Data-Driven Methods for Managing Anomalies in Energy Time Series

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

von der KIT-Fakultät für Informatik

des Karlsruher Instituts für Technologie (KIT)

genehmigte

Dissertation

von

Marian Turowski

Tag der mündlichen Prüfung:	28.11.2022
Referenten:	Prof. Dr. Veit Hagenmeyer
	Prof. Dr.-Ing. Jorge Ángel González Ordiano
	apl. Prof. Dr.-Ing. Ralf Mikut

Marian Turowski
Institute for Automation and Applied Informatics
Karlsruhe Institute of Technology
Eggenstein-Leopoldshafen
Germany

DOI 10.5445/IR/1000154434

Acknowledgements

As completing this dissertation has been a challenging journey for me, I am grateful to all those who have supported me along this journey, in small and large ways.

First and foremost, I would like to thank my supervisors. I thank Veit Hagenmeyer for his guidance, inspiration, and encouragement as well as the opportunities to learn, grow, and contribute in research, teaching, and organizational development. I also thank Ralf Mikut for the numerous intense discussions, his openness and ideas, and the extensive feedback.

Moreover, I would like to thank Jorge Ángel González Ordiano for being a reviewer of my dissertation and for participating in my defense via video conference. I also thank Ralf Reussner for his participation in my doctoral committee.

I would not have been able to pursue this dissertation without the financial support I received. Therefore, I would like to thank the State of Baden-Württemberg and the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI for the provided support.

I would also like to extend my thanks to all my colleagues at the Automated Image and Data Analysis group and later Machine Learning for Time Series and Images group, namely Andreas, Ángel, Benjamin, Christian, Friedrich, Ines, Jan, Kaleb, Katharina, Lisa, Luca, Markus, Matthias, Moritz, Nicole, Oliver, Vojtech, Simon, Stefan, Tim, and Yanke. Thanks also to all my colleagues at IAI, especially Andreas, Anne, Benedikt, Bernadette, Claudia, Dominique, Eric, Fabian, Friedrich, Jan, Kai, Kemal, Luigi, Moritz, Richard, and Tom.

A special thanks goes to my office mates Nicole, Kaleb, and Katharina. I am very thankful for all the discussions, laughter, and support before, during, and after the pandemic.

Another special thanks to all my co-authors, collaborators, and students with whom I had the pleasure of working. I am grateful for the fruitful collaborations leading to many successful joint projects. I would especially like to mention Benedikt, Kaleb, Kai, Moritz, Nicole, and Oliver who have accompanied and supported me in my pursuit of relevant, valuable, and comprehensible research.

Finally, I am deeply grateful that I have the privilege of having such great family and friends. An extraordinary thanks to my parents, my sister, my girlfriend, my dancing partners, and my friends for always believing in me and supporting me in difficult times.

Karlsruhe, November 2022 *Marian Turowski*

Abstract

With the progressing implementation of the smart grid, more and more smart meters record power or energy consumption and generation as time series. The increasing availability of these recorded energy time series enables the goal of the automated operation of smart grid applications such as load analysis, load forecasting, and load management. However, to perform well, these applications usually require clean data that describes the typical behavior of the underlying system well. Unfortunately, recorded energy time series are usually not clean but contain anomalies, i. e., patterns that deviate from what is considered normal. Since anomalies thus potentially contain data points or patterns that represent false or misleading information, they can be problematic for any analysis of this data performed by smart grid applications.

Therefore, the present thesis proposes data-driven methods for managing anomalies in energy time series. It introduces an anomaly management whose characteristics correspond to steps in a sequential pipeline, namely anomaly detection, anomaly compensation, and a subsequent application. Using forecasting as an exemplary subsequent application and real-world data with inserted synthetic and labeled anomalies, this thesis answers four research questions along that pipeline for managing anomalies in energy time series. Based on the answers to these four research questions, the anomaly management presented in this thesis exhibits four characteristics. First, the presented anomaly management is guided by well-defined anomalies derived from real-world energy time series. These anomalies serve as a basis for generating synthetic anomalies in energy time series to promote the development of powerful anomaly detection methods. Second, the presented anomaly management applies an anomaly detection approach to energy time series that is capable of providing a high anomaly detection performance. Third, the presented anomaly management also compensates detected anomalies in energy time series realistically by considering the characteristics of the respective data. Fourth, the proposed anomaly management applies and evaluates general anomaly management strategies in view of the subsequent forecasting that uses this data. The comparison shows that managing anomalies well is essential, as the compensation strategy, which detects and compensates anomalies in the input data before applying a forecasting method, is the most beneficial strategy when the input data contains anomalies.

Kurzfassung

Im Zuge der fortschreitenden Realisierung des intelligenten Stromnetzes zeichnen immer mehr intelligente Zähler die verbrauchte oder erzeugte Leistung oder Energie als Zeitreihe auf. Die zunehmende Verfügbarkeit dieser aufgezeichneten Energiezeitreihen ermöglicht den automatisierten Betrieb von Anwendungen des intelligenten Stromnetzes wie Lastanalysen, Lastprognosen und Lastmanagement. Damit diese Anwendungen gut funktionieren, benötigen sie jedoch in der Regel saubere Daten, die das typische Verhalten des zugrunde liegenden Systems richtig beschreiben. Leider sind die aufgezeichneten Energiezeitreihen in der Regel nicht sauber, sondern enthalten Anomalien, d.h. Muster, die von dem abweichen, was als normal angesehen wird. Da Anomalien also potenziell Datenpunkte oder Muster enthalten, die falsche oder irreführende Informationen darstellen, können sie für jegliche Analyse dieser Daten durch Anwendungen des intelligenten Stromnetzes problematisch sein.

