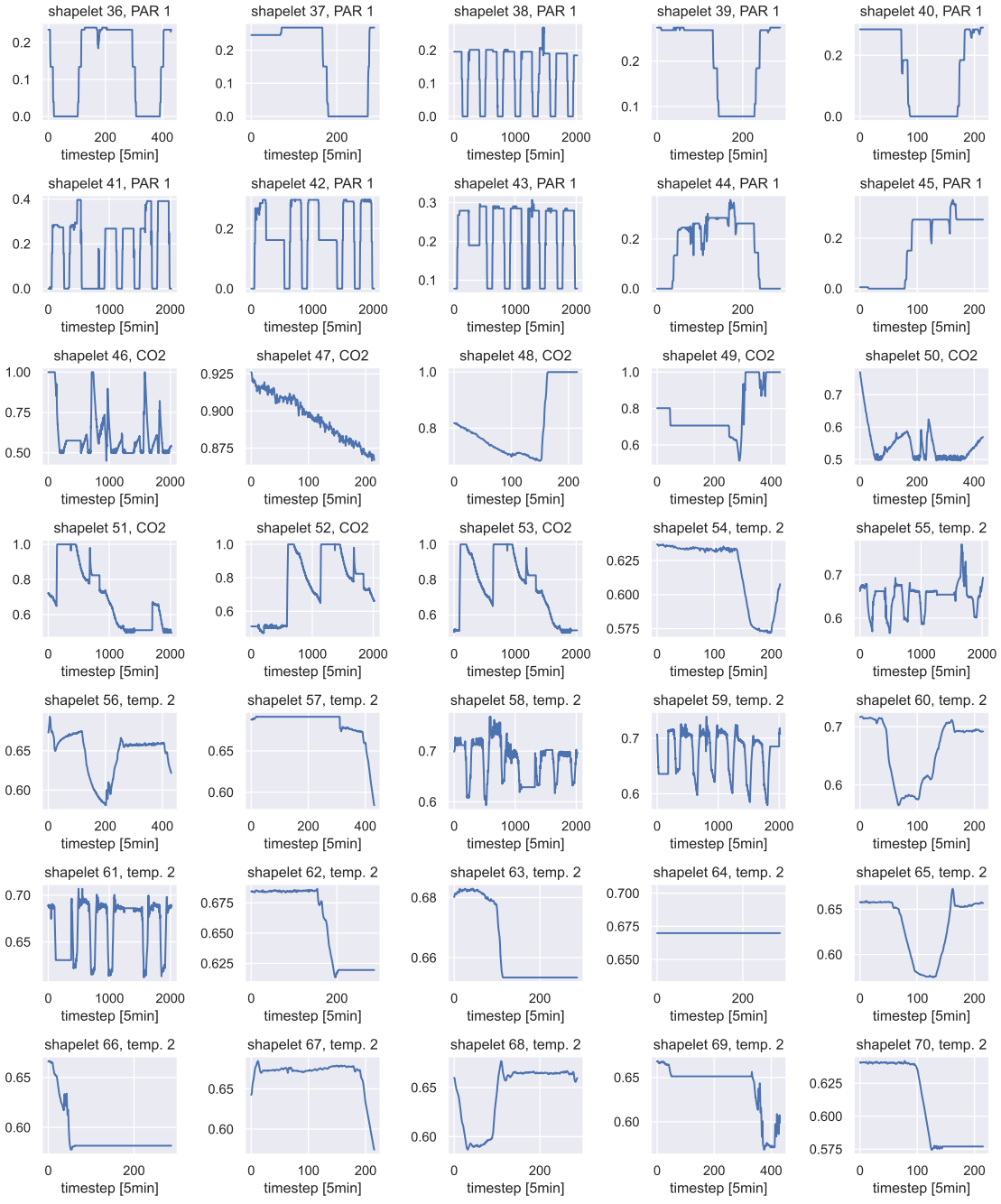


Figure A.10. – Part 2 of all 108 shapelets, that are closest to at least one as anomalous detected timestep or subsequence of the EDEN ISS FEG datdaset.