In der vorliegenden Arbeit werden daher datengetriebene Methoden für das Management von Anomalien in Energiezeitreihen vorgestellt. Es wird dabei ein Anomaliemanagement eingeführt, dessen Merkmale den Schritten einer sequentiellen Pipeline entsprechen, nämlich der Erkennung von Anomalien, der Kompensation von Anomalien und einer nachfolgenden Anwendung. Anhand von Prognosen als beispielhafter Folgeanwendung und realen Daten mit eingefügten synthetischen und markierten Anomalien beantwortet diese Arbeit vier Forschungsfragen entlang dieser Pipeline zum Management von Anomalien in Energiezeitreihen. Ausgehend von den Antworten auf diese vier Forschungsfragen weist das in dieser Arbeit vorgestellte Anomaliemanagement vier Merkmale auf. Erstens orientiert sich das vorgestellte Anomaliemanagement an genau definierten Anomalien, die aus realen Energiezeitreihen abgeleitet werden. Diese Anomalien dienen als Grundlage für die Erzeugung synthetischer Anomalien in Energiezeitreihen, um die Entwicklung leistungsfähiger Anomalieerkennungsmethoden zu fördern. Zweitens wendet das vorgestellte Anomaliemanagement einen Ansatz zur Erkennung von Anomalien auf Energiezeitreihen an, der eine hohe Leistung bei der Erkennung von Anomalien bietet. Drittens kompensiert das vorgestellte Anomaliemanagement auch erkannte Anomalien in Energiezeitreihen realistisch, indem es die Eigenschaften der jeweiligen Daten berücksichtigt. Viertens wendet das vorgeschlagene Anomaliemanagement allgemeine Strategien zum Anomaliemanagement an und bewertet sie im Hinblick auf die anschließende Vorhersage, die diese Daten nutzt. Der Vergleich zeigt, dass ein gutes Anomaliemanagement von wesentlicher Bedeutung ist, da die Kompensationsstrategie, die Anomalien in den Eingabedaten vor der Anwendung einer Prognosemethode erkennt und kompensiert, die vorteilhafteste Strategie ist, wenn die Eingabedaten Anomalien enthalten.

Contents

List of Figures

List of Tables

Abbreviations

AE	Autoencoder
cINN	conditional Invertible Neural Network
CPI	Copy-Paste Imputation
cVAE	conditional Variational Autoencoder
GAN	Generative Adversarial Network
iForest	Isolation Forest
INN	Invertible Neural Network
KIT	Karlsruhe Institute of Technology
kNN	k-Nearest Neighbor
LinR	Linear Regression
LOF	Local Outlier Factor
LogR	Logistic Regression
MAPE	Mean Absolute Percentage Error
MLP	Multi-Layer Perceptron
NB	Gaussian Naïve Bayes
NN	Neural Network
PNN	Profile Neural Network
RF	Random Forest
RMSE	Root Mean Squared Error
SVC	Support Vector Machine for Classification
SVR	Support Vector Regression
t-SNE	t-distributed Stochastic Neighbor Embedding
TRTR	Train Real Test Real
TSTR	Train Synthetic Test Real
VAE	Variational Autoencoder
WAPE	Weighted Absolute Percentage Error

1 Introduction

In view of climate change as a serious threat to life on earth, the countries of the world agreed on the Paris Agreement. It specifies to take measures to limit the increase in global mean temperature to well below 2 °C and preferably below 1.5 °C compared to preindustrial values (*Paris Agreement* 2015). Since anthropogenic greenhouse gas emissions are considered to be the main cause of the observed temperature rise, strong and timely measures are necessary to reduce global greenhouse gas emissions and to reach the agreed net zero emissions by 2050 (Skea et al. 2022). In addition to the observed rise in temperature, it also becomes increasingly apparent that global resources are finite and that resource use must be changed (Giljum and Hinterberger 2014; Shafiee and Topal 2009).

The selection of the energy sources, the amount of energy used, and the efficiency of its use strongly influence both the greenhouse gas emissions and the use of finite resources. Conventional energy supply based on converting the inner energy of finite resources such as hard coal or natural gas into, for example, mechanical or electrical energy causes tremendous greenhouse gas emissions. For this reason, several societies aim to reduce the environmental impact of their energy supply and use through the increased use of so-called renewable energy sources such as wind and sun.

Transitioning from conventional to renewable energy sources, i. e., increasing the share of renewable energy sources in energy supply, implies changes in the corresponding energy system. A core change and at the same time enabler of the increased use of renewable energy sources is the implementation of the smart grid (Fang et al. 2012). It aims to intelligently connect and control all elements of an energy system using information and communication technology (Fang et al. 2012; Ipakchi and Albuyeh 2009; Li et al. 2010a). To acquire relevant information, smart grids use, among others, smart meters that are installed at consumers and producers (Alahakoon and Yu 2016). Smart meters record and transmit various information as time series, including voltage, reactive power, and power or energy consumption and generation (Alquthami et al. 2020).

Since more and more smart meters are installed, the number of recorded energy time series increases, enabling and opening up a wide range of possible applications for this data. Exemplary applications include customer profiling, load analysis, load forecasting, and load management (Rossi and Chren 2020; Wang et al. 2019), but also grid simulations for stability analyses, grid development, fault-detection, and efficiency improvements as well as research platforms to develop technologies for the grid of the

future (Hagenmeyer et al. 2016). In addition, the increasing availability of recorded energy time series enables the goal of automated operation of the mentioned applications (see e.g., Capozzoli et al. 2018; Meisenbacher et al. 2022). However, to perform well, these applications usually require clean data that describes the typical behavior of the underlying system well (Luo et al. 2018c; Wang et al. 2019).