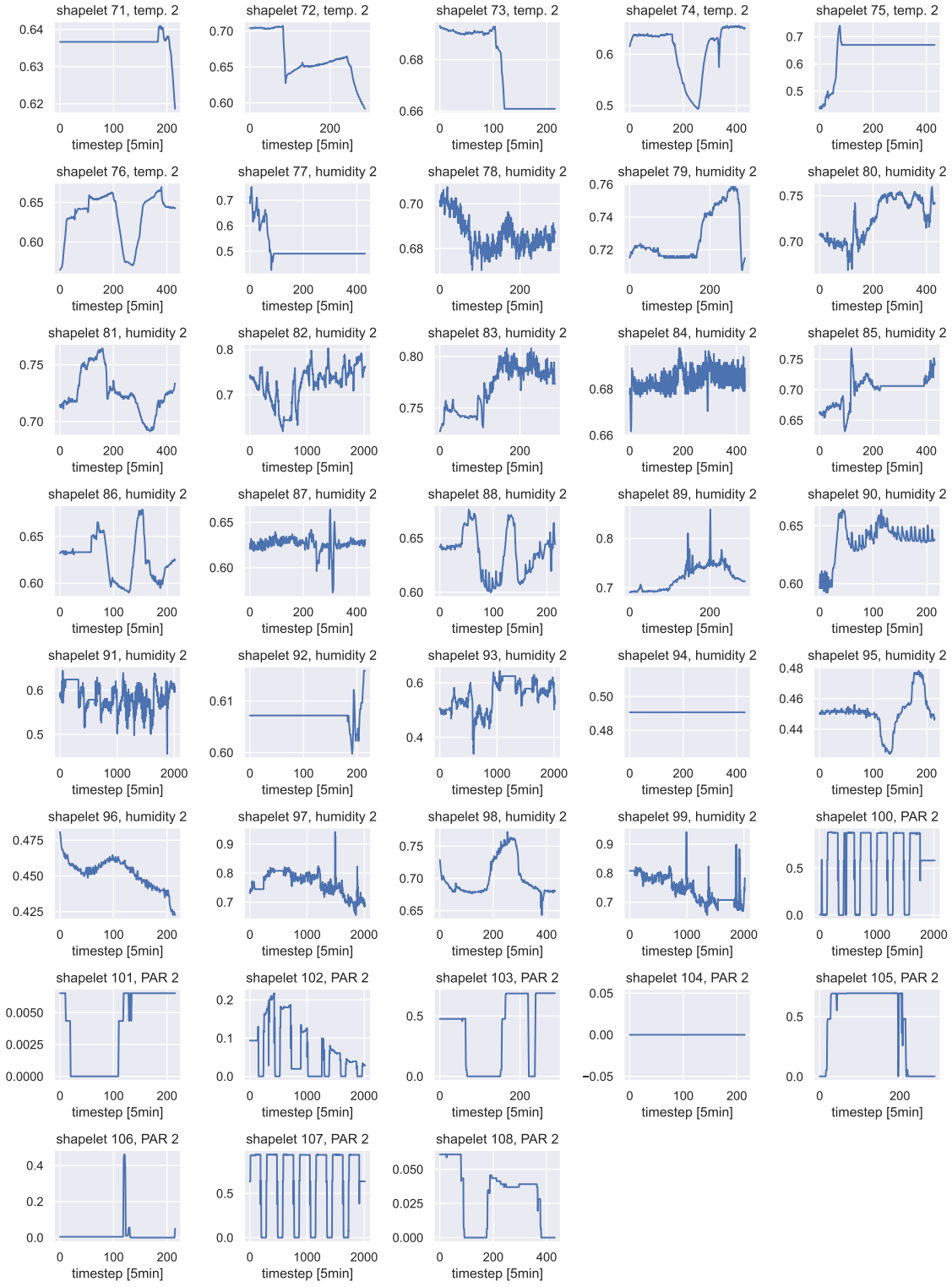


Figure A.11. – Part 3 of all 108 shapelets, that are closest to at least one as anomalous detected timestep or subsequence of the EDEN ISS FEG datdaset.

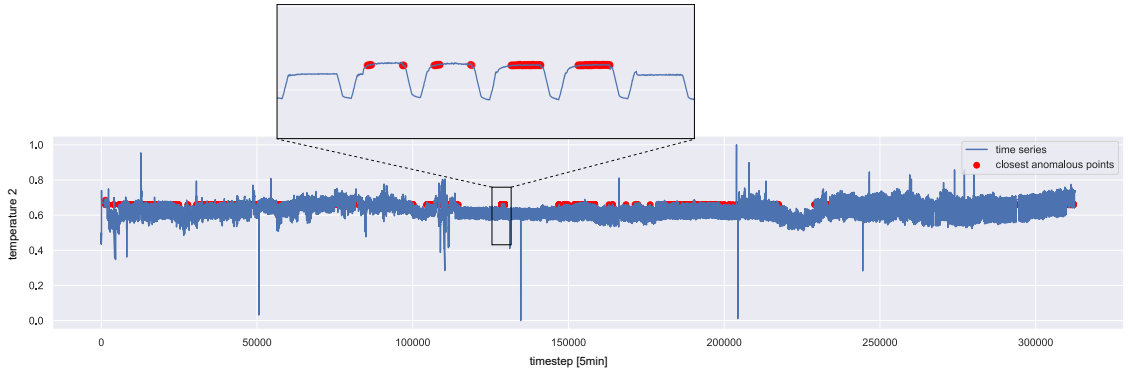


Figure A.12. – Closest anomalous points to shapelet no. 68 in A.13. Detected anomalies are mainly level shifts, as visible in the excerpt

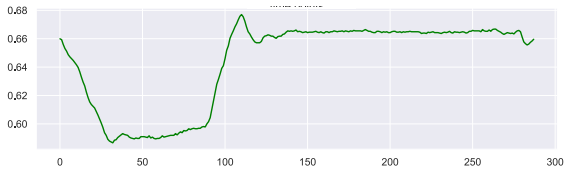


Figure A.13. – Normal shapelet no. 68 for temperature 2 extracted using α_{24h}

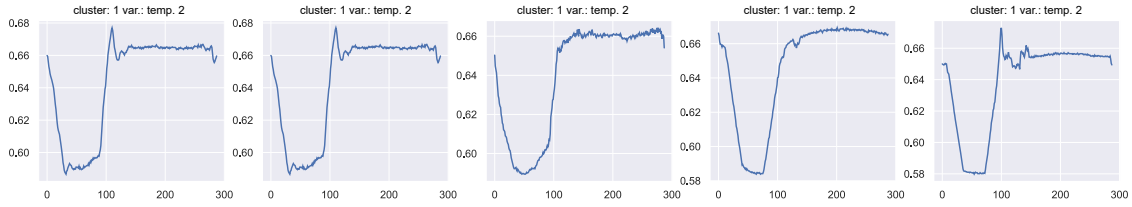


Figure A.14. – The shapelet no. 68 in A.13 (leftmost) and four nearest shapelets within the same cluster

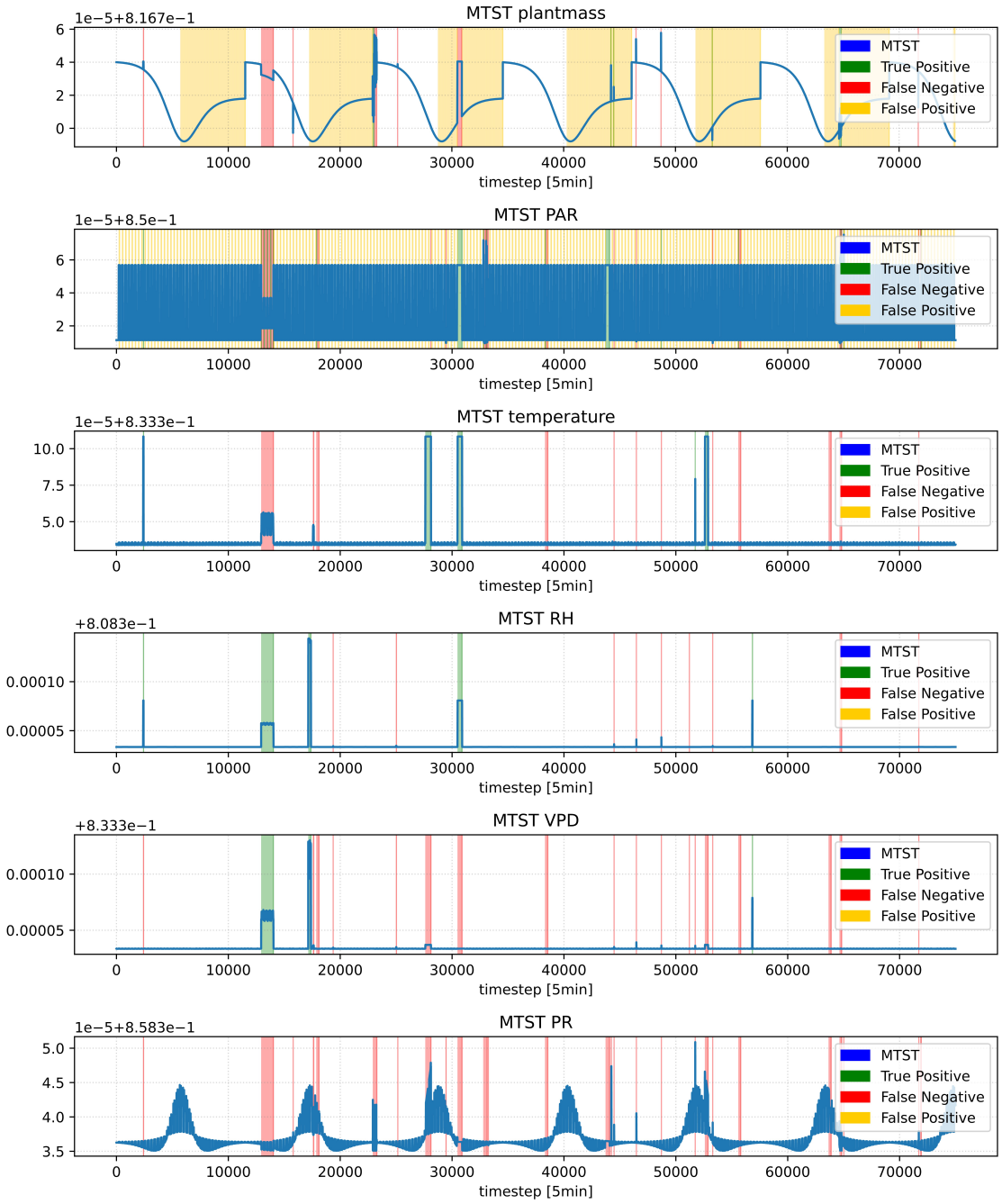


Figure A.15. – First 75000 averaged time steps for the multivariate time series transformation of the synthetic dataset

CD Structure

A soft version of the current work is submitted. Following, the CD structure is presented.

1. Masterarbeitstext
2. Masterarbeitscode
    - Shapelet Based Anomaly Detection
    - Time Series Anomaly Generator
3. Masterarbeitsergebnisse
    - Realdatensatz
    - Synthetischer Datensatz

List of Tables

List of Figures

List of Source Code

Bibliography

[AA22] ArulMonica ; Ahsan, Kareem: *Data anomaly detection for structural health monitoring of bridges using shapelet transform*. 01 2022 1, 2.3, 2.3, 4.2, 4.3, 4.3, 4.4

[AAXJ21] Alqahtani, Ali ; Ali, Mohammed ; Xie, Xianghua ; Jones, Mark W.: Deep Time-Series Clustering: A Review. In: *Electronics* 10 (2021), dec, Nr. 23, 3001. http://dx.doi.org/10.3390/electronics10233001. – DOI 10.3390/electronics10233001 1, 2.1.2, 2.1.2, 2.2, 2.2, 2.4, 4.4

[AB21] Altay, Tayip ; Baydoğan, Mustafa G.: "A new feature-based time series classification method by using scale-space extrema". In: *Engineering Science and Technology, an International Journal* 24 (2021), Nr. 6, pp. 1490-1497. http://dx.doi.org/https://doi.org/10.1016/j.jestch.2021.03.017. – DOI https://doi.org/10.1016/j.jestch.2021.03.017. – ISSN 2215–0986 2.1.1

[AK21] Arul, Monica ; Kareem, Ahsan: Applications of shapelet transform to time series classification of earthquake, wind and wave data. In: *Engineering Structures* 228 (2021), 01, S. 111564 2.3.1

[ASW15] Aghabozorgi, Saeed ; Shirkhorshidi, Ali S. ; Wah, Teh Y.: Time-series clustering – A decade review. In: *Information Systems* 53 (2015), oct, 16–38. http://dx.doi.org/10.1016/j.is.2015.04.007. – DOI 10.1016/j.is.2015.04.007 2.1, 2.1.4, 2.1.6, 2.1.2, 2.1.2, 2.1.2, 2.2, 2.4, 4.1, 4.3, 4.4, 5.3.1, 5.3.2

[AV07] Arthur, David ; Vassilvitskii, Sergei: K-Means++: The Advantages of Careful Seeding. In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. USA : Society for Industrial and Applied Mathematics, 2007 (SODA '07). – ISBN 9780898716245, S. 1027–1035 2.1.2, 2.1.2, 4.1.1

[BB17] Bostrom, Aaron ; Bagnall, Anthony J.: *A Shapelet Transform for Multivariate Time Series Classification*. http://dx.doi.org/10.48550/ARXIV.1712.06428. Version: 2017 2.1.1, 2.2, 2.4, 4.3, 4.3

[BDL+18] Bagnall, Anthony ; Dau, Hoang A. ; Lines, Jason ; Flynn, Michael ; Large, James ; Bostrom, Aaron ; Southam, Paul ; Keogh, Eamonn: *The UEA multivariate time series classification archive, 2018*. 2018 5.4.6

[BGCML21] Blázquez-García, Ane ; Conde, Angel ; Mori, Usue ; Lozano, Jose A.: A Review on Outlier/Anomaly Detection in Time Series Data. In: *ACM Computing Surveys* 54 (2021), apr, Nr. 3, 1–33. http://dx.doi.org/10.1145/3444690. – DOI 10.1145/3444690 1, 2.1.7, 2.1.3, 2.1.8, 2.1.9, 2.1.3, 2.2, 2.3, 4.4, 5.2.3

[BJR08] *Kapitel* 2. In: Box, George E. P. ; Jenkins, Gwilym M. ; Reinsel, Gregory C.: *Autocorrelation Function and Spectrum of Stationary Processes*. John Wiley Sons, Ltd, 2008. – ISBN 9781118619193, 21-46 5.4.3

[BKK18] Bai, Shaojie ; Kolter, J. Z. ; Koltun, Vladlen: *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. http://arxiv.org/abs/1803.01271. Version: 2018 4.1.3, 5.4.6