Unfortunately, recorded time series are usually not clean, but contain anomalies (Chen et al. 2017). Anomalies are patterns that deviate from what is considered normal (Chandola et al. 2009). They can occur in energy time series for many reasons, including smart meter failures (Wang et al. 2020), abnormal user behavior (Nordahl et al. 2017), unusual consumption (Seem 2007), and energy theft (Jokar et al. 2016). All anomalies have in common that they potentially contain data points or patterns that represent false or misleading information, which can be problematic for any analysis of this data performed by the mentioned applications (Wang et al. 2019). For example, anomalies such as positive or negative spikes may strongly deviate from what is considered normal and a subsequent forecasting method using the data as input, may generate an incorrect forecast, which in turn could lead to an inappropriate energy schedule and ultimately affect the stability of the energy system in an automated smart grid setting.

Therefore, managing anomalies in energy time series is an important issue in energy systems. To effectively manage anomalies, an anomaly management for energy time series ideally models the relevant anomalies or what is considered normal and is able to choose an appropriate strategy to deal with the anomalies. To apply anomaly management strategies, the anomaly management commonly has to consider the typically used anomaly detection and anomaly compensation. Moreover, the anomaly management should select and apply the appropriate strategy considering the subsequent application thus providing suitable data to this application. Furthermore, this anomaly management ideally leverages the increasingly available energy time series and the information contained therein.

While works in other domains such as cybersecurity (Donevski and Zia 2019) and space systems (Kitts 2006) design such an anomaly management, energy system research has not yet taken this comprehensive view. For example, some works consider the management of anomalies in the context of the predictive maintenance of technical or industrial facilities such as power plants (e.g., Lee et al. 2017; Zhao et al. 2021), which mainly addresses the anomaly detection of the described anomaly management. In a similar manner, Samitier (2017) describes how to manage faults and anomalies in the communication networks and services of utilities, which takes up the mentioned anomaly detection and anomaly compensation. In literature using energy time series, researchers also mostly consider only one element, such as anomaly detection (e.g., Himeur et al. 2021), or at most few of the described elements of an anomaly management. Energy management solutions for buildings, for example, only model and detect anomalies to reduce the energy consumption (e.g., Capozzoli et al. 2018; Khalilnejad et al. 2020; Markus et al. 2021; Zhu et al. 2019). Similarly, Akouemo and Povinelli (2017) apply anomaly detection and compensation to improve the data quality used for a subsequent forecast. However, no research on managing anomalies in energy time series takes a data-driven approach to develop an anomaly management that coherently considers modeled anomalies, anomaly

detection, and anomaly compensation for energy time series in view of general anomaly management strategies and a potential subsequent application.

1.1 Research Questions and Contributions

The present thesis aims to provide data-driven methods for managing anomalies in energy time series. It proposes an anomaly management for energy time series that exhibits the following four characteristics: First, the proposed anomaly management is guided by well-defined anomalies observed in real-world energy time series, which can serve as a basis for generating synthetic anomalies in energy time series to promote the development of powerful anomaly detection methods. Second, the proposed anomaly management applies an anomaly detection to energy time series that is capable of providing a high anomaly detection performance. Third, the proposed anomaly management compensates detected anomalies in energy time series realistically by considering the characteristics of the respective data. Fourth, the proposed anomaly management considers and evaluates general anomaly management strategies in view of a subsequent application that uses this data.

The described characteristics for managing anomalies in energy time series correspond to sequential steps that can be organized in a pipeline similar to the common process of designing time series forecasts (Meisenbacher et al. 2022). As illustrated in Figure 1.1, the corresponding pipeline comprises three steps, namely anomaly detection, anomaly compensation, and a subsequent application such as load analysis, load forecasting, or load management. More specifically, the pipeline starts with the detection of anomalies in a given time series containing anomalies using an anomaly detection method. The effective application and evaluation of this anomaly detection method requires the prior identification and modeling or labeling of the anomalies to be detected and results in a time series with detected anomalies. The detected anomalies are then compensated with an anomaly compensation method. The time series with compensated anomalies finally serves as an input for a subsequent application.

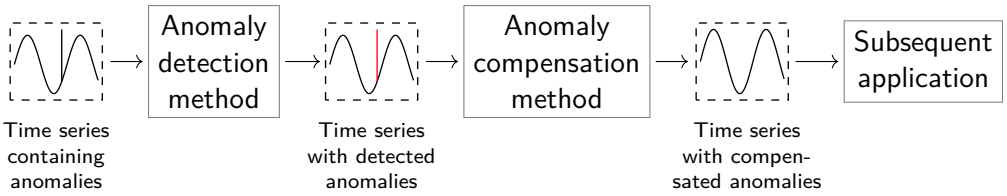


Figure 1.1 Pipeline for managing anomalies in energy time series. The pipeline starts with the detection of anomalies in a given time series containing anomalies using an anomaly detection method, resulting in a time series with detected anomalies. The detected anomalies are then compensated with an anomaly compensation method. The time series with compensated anomalies finally serves as an input for a subsequent application.

For the present thesis, we choose forecasting as exemplary subsequent application of the proposed anomaly management since it is a representative and relevant application

that is suitable for anomaly management. Forecasting is a highly relevant application in energy systems faced with volatile renewable energy sources as it allows for good planning, scheduling, and thus balanced and stable operation of the respective grid (González Ordiano et al. 2018). At the same time, it is important to anticipate that the energy time series used as input to forecasting methods can be inaccurate and can contain anomalies (Akouemo and Povinelli 2017; Luo et al. 2018c). Anomalies can severely affect time series forecasts, leading to erroneous forecasts in the worst case and making anomaly management particularly crucial (Chen and Liu 1993; Petropoulos et al. 2022).

Given that forecasting serves as an example of a subsequent application, we adapt the pipeline shown in Figure 1.1 accordingly. For this, we simply replace the subsequent application with a forecasting method so that the time series with compensated anomalies serves as an input to this forecast method. The forecast method then provides a forecast based on that input time series. The resulting pipeline comprises the steps that one can use to manage anomalies in energy time series forecasting.