[BKNS00a] Breunig, Markus M. ; Kriegel, Hans-Peter ; Ng, Raymond T. ; Sander, Jörg: LOF: Identifying Density-Based Local Outliers. In: *SIGMOD Rec.* 29 (2000), may, Nr. 2, 93–104. http://dx.doi.org/10.1145/335191.335388. – DOI 10.1145/335191.335388. – ISSN 0163–5808

[BKNS00b] Breunig, Markus M. ; Kriegel, Hans-Peter ; Ng, Raymond T. ; Sander, Jörg: LOF: Identifying Density-Based Local Outliers. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA : Association for Computing Machinery, 2000 (SIGMOD '00). – ISBN 1581132174, 93–104

[BKS+18] Beggel, Laura ; Kausler, Bernhard X. ; Schiegg, Martin ; Pfeiffer, Michael ; Bischl, Bernd: Time series anomaly detection based on shapelet learning. In: *Computational Statistics* 34 (2018), Juli, Nr. 3, 945–976. http://dx.doi.org/10.1007/s00180-018-0824-9. – DOI 10.1007/s00180–018–0824–9 1, 2.1.3, 2.2, 2.3, 4.3, 4.3, 4.3, 4.4, 5.4.2, 7

[BLB+16] Bagnall, Anthony ; Lines, Jason ; Bostrom, Aaron ; Large, James ; Keogh, Eamonn: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. In: *Data Mining and Knowledge Discovery* 31 (2016), November, Nr. 3, pp. 606–660. http://dx.doi.org/10.1007/s10618-016-0483-9. – DOI 10.1007/s10618–016–0483–9 2.1.1, 2.2, 2.2

[BNZ19] Bochenek, Grazyna ; Nyberg, P. ; Zabel, Paul: *D5.4 LS Performance Document*. http://dx.doi.org/10.5281/zenodo.3229925. Version: apr 2019 5.2.3, 5.2.3, 5.2.3, 5.2.3, 5.2.3

[BPPF21] Boniol, Paul ; Paparrizos, John ; Palpanas, Themis ; Franklin, Michael J.: SAND in Action: Subsequence Anomaly Detection for Streams. In: *Proc. VLDB Endow.* 14 (2021), jul, Nr. 12, 2867–2870. http://dx.doi.org/10.14778/3476311.3476365. – DOI 10.14778/3476311.3476365. – ISSN 2150–8097

[BRGD19] Barz, Bjorn ; Rodner, Erik ; Garcia, Yanira G. ; Denzler, Joachim: Detecting Regions of Maximal Divergence for Spatio-Temporal Anomaly Detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2019), may, Nr. 5, 1088–1101. http://dx.doi.org/10.1109/tpami.2018.2823766. – DOI 10.1109/tpami.2018.2823766 2.1.3, 5.2, 5.2.3

[BV17] Bonzano, Guiseppe ; Vrakking, Vincent: *D3.7 Environmental Control Design Document*. http://dx.doi.org/10.5281/zenodo.3229763. Version: September 2017 5.1, 5.2, 5.2.1, 5.2.2

[BW20] Braei, Mohammad ; Wagner, Sebastian: *Anomaly Detection in Univariate Time-series: A Survey on the State-of-the-Art*. https://arxiv.org/abs/2004.00433. Version: 2020

[CC08] Cryer, J.D. ; Chan, K.S.: *Time Series Analysis: With Applications in R*. Springer New York, 2008 (Springer Texts in Statistics). https://books.google.de/books?id=bHke2k-QYP4C. – ISBN 9780387759593 2.1

[CH74] Caliński, T. ; Harabasz, J: A dendrite method for cluster analysis. In: *Communications in Statistics* 3 (1974), Nr. 1, 1-27. http://dx.doi.org/10.1080/03610927408827101. – DOI 10.1080/03610927408827101 5.3.1

[Cop04] Coppin, B.: *Artificial Intelligence Illuminated*. Jones and Bartlett Publishers, 2004 (Jones and Bartlett illuminated series). https://books.google.de/books?id=LcOLqodW28EC. – ISBN 9780763732301 5.2.2

[CYPY21] Choi, Kukjin ; Yi, Jihun ; Park, Changhwa ; Yoon, Sungroh: Deep Learning for Anomaly Detection in Time-Series Data: Review, Analysis, and Guidelines. In: *IEEE Access* 9 (2021), 120043–120065. http://dx.doi.org/10.1109/access.2021.3107975. – DOI 10.1109/access.2021.3107975 1, 2.1.7, 2.1.3, 2.1.9, 2.1.3, 4.1.3, 5.2.3, 5.2.3, 5.2.3, 5.2.3

[Dan17] Dankis, Karin: *D3.9 Lighting System Design Document*. http://dx.doi.org/10.5281/zenodo.3229765. Version: sep 2017 5.1, 5.2, 5.2.1

[DB79] Davies, David L. ; Bouldin, Donald W.: A Cluster Separation Measure. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-1 (1979), Nr. 2, S. 224–227. http://dx.doi.org/

10.1109/TPAMI.1979.4766909. – DOI 10.1109/TPAMI.1979.4766909 5.3.1

[dLFRA21] de la Calle-Arroyo, Carlos ; López-Fidalgo, Jesús ; Rodríguez-Aragón, Licesio J.: Optimal designs for Antoine Equation. In: *Chemometrics and Intelligent Laboratory Systems* 214 (2021), 104334. http://dx.doi.org/https://doi.org/10.1016/j.chemolab.2021.104334. – DOI https://doi.org/10.1016/j.chemolab.2021.104334. – ISSN 0169–7439 5.2.2, 5.2.2

[DPA05] Dayan, E. ; Presnov, E. ; Albright, L.D.: METHODS TO ESTIMATE AND CALCULATE LETTUCE GROWTH. In: *Acta Horticulturae* (2005), may, Nr. 674, 305–312. http://dx.doi.org/10.17660/actahortic.2005.674.36. – DOI 10.17660/actahortic.2005.674.36 5.2.1, 5.2.2

[DPW20] Dempster, Angus ; Petitjean, François ; Webb, Geoffrey I.: ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels. In: *Data Mining and Knowledge Discovery* 34 (2020), jul, Nr. 5, 1454–1495. http://dx.doi.org/10.1007/s10618-020-00701-z. – DOI 10.1007/s10618–020–00701–z 2.1.1, 2.1.1

[EGW+09] Everingham, Mark ; Gool, Luc V. ; Williams, Christopher K. I. ; Winn, John ; Zisserman, Andrew: The Pascal Visual Object Classes (VOC) Challenge. In: *International Journal of Computer Vision* 88 (2009), sep, Nr. 2, 303–338. http://dx.doi.org/10.1007/s11263-009-0275-4. – DOI 10.1007/s11263–009–0275–4 5.3.2