Along this pipeline, the present thesis addresses four research questions on data-driven methods for managing anomalies (see Figure 1.2). In the following, we introduce each research question in detail and describe what is required to answer these questions given the information contained in the data.

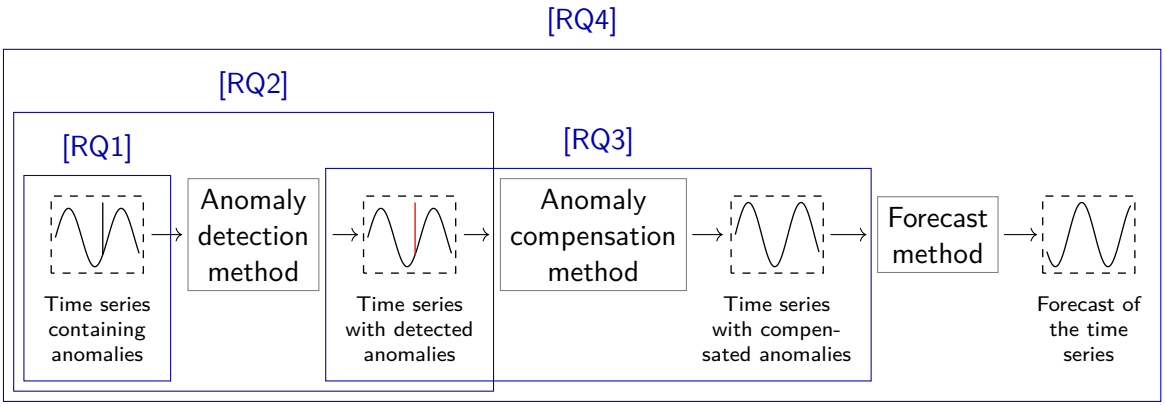


Figure 1.2 The four research questions [RQ1] to [RQ4] on data-driven methods for managing anomalies that the present thesis addresses along the pipeline for managing anomalies in energy time series forecasting.

Research Question 1 [RQ1]: *How can anomalies in energy time series be modeled and generated to improve anomaly detection?*

Anomaly management benefits from an anomaly detection that reliably identifies all relevant anomalies in an energy time series. Anomaly detection, in turn, profits from a precise knowledge of the anomalies. However gaining this precise knowledge is challenging. Common approaches to achieve this aim are manually labeling anomalies, defining a minority of the data as anomalous, or reproducing anomaly-free or labeled time series. However, all of these approaches are dependent on the considered data set and the number and location of the contained anomalies. Furthermore, these approaches require

expert knowledge throughout the process, which can be costly and time-consuming. These drawbacks therefore can limit the applicability of these approaches in developing anomaly detection methods for an anomaly management.

To improve the detection of anomalies in energy time series for anomaly management, we therefore aim to accurately model anomalies in real-world time series and to provide means to intentionally generate them. This approach should include identifying typical anomalies in both time series containing energy measurements and time series containing power measurements, before modeling the identified anomalies including parameters derived from real-world data. The modeled anomalies should serve as the basis for generating synthetic anomalies in time series containing energy measurements or time series containing power measurements. Taking into account these aspects, the present thesis contributes to this research question by proposing a method that is capable of generating synthetic anomalies that are modeled on real-world anomalies and that can be inserted in arbitrary quantity at random points of time into an arbitrary energy or power time series.

Research Question 2 [RQ2]: *How can anomaly detection methods for energy time series be enhanced?*

Before anomalies can be managed to limit their potential impact, we need to detect them in a given energy time series so that we obtain a time series with detected anomalies. While a large variety of methods addresses the detection of anomalies, the focus mostly lies on improving a method's individual performance. The aim of finding a single high performing anomaly detection method may be advantageous for certain use cases, but it also comes with limitations. First, in a given use case, it may be necessary to search for the best hyperparameters of an anomaly detection method. Second, one has to compare available methods in order to eventually select the best performing anomaly detection method. This hyperparameter search and selection process might be too time-consuming and costly in the context of anomaly management, even when using existing data-driven anomaly detection methods that work independent from expert knowledge.

To provide a solid foundation for the application in anomaly management, we thus aim to generally enhance anomaly detection methods for energy time series using a data-driven approach that does not require additional input from an expert. This aim implies that the enhancement considers both supervised and unsupervised anomaly detection methods. Moreover, the data-driven enhancement should take into account that various anomalies can occur in energy time series. While some anomalies such as the anomalies previously modeled in research question [RQ 1] can violate the underlying distribution corresponding to normal behavior and can be easily recognized by a human, other anomalies remain in the underlying distribution and are hard to detect. Considering these aspects, the present thesis contributes to this research question by introducing a novel approach that generally enhances anomaly detection methods for energy time series.

Research Question 3 [RQ3]: *How can anomalies detected in energy time series be compensated?*

Once anomalies are detected, they need to be compensated to obtain an energy time series that better reflects the actual normal behavior. When detected anomalies are regarded as missing values, compensating anomalies can be considered as imputation, where missing data in a time series is replaced with substitution values. While various methods address the imputation of time series, these methods are mostly only designed for imputing missing values in time series containing power measurements. Moreover, none of these methods considers inherent properties of time series containing energy measurements. As a result, detected anomalies compensated by imputation may result in time series that exhibit atypical patterns and do not match the total recorded energy, thus limiting their applicability in an anomaly management for energy time series where human intervention should be restricted.

To provide a good basis for the application in an anomaly management, we therefore aim to compensate anomalies detected in energy time series with a realistic imputation using information contained in the data. This implies that the imputation preserves the inherent patterns of the respective time series. Moreover, the imputation should consider the total recorded energy and should guarantee that it remains unchanged during the imputation. Considering these aspects, the present thesis contributes to this research question by a novel imputation method for time series containing energy measurements that uses matching patterns and preserves the total energy of each part with missing values.

Research Question 4 [RQ4]: *How can an anomaly management account for anomalies in energy time series forecasting?*

Given that anomalies detected in an energy time series are compensated, the resulting time series with compensated anomalies can serve as an input for a forecast method that then calculates a forecast. However, using a time series with compensated anomalies as input is only one strategy for managing anomalies in energy time series forecasting. Another general strategy is to only apply an anomaly detection method and to let the applied forecasting method use the information about detected anomalies. An alternative general strategy is to apply neither an anomaly detection method nor an anomaly compensation method, but to use a forecasting method that is known to be robust against anomalies. Furthermore, one could simply rely on the assumption that the anomalies possibly contained in the time series do not strongly influence the forecast, and thus use the respective time series unchanged as input for the applied forecasting method. While all of these strategies aim to achieve a good forecast accuracy, it is not known which strategy is actually the best. Despite being essential for anomaly management, a rigorous comparison of the available strategies for managing anomalies in energy time series forecasting is lacking.

To provide a sound basis for managing anomalies in energy time series forecasting, we therefore aim to take on the typically used strategies mentioned above and describe general strategies. Given these described strategies, matching representative forecasting methods should be selected and compared on real-world data to determine the best anomaly

management strategy. The comparison should consider that some of the strategies use the same input data but different forecasting methods and could be compared beforehand to keep the number of strategies in the overall comparison manageable. Furthermore, the comparison should take into account that certain strategies principally include a choice between a supervised and an unsupervised detection method. Considering these aspects, the present thesis contributes to this research question by proposing and comparing different general strategies for managing anomalies in energy time series forecasting to identify the most beneficial strategy.

1.2 Outline

The present thesis is organized along the above mentioned research questions [RQ1]-[RQ4] (see Figure 1.3). Before addressing these research questions, we introduce in Chapter 2 the fundamental concepts on which this thesis is based. Afterward, we start with research question [RQ1]. In Chapter 3, we describe how we model anomalies in energy time series and generate synthetic anomalies. Using these synthetic anomalies, we proceed with research question [RQ2] in Chapter 4, which focuses on improving the detection of anomalies in energy time series. Afterward, we continue with research question [RQ3] in Chapter 5 where we present how we compensate previously detected anomalies in energy time series. Since the answers on the research questions [RQ1], [RQ2] and [RQ3] contribute to the answer on research question [RQ4], we finally turn to research question [RQ4]. Chapter 6 brings the previous chapters together and considers strategies for managing anomalies in energy time series forecasting. Lastly, we discuss this thesis and its contributions in Chapter 7, before we conclude it with a summary and an outlook in Chapter 8.

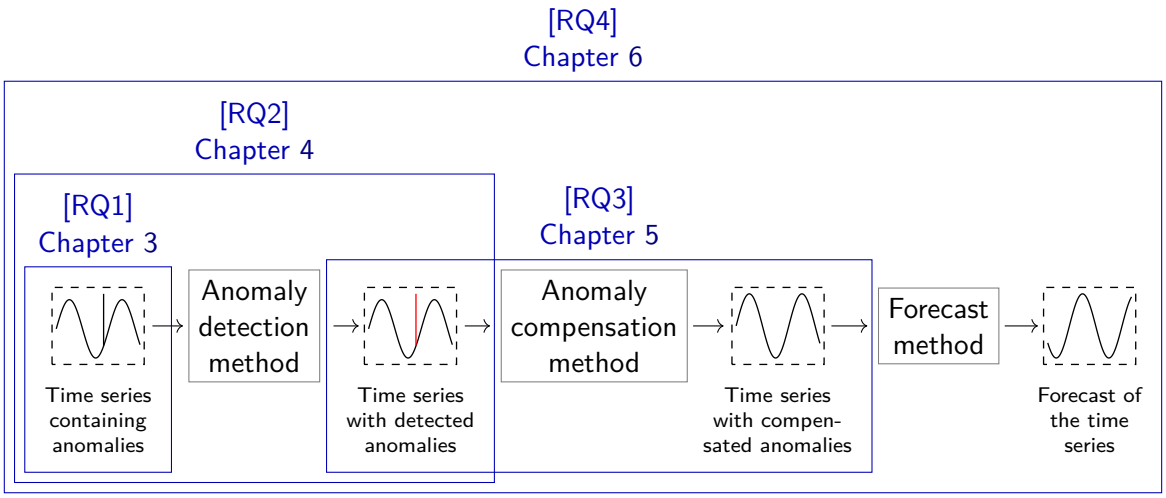


Figure 1.3 Organization of the present thesis along the research questions [RQ1] to [RQ4].

2 Preliminaries

In order to answer the research questions mentioned above, we need to introduce some fundamental concepts on which this thesis is based. After defining energy time series, we describe anomalies in energy time series that are the main object of investigation in this thesis.

2.1 Energy Time Series

In the present thesis, we consider a time series to be a set of sequential observations obtained at regular intervals of time (Hyndman and Athanasopoulos 2021). Formally, a time series is defined as follows:

Definition 2.1 (Time Series). The observations $x_1, x_2, \ldots, x_N$ obtained at the equidistant points of time $1, 2, \ldots, N$ form the time series $X = x_1, x_2, \ldots, x_N$ with the length N.

In the context of energy systems, multiple time series exist including frequency measurements, energy prices, and weather observations. From available time series in the energy system, this thesis focuses on energy time series[1] that contain observations of various forms of energy such as gas, heat, or electricity. In the smart grid, these forms of energy are typically measured with smart meters (Mohassel et al. 2014). Smart meters usually meter the amount of energy consumed or generated up to a certain point in time or, less often, the average power consumption or production in a given period of time, resulting in energy measurements or power measurements. In this thesis, we define time series containing energy or power measurements as follows.