[EH04] Elpelt, Barbel ; Hartung, Joachim: *Grundkurs Statistik: Lehr- und Übungsbuch der angewandten Statistik*. 3. Walter de Gruyter, 2004. – ISBN 3486204181 5.2.3, 5.2.3

[EKSX96] Ester, Martin ; Kriegel, Hans-Peter ; Sander, Jörg ; Xu, Xiaowei: A Density-Based Algorithm for Discovering Clusters in Large Spatial

Databases with Noise. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, AAAI Press, 1996 (KDD'96), S. 226–231 2.1.2, 4.1.2, 5.4.5

[FFW+19] Fawaz, Hassan I. ; Forestier, Germain ; Weber, Jonathan ; Idoumghar, Lhassane ; Muller, Pierre-Alain: Deep learning for time series classification: a review. In: *Data Mining and Knowledge Discovery* 33 (2019), mar, Nr. 4, 917–963. http://dx.doi.org/10.1007/s10618-019-00619-1. – DOI 10.1007/s10618–019–00619–1 2.1.1, 2.1.1, 2.1.1, 2.4, 4.1.3, 4.1.3

[FK15] Flach, Peter ; Kull, Meelis: Precision-Recall-Gain Curves: PR Analysis Done Right. Version: 2015. http://papers.nips.cc/paper/5867-precision-recall-gain-curves-pr-analysis-done-right.pdf. In: Cortes, C. (Hrsg.) ; Lawrence, N. D. (Hrsg.) ; Lee, D. D. (Hrsg.) ; Sugiyama, M. (Hrsg.) ; Garnett, R. (Hrsg.): *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, 838–846 5.3.2, 5.3.2

[GBC16] Goodfellow, Ian ; Bengio, Yoshua ; Courville, Aaron: *Deep Learning*. MIT Press, 2016. – http://www.deeplearningbook.org 2.1.1, 2.1.1, 2.1.5, 2.1.1, 1, 4.1.3, 4.1.3

[GCC+23] Goswami, Mononito ; Challu, Cristian ; Callot, Laurent ; Minorics, Lenon ; Kan, Andrey: Unsupervised model selection for time-series anomaly detection. In: *ICLR 2023*, 2023 1, 2.1.3, 5.2

[GSWST14] Grabocka, Josif ; Schilling, Nicolas ; Wistuba, Martin ; Schmidt-Thieme, Lars: Learning time-series shapelets. In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, aug 2014 1, 2.2, 2.3

[HB16] Hahsler, Michael ; Bolaos, Matthew: Clustering Data Streams Based on Shared Density between Micro-Clusters. In: *IEEE Transactions on Knowledge and Data Engineering* 28 (2016), jun, Nr. 6, 1449–

1461. http://dx.doi.org/10.1109/tkde.2016.2522412. – DOI 10.1109/tkde.2016.2522412 2.1.3, 2.4, 4.1.2

[HWL15] Hyndman, Rob J. ; Wang, Earo ; Laptev, Nikolay: Large-Scale Unusual Time Series Detection. In: *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, 2015, S. 1616–1619 2.1.3

[HXD03] He, Zengyou ; Xu, Xiaofei ; Deng, Shengchun: Discovering cluster-based local outliers. In: *Pattern Recognition Letters* 24 (2003), jun, Nr. 9-10, 1641–1650. http://dx.doi.org/10.1016/s0167-8655(03)00003-5. – DOI 10.1016/s0167–8655(03)00003–5

[JKY+16] Jung, Dae H. ; Kim, Damin ; Yoon, Hyo I. ; Moon, Tae W. ; Park, Kyoung S. ; Son, Jung E.: Modeling the canopy photosynthetic rate of romaine lettuce (Lactuca sativa L.) grown in a plant factory at varying CO2 concentrations and growth stages. In: *Horticulture, Environment, and Biotechnology* 57 (2016), oct, Nr. 5, 487–492. http://dx.doi.org/10.1007/s13580-016-0103-z. – DOI 10.1007/s13580–016–0103–z 5.2.2, 5.2.2, 5.2.2

[KB15] Kingma, Diederik P. ; Ba, Jimmy ; Bengio, Yoshua (Hrsg.) ; LeCun, Yann (Hrsg.): *Adam: A Method for Stochastic Optimization*. http://arxiv.org/abs/1412.6980. Version: 2015 A

[KKA+13] Kang, Jeong H. ; KrishnaKumar, Sugumaran ; Atulba, Sarah Louise S. ; Jeong, Byoung R. ; Hwang, Seung J.: Light intensity and photoperiod influence the growth and development of hydroponically grown leaf lettuce in a closed-type plant factory system. In: *Horticulture, Environment, and Biotechnology* 54 (2013), dec, Nr. 6, 501–509. http://dx.doi.org/10.1007/s13580-013-0109-8. – DOI 10.1007/s13580–013–0109–8 5.2.1, 5.2.2

[KLF06] Keogh, E ; Lin, J ; Fu, A: HOT SAX: Efficiently finding the most unusual time series subsequence. In: *Fifth IEEE International Conference on Data Mining (ICDM'05)*, IEEE, 2006

[KMRH19] Kovácsné Madar Ágota ; Rubóczki, Tímea ; Hájos, Mária T.: Lettuce production in aquaponic and hydroponic systems. In: *Acta Universitatis Sapientiae, Agriculture and Environment* 11 (2019), Nr. 1, 51–59. http://dx.doi.org/doi:10.2478/ausae-2019-0005. – DOI doi:10.2478/ausae–2019–0005 5.2.1

[LCX+21a] Li, Guozhong ; Choi, Byron ; Xu, Jianliang ; Bhowmick, Sourav S. ; Chun, Kwok-Pan ; Wong, Grace Lai-Hung: ShapeNet: A Shapelet-Neural Network Approach for Multivariate Time Series Classification. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35 (2021), may, Nr. 9, 8375–8383. http://dx.doi.org/10.1609/aaai.v35i9.17018. – DOI 10.1609/aaai.v35i9.17018 1, 2.4, 3.1, 4.1.3, 4.1.3, 4.2, 4.3, 4.3, 5.4.1, 5.4.2, 5.4.4, 5.4.6, 7, A.1, A

[LCX+21b] Li, Guozhong ; Choi, Byron ; Xu, Jianliang ; Bhowmick, Sourav S. ; Chun, Kwok-Pan ; Wong, Grace Lai-Hung: Supplementary Material: ShapeNet: A Shapelet-Neural Network Approach for Multivariate Time Series Classification. (2021), may. https://www.comp.hkbu.edu.hk/~csgzli/mtsc/ 2.2.1, A

[LHBB99] LeCun, Yann ; Haffner, Patrick ; Bottou, Léon ; Bengio, Yoshua: Object Recognition with Gradient-Based Learning. In: *Shape, Contour and Grouping in Computer Vision*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1999. – ISBN 978–3–540–46805–9, 319–345 4.1.3

[LIPJ21] Li, Jinbo ; Izakian, Hesam ; Pedrycz, Witold ; Jamal, Iqbal: Clustering-based anomaly detection in multivariate time series data. In: *Applied Soft Computing* 100 (2021), 106919. http://dx.doi.org/https://doi.org/10.1016/j.asoc.2020.106919. – DOI https://doi.org/10.1016/j.asoc.2020.106919. – ISSN 1568–4946 5.2

[LKO+12] Li, Ming ; Kozai, Toyoki ; Ohyama, Katsumi ; Shimamura, Shigeharu ; Gonda, Kaori ; Sekiyama, Tetsuo: CO2 Balance of a Commercial Closed System with Artificial Lighting for Producing Lettuce Plants. In: *HortScience horts* 47 (2012), Nr. 9, 1257 -

1260. http://dx.doi.org/10.21273/HORTSCI.47.9.1257. – DOI 10.21273/HORTSCI.47.9.1257 5.2.2, 5.2.2, 5.2.2, 5.2.2

[LKWL07] Lin, Jessica ; Keogh, Eamonn ; Wei, Li ; Lonardi, Stefano: Experiencing SAX: a novel symbolic representation of time series. In: *Data Mining and Knowledge Discovery* 15 (2007), apr, Nr. 2, 107–144. http://dx.doi.org/10.1007/s10618-007-0064-z. – DOI 10.1007/s10618–007–0064–z 4.1

[LLX+10] Liu, Yanchi ; Li, Zhongmou ; Xiong, Hui ; Gao, Xuedong ; Wu, Junjie: Understanding of Internal Clustering Validation Measures. In: *2010 IEEE International Conference on Data Mining*, 2010, S. 911–916 6.1.1

[LZPK20] Linardi, Michele ; Zhu, Yan ; Palpanas, Themis ; Keogh, Eamonn: Matrix profile goes MAD: variable-length motif and discord discovery in data series. In: *Data Min. Knowl. Discov.* 34 (2020), jul, Nr. 4, S. 1022–1071

[LZX+21] Lai, Kwei-Herng ; Zha, Daochen ; Xu, Junjie ; Zhao, Yue ; Wang, Guanchu ; Hu, Xia: Revisiting Time Series Outlier Detection: Definitions and Benchmarks. In: Vanschoren, J. (Hrsg.) ; Yeung, S. (Hrsg.): *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks* Bd. 1, Curran, 2021 2.1.3, 2.1.8, 2.1.9, 2.1.10, 2.1.3, 5.2, 5.2.3

[MCT21] Murugesan, Nivedha ; Cho, Irene ; Tortora, Cristina: Benchmarking in Cluster Analysis: A Study on Spectral Clustering, DBSCAN, and K-Means. Version: 2021. http://dx.doi.org/10.1007/978-3-030-60104-1_20. In: *Data Analysis and Rationality in a Complex World*. Springer International Publishing, 2021. – DOI 10.1007/978–3–030–60104–1$_2$0, 175 – –185 7

[MDSK17] Meinen, Esther ; Dueck, Tom ; Stanghellini, Cecilia ; Kempkes, Frank: *D4.2 Growing Fresh Food Crops in the FEG: Cultivation Recipe*. http://dx.doi.org/10.5281/zenodo.3229891. Version: aug 2017 5.2.1

[Met78] Metz, Charles E.: Basic principles of ROC analysis. In: *Seminars in Nuclear Medicine* 8 (1978), oct, Nr. 4, 283–298. http://dx.doi.org/10.1016/s0001-2998(78)80014-2. – DOI 10.1016/s0001–2998(78)80014–2 5.3.2

[MP03] Ma, J. ; Perkins, S.: Time-series novelty detection using one-class support vector machines. In: *Proceedings of the International Joint Conference on Neural Networks, 2003.* Bd. 3, 2003, S. 1741–1745 vol.3 2.1.3, 2.3, 2.4

[MRDD21] Medico, Roberto ; Ruyssinck, Joeri ; Deschrijver, Dirk ; Dhaene, Tom: Learning multivariate shapelets with multi-layer neural networks for interpretable time-series classification. In: *Advances in Data Analysis and Classification* 15 (2021), mar, Nr. 4, 911–936. http://dx.doi.org/10.1007/s11634-021-00437-8. – DOI 10.1007/s11634–021–00437–8 2.2

[NIMK20] Nakamura, Takaaki ; Imamura, Makoto ; Mercer, Ryan ; Keogh, Eamonn: MERLIN: Parameter-Free Discovery of Arbitrary Length Anomalies in Massive Time Series Archives. In: *2020 IEEE International Conference on Data Mining (ICDM)*, 2020, S. 1190–1195 2.1, 2.1

[RDN23] Rewicki, Ferdinand ; Denzler, Joachim ; Niebling, Julia: Is It Worth It? Comparing Six Deep and Classical Methods for Unsupervised Anomaly Detection in Time Series. In: *Applied Sciences* 13 (2023), Nr. 3. http://dx.doi.org/10.3390/app13031778. – DOI 10.3390/app13031778. – ISSN 2076–3417 2.1, 2.1.3, 2.1.9, 2.1.10

[RFL+20] Ruiz, Alejandro P. ; Flynn, Michael ; Large, James ; Middlehurst, Matthew ; Bagnall, Anthony: The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. In: *Data Mining and Knowledge Discovery* 35 (2020), dec, Nr. 2, 401–449. http://dx.doi.org/10.1007/s10618-020-00727-3. – DOI 10.1007/s10618–020–00727–3 2.1.1, 2.1.1, 2.1.1, 2.4

[RKB+22] Reis, Thoralf ; Kreibich, Alexander ; Bruchhaus, Sebastian ; Krause, Thomas ; Freund, Florian ; Bornschlegl, Marco ; Hemmje, Matthias: An Information System Supporting Insurance Use Cases by Automated Anomaly Detection. In: *Big*