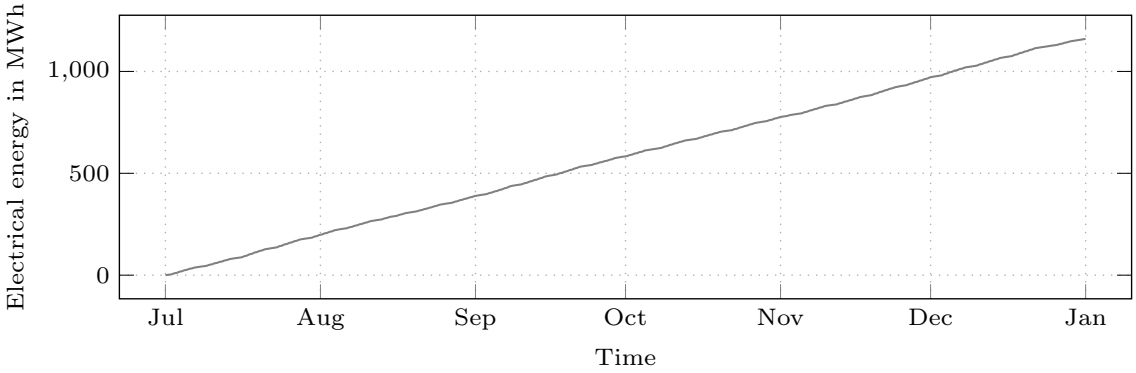
Definition 2.2 (Time Series Containing Energy or Power Measurements). A time series containing N_E energy measurements is defined by $E = e_1, e_2, \ldots, e_{N_E}$ and a time series containing N_P power measurements by $P = p_1, p_2, \ldots, p_{N_P}$.

1 Note that this commonly used term does not refer to energy as the specific physical quantity in which the observations are measured. Sometimes power measurements are also referred to as energy time series as they belong to the energy domain.

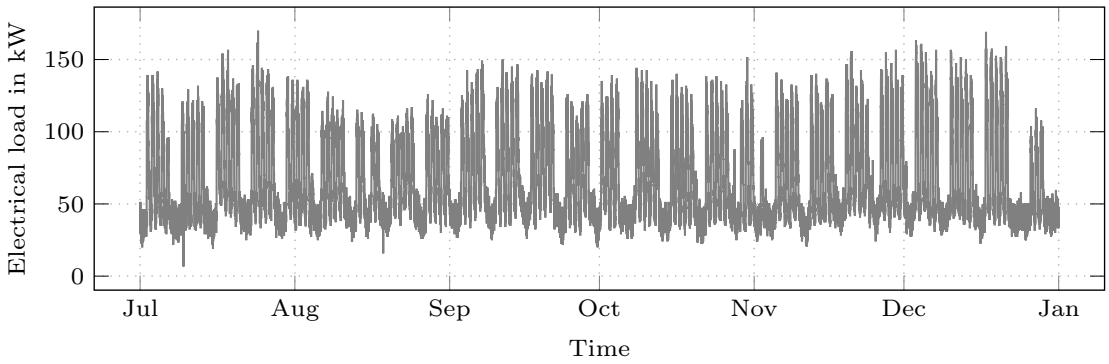
In the following, we use *energy time series* to refer to a time series observed in the energy system and *energy time series* E and *power time series* P to refer to time series containing measurements in a specific physical quantity.

Since power is the amount of energy converted at a unit of time, one can derive an energy time series E to obtain the corresponding power time series P (see Figure 2.1 for an example of a energy time series E and its corresponding power time series P). Each entry p_t of the power time series P represents the average power between the two time steps t and $t-1$. Therefore, each p_t is calculated as the difference between two consecutive entries e_{t-1} and e_t of the energy time series E divided by the time Δt between the two time steps t and $t-1$, i. e.,

$$p_t = \frac{e_t - e_{t-1}}{\Delta t}. \tag{2.1}$$



(a) An energy time series E containing energy measurements.



(b) A power time series P containing the corresponding derived power measurements.

Figure 2.1 An exemplary energy time series E and the corresponding power time series P. The shown time series is a 2012 section of *MT_013* from the "ElectricityLoadDiagrams20112014 Data Set" provided by the UCI Machine Learning Repository (Dua and Graff 2019).

Reversely, an energy time series E can also be obtained from a power time series P by integration. For each entry e_t of the energy time series E, all entries p_i of the power time series P up to this point in time t are integrated in a time-discrete manner, i. e.,

$$e_t = \Delta t \cdot \sum_{i=1}^{t} p_i + k, \tag{2.2}$$

where k is a constant representing the offset of the energy time series E. This offset is required to account for the potentially unknown history of the power time series P. Therefore, this calculation requires a power time series P without any missing values and with a known offset k to reproduce the corresponding original energy time series E.