Data and Cognitive Computing 7 (2022), 12, S. 4. http://dx.doi.org/10.3390/bdcc7010004. – DOI 10.3390/bdcc7010004

[RKV+21] Ruff, Lukas ; Kauffmann, Jacob R. ; Vandermeulen, Robert A. ; Montavon, Grégoire ; Samek, Wojciech ; Kloft, Marius ; Dietterich, Thomas G. ; Müller, Klaus-Robert: A Unifying Review of Deep and Shallow Anomaly Detection. In: *Proceedings of the IEEE* 109 (2021), Nr. 5, S. 756–795. http://dx.doi.org/10.1109/JPROC.2021.3052449. – DOI 10.1109/JPROC.2021.3052449

[Rou87] Rousseeuw, Peter J.: Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. In: *Journal of Computational and Applied Mathematics* 20 (1987), 53-65. http://dx.doi.org/https://doi.org/10.1016/0377-0427(87)90125-7. – DOI https://doi.org/10.1016/0377–0427(87)90125–7. – ISSN 0377–0427 5.3.1

[RPY+22] Ren, Yazhou ; Pu, Jingyu ; Yang, Zhimeng ; Xu, Jie ; Li, Guofeng ; Pu, Xiaorong ; Yu, Philip S. ; He, Lifang: *Deep Clustering: A Comprehensive Survey*. http://dx.doi.org/10.48550/ARXIV.2210.04142. Version: 2022 2.4, 4.1

[RRS00] Ramaswamy, Sridhar ; Rastogi, Rajeev ; Shim, Kyuseok: Efficient Algorithms for Mining Outliers from Large Data Sets. In: *SIGMOD Rec.* 29 (2000), may, Nr. 2, 427–438. http://dx.doi.org/10.1145/335191.335437. – DOI 10.1145/335191.335437. – ISSN 0163–5808

[Sas07] Sasaki, Yutaka: The truth of the F-measure. In: *Teach Tutor Mater* (2007), 01 5.3.2

[SC07] Salvador, Stan ; Chan, Philip: Toward Accurate Dynamic Time Warping in Linear Time and Space. In: *Intell. Data Anal.* 11 (2007), oct, Nr. 5, S. 561–580. – ISSN 1088–467X 2.1.1

[Sch17] Schubert, D.: Greenhouse production analysis of early mission scenarios for Moon and Mars habitats. In: *Open Agriculture* 2 (2017), feb, Nr. 1, 91–115. http://dx.doi.org/10.1515/opag-2017-0010. – DOI 10.1515/opag–2017–0010 5.1, A

[SEKX98] Sander, Jörg ; Ester, Martin ; Kriegel, Hans-Peter ; Xu, Xiaowei: Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. In: *Data Mining and Knowledge Discovery* 2 (1998), Nr. 2, 169–194. http://dx.doi.org/10.1023/a:1009745219419. – DOI 10.1023/a:1009745219419 2.1.2, 4.1.2, 5.4.5

[SJMY17] Song, Hongchao ; Jiang, Zhuqing ; Men, Aidong ; Yang, Bo: A hybrid semi-supervised anomaly detection model for high-dimensional data. In: *Comput. Intell. Neurosci.* 2017 (2017), nov, S. 8501683. http://dx.doi.org/10.1155/2017/8501683. – DOI 10.1155/2017/8501683

[SKF08] SHIMIZU, Hiroshi ; KUSHIDA, Megumi ; FUJINUMA, Wataru: A Growth Model for Leaf Lettuce under Greenhouse Environments. In: *Environmental Control in Biology* 46 (2008), Nr. 4, S. 211–219. http://dx.doi.org/10.2525/ecb.46.211. – DOI 10.2525/ecb.46.211 5.2.1

[SL17] Schäfer, Patrick ; Leser, Ulf: *Multivariate Time Series Classification with WEASEL+MUSE*. http://arxiv.org/abs/1711.11343. Version: 2017 2.1.1

[SSE+17] Schubert, Erich ; Sander, Jörg ; Ester, Martin ; Kriegel, Hans P. ; Xu, Xiaowei: DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN. In: *ACM Trans. Database Syst.* 42 (2017), jul, Nr. 3. http://dx.doi.org/10.1145/3068335. – DOI 10.1145/3068335. – ISSN 0362–5915 2.1.2, 4.1.2, 4.1, 5.4.5, A

[SWP22] Schmidl, Sebastian ; Wenig, Phillip ; Papenbrock, Thorsten: Anomaly detection in time series. In: *Proceedings of the VLDB Endowment* 15 (2022), may, Nr. 9, 1779–1797. http://dx.doi.org/10.14778/3538598.3538602. – DOI 10.14778/3538598.3538602 1, 2.1, 2.1, 2.1.3, 2.1.3

[TCFC02] Tang, Jian ; Chen, Zhixiang ; Fu, Ada W. ; Cheung, David W.: Enhancing Effectiveness of Outlier Detections for Low Density Patterns. Version: 2002. http://dx.doi.org/10.1007/3-540-47887-6_53. In: *Advances in Knowledge Discovery and Data Mining*. Springer Berlin Heidelberg, 2002. – DOI 10.1007/3–540–47887–$6_5$3, 535 – –548

[van80] van Holsteijn, H.M.C.: *Growth of lettuce: Quantitative analysis of growth.* Veenman, 1980 (Mededelingen / Landbouwhogeschool Wageningen 80-13). https://edepot.wur.nl/287596. – Met lit. opg 5.2.1

[van81] van Holsteijn, H.M.C.: *Growth and photosynthesis of lettuce*, Diss., 1981. https://edepot.wur.nl/290026. – WU thesis 868 Proefschrift Wageningen 5.2.2, 5.2.2, 5.2.2

[van19] (2019), jun. http://dx.doi.org/10.1080/20464177.2019.1633223. – DOI 10.1080/20464177.2019.1633223 2.1.3

[VBBD17] Valenzuela, Ira C. ; Baldovino, Renann G. ; Bandala, Argel A. ; Dadios, Elmer P.: Optimization of Photosynthetic Rate Parameters using Adaptive Neuro-Fuzzy Inference System (ANFIS). In: *2017 International Conference on Computer and Applications (ICCA)*, 2017, S. 129–134 5.2.2, 5.2.2, 5.2.2, 5.2.2

[WMS+94] Wheeler, R.M. ; Mackowiak, C.L. ; Sager, J.C. ; Yorio, N.C. ; Knott, W.M. ; Berry, W.L.: Growth and Gas Exchange by Lettuce Stands in a Closed, Controlled Environment. In: *Journal of the American Society for Horticultural Science jashs* 119 (1994), Nr. 3, 610 - 615. http://dx.doi.org/10.21273/JASHS.119.3.610. – DOI 10.21273/JASHS.119.3.610 5.2.1, 5.2.2, 5.2.2

[yan18] Multi-view clustering: A survey. In: *Big Data Mining and Analytics* 1 (2018), jun, Nr. 2, 83–107. http://dx.doi.org/10.26599/bdma.2018.9020003. – DOI 10.26599/bdma.2018.9020003 2.4

[YK09] Ye, Lexiang ; Keogh, Eamonn: Time Series Shapelets: A New Primitive for Data Mining. In: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* New York, NY, USA : Association for Computing Machinery, 2009 (KDD '09). – ISBN 9781605584959, 947–956 1, 2.2, 2.2, 2.2, 2.3, 2.3

[YKH01] Yairi, Takehisa ; Kato, Yoshikiyo ; Hori, Koichi: Fault detection by mining association rules from house-keeping data. In: *Proceedings of the International Symposium on Artificial Intelligence*, Canadian Space Agency, 2001 2.1.3, 2.4, 4.4

[YS20] Yang, Li ; Shami, Abdallah: On Hyperparameter Optimization of Machine Learning Algorithms: Theory and Practice. (2020). http://dx.doi.org/10.1016/j.neucom.2020.07.061. – DOI 10.1016/j.neucom.2020.07.061 5.4.1

[YSGG10] Yao, Yuan ; Sharma, Abhishek ; Golubchik, Leana ; Govindan, Ramesh: Online anomaly detection for sensor systems: A simple and efficient approach. In: *Perform. Eval.* 67 (2010), nov, Nr. 11, S. 1059–1075

[YZU+18] Yeh, Chin-Chia M. ; Zhu, Yan ; Ulanova, Liudmila ; Begum, Nurjahan ; Ding, Yifei ; Dau, Hoang A. ; Zimmerman, Zachary ; Silva, Diego F. ; Mueen, Abdullah ; Keogh, Eamonn: Time Series Joins, Motifs, Discords and Shapelets: A Unifying View That Exploits the Matrix Profile. In: *Data Min. Knowl. Discov.* 32 (2018), jan, Nr. 1, 83–123. http://dx.doi.org/10.1007/s10618-017-0519-9. – DOI 10.1007/s10618–017–0519–9. – ISSN 1384–5810

[ZBV+15] Zabel, Paul ; Bamsey, Matthew ; Vrakking, Vincent ; Zeidler, Conrad ; Schubert, Daniel ; Rettberg, Petra ; Barczyk, Simon ; Davenport, Bob ; Waclavicek, Rene ; Battistelli, Alberto ; Nazzaro, Filomena ; Stasiak, Mike ; Kohlberg, Eberhard ; Mazzoleni, Erik ; Fetter, Viktor ; Boscheri, Giorgio ; Guarnieri, Enzo ; Locantore, Ilaria ; Bonzano, Giuseppe ; Kempkes, Frank ; Dueck, Tom ; Stanghellini, Cecilia ; Gilley, Anthony ; Bennett, Michelle ; Downey, Peter ; Larkin, Tracey ; Ceriello, Antonio ; Fortezza, Raimondo: *D2.5 Design Report*. http://dx.doi.org/10.5281/zenodo.3229731. Version: dec 2015 1, 5.1, 5.1, 5.2, 5.2.1, 5.2.1, 5.2.2, 5.2.2, 5.2.2, 5.2.2, 5.4.3

[Zhu04] Zhu, Mu: Recall, precision and average precision. In: *Department of Statistics and Actuarial Science* 2 (2004)

[ZLW22] Zhou, Jing ; Li, Pingping ; Wang, Jizhang: Effects of Light Intensity and Temperature on the Photosynthesis Characteristics and Yield of Lettuce. In: *Horticulturae* 8 (2022), Nr. 2. http://dx.doi.org/10.3390/horticulturae8020178. – DOI 10.3390/horticulturae8020178. – ISSN 2311–7524 5.2.2, 5.2.2, 5.2.2

[ZMK12] Zakaria, Jesin ; Mueen, Abdullah ; Keogh, Eamonn: Clustering Time Series Using Unsupervised-Shapelets. In: *2012 IEEE 12th International Conference on Data Mining*, 2012. – ISBN 978–1–4673–4649–8, S. pp. 785–794 1, 2.2.1, 2.2

[ZS23] Zhang, Nan ; Sun, Shiliang: Multiview Unsupervised Shapelet Learning for Multivariate Time Series Clustering. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45 (2023), Nr. 4, S. 4981–4996. http://dx.doi.org/10.1109/TPAMI.2022.3198411. – DOI 10.1109/TPAMI.2022.3198411 1, 2.1.2, 2.2, 2.2

[ZZ19] Zabel, Paul ; Zeidler, Conrad: *D5.3 Environmental Control Performance*. http://dx.doi.org/10.5281/zenodo.3229923. Version: apr 2019 5.2.3

[ZZS+16] Zhu, Yan ; Zimmerman, Zachary ; Senobari, Nader S. ; Yeh, Chin-Chia M. ; Funning, Gareth ; Mueen, Abdullah ; Brisk, Philip ; Keogh, Eamonn: Matrix profile II: Exploiting a novel algorithm and GPUs to break the one hundred million barrier for time series motifs and joins. In: *2016 IEEE 16th International Conference on Data Mining (ICDM)*, IEEE, dec 2016

[ZZSG22] Zhang, Jitao ; Zeng, Bing ; Shen, Weiming ; Gao, Liang: A one-class Shapelet dictionary learning method for wind turbine bearing anomaly detection. In: *Measurement* 197 (2022), 111318. http://dx.doi.org/https://doi.org/10.1016/j.measurement.2022.111318. – DOI https://doi.org/10.1016/j.measurement.2022.111318. – ISSN 0263–2241 2.3

Declaration Of Authorship

I, Tobias Merlin Fischer, hereby affirm, that I have prepared the present work without the unauthorized help of third parties and without the use of any aids other than those specified. The data and concepts taken directly or indirectly from sources are marked with a reference to the source. This thesis has not been submitted to an examination board in the same or a similar form, or published in any other way, either at home or abroad.

Ilmenau, 24.07.2023

Tobias Merlin Fischer