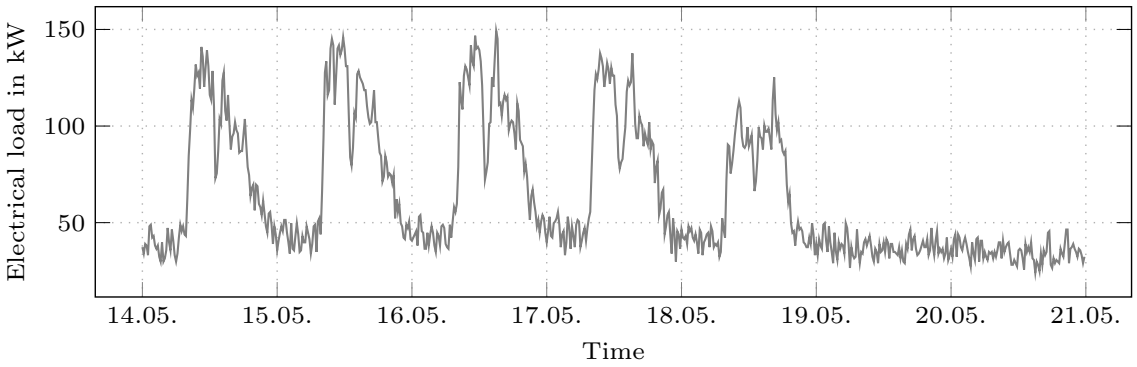
Power time series P from the energy system usually share three, potentially interdependent characteristics because they originate from the same complex physical system. The first characteristic is the multi-seasonality. A power time series P typically comprises, for example, daily, weekly, and seasonal patterns (see Figure 2.2). The second characteristic is the aggregation-level-dependent predictability. A power time series P is easier to forecast the more it is aggregated in, for example, time and space. The third characteristic is the exogenous influence. A power time series P strongly depends on, for example, meteorological influences such as temperature and economic influences such as large events (Dannecker 2015).

In the present thesis, we consider energy time series that principally exhibit these characteristics. However, we focus on energy time series representing electrical power or energy consumption with a low to medium aggregation level regarding time and space. The considered time series originate from a single client or building and thus only aggregate the consumption of a small entity. Moreover, the considered time series have a quarter-hourly temporal resolution, which is typical for the majority of data collected by smart meters (Wang et al. 2019).

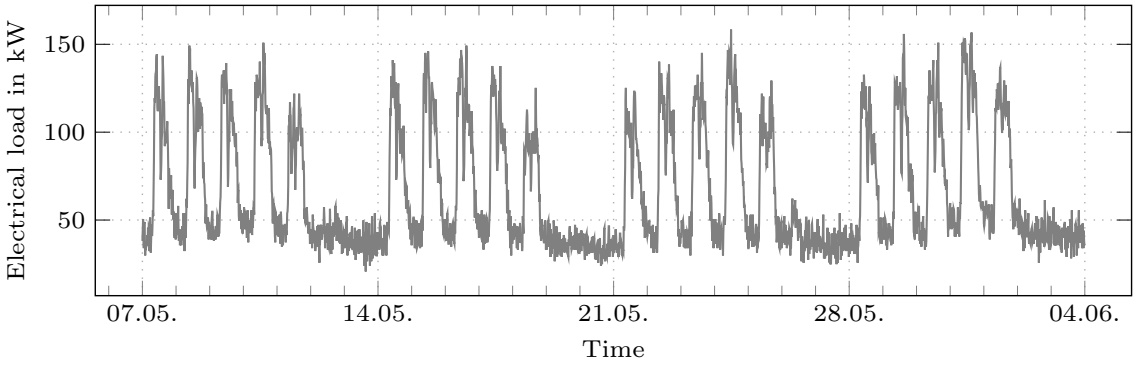
2.2 Anomalies in Energy Time Series

Since the energy system is a complex physical system, the energy time series measured in this system can contain atypical patterns. Depending on the goal and focus of the investigation of these patterns, different terms are used for them, including, for example, anomaly (Chandola et al. 2009), outlier (Aggarwal 2017; Barnett and Lewis 1978; Hawkins 1980; Rousseeuw and Leroy 1987), novelty (Dasgupta and Forrest 1996), surprise (Keogh et al. 2002; Shahabi et al. 2000), deviant (Jagadish et al. 1999), unusual pattern (Lonardi et al. 2006), change point (Yamanishi and Takeuchi 2002), concept drift (Heidrich et al. 2022), event (Guralnik and Srivastava 1999), discord (Yankov et al. 2008), abnormality (Duong et al. 2005), exception (Arning et al. 1996), fault (Katipamula and Brambley 2005a; Katipamula and Brambley 2005b), and noise (Aggarwal 2017). Due to the focus on the management of these patterns, the present thesis uses the term anomaly – and the reference to normality contained therein. In this thesis, we generally define anomalies as follows:

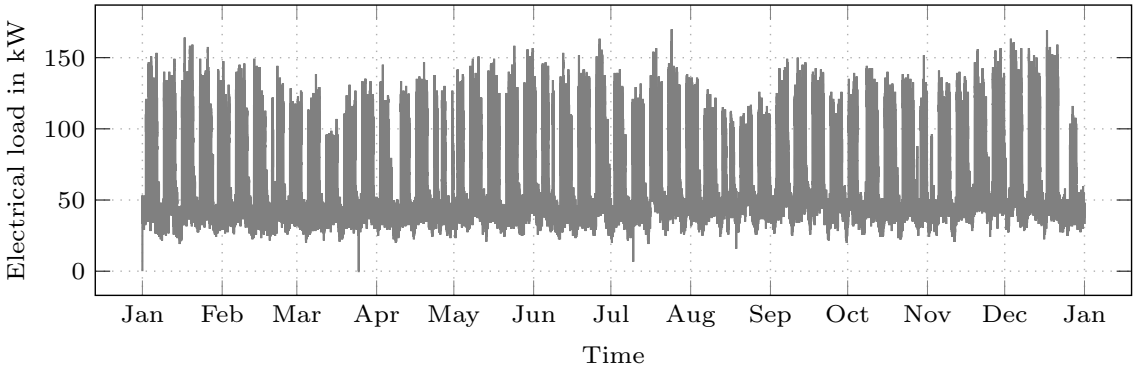
Definition 2.3 (Anomalies). "Anomalies are patterns in data that do not conform to a well defined notion of normal behavior." (Chandola et al. 2009, p. 15:2)



(a) Daily patterns.



(b) Weekly patterns.



(c) Seasonal patterns.

Figure 2.2 An exemplary power time series P that exhibits daily, weekly, and yearly patterns. To illustrate these patterns, one year, four weeks, and one week of the same time series from the year 2012 are shown. The shown time series is *MT_013* from the "ElectricityLoadDiagrams20112014 Data Set" provided by the UCI Machine Learning Repository (Dua and Graff 2019).

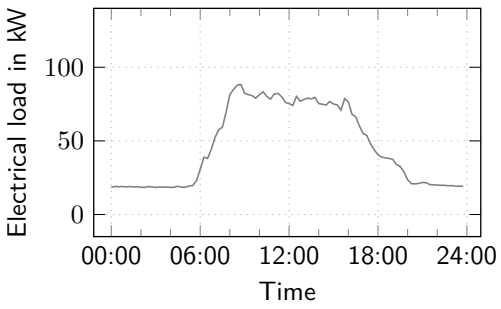
As this definition refers to normal behavior, it is related to a perception of normality that can be "a matter of subjective judgment" (Barnett and Lewis 1978, p. 4). In data mining, for example, this perception is associated with the assumption that at least one process exists that generates the data points representing normal behavior (Aggarwal 2017). From this perspective, anomalies are assumed to be generated by another mechanism

(Hawkins 1980) or by an unusual behavior of the actual generation process (Aggarwal 2017). In statistics, the perception of normality relates to the selection of a model that covers the normal behavior of the generation process such that data points deviating from the model are considered as anomalies (Rousseeuw and Leroy 1987).

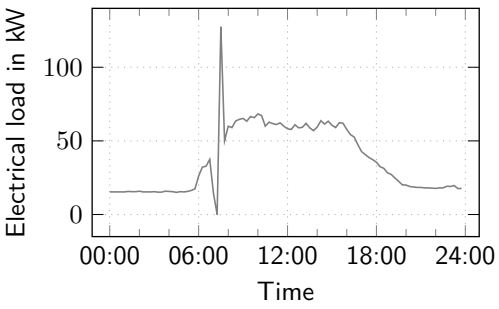
For the present thesis, one could naturally consider the above-mentioned three characteristics of energy time series – i. e., multi-seasonality, aggregation-level-dependent predictability, and exogenous influence – as the basic perception of normality. Since the present thesis uses energy time series with a low to medium aggregation level and considers only energy time series as data to develop data-driven methods for the management of anomalies, the multi-seasonality commonly observed in energy time series and as shown in Figure 2.2 serves as a basic perception of normality in the present thesis. More specifically, it focuses on the daily patterns as the smallest element of multi-seasonality because anomalies contained herein are very likely to manifest themselves in the larger weekly and seasonal patterns as well.

In addition to the underlying perception of normality, the definition of anomalies also depends on the considered data structure (Foorthuis 2021). With regard to time series data, anomalies can, for example, either refer to entire time series or parts of a time series (Gupta et al. 2014). In the present thesis, we focus on anomalies as parts of a single time series. In a single energy time series, such anomalies can occur due to a variety of causes, including technical faults (see e.g., Moghaddass and Wang 2018; Wang et al. 2020), theft (see e.g., Jokar et al. 2016; Nizar et al. 2008), and unusual consumption (see e.g., Seem 2007; Wang et al. 2020).

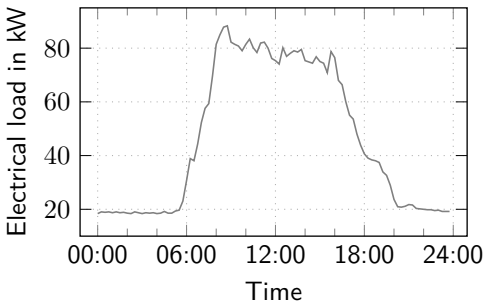
In the present thesis, we consider the two common anomaly groups of technical faults and unusual consumption (see Figure 2.3 for examples). While technical faults are related to the measuring using the metering infrastructure, unusual consumption can be associated with the unusual behavior of users or technical devices. Since these two groups contain distinct anomalies, they cover a variety of anomalies. While technical faults comprise anomalies whose values are not in the common range or are not part of typical patterns, unusual consumption consists of anomalies that represent usual patterns within the common value range but at an uncommon level (Wang et al. 2020).



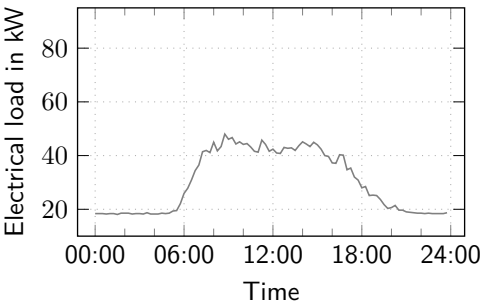
(a) Normal behavior.



(b) Technical fault.



(c) Normal behavior.



(d) Unusual consumption.

Figure 2.3 Examples of normal behavior and anomalies from the two considered groups of anomalies, namely technical faults and unusual consumption. The examples are specific days from a power time series P of an office building on the Campus North of the KIT.

3 Modeling Anomalies in Energy Time Series

As established in the introduction, anomaly management benefits from an anomaly detection that reliably identifies all relevant anomalies in an energy time series. Since reliably and effectively detecting anomalies is also important for various applications such as load analysis, load forecasting, and load management (Wang et al. 2019), anomaly detection in energy time series is generally a recent research topic (Himeur et al. 2021). While a portion of existing works develop detection methods with an exploratory approach to discover anomalies contained in time series (e.g., De Nadai and van Someren 2015; Fan et al. 2018; Li et al. 2010b), several works base their method development on knowledge about the anomalies to be detected (e.g., Himeur et al. 2020a; Jakkula and Cook 2010; Jokar et al. 2016; Pereira and Silveira 2018). However, gaining precise knowledge of the anomalies to be detected is associated with several challenges. By definition, anomalies are scarce and thus available data sets are imbalanced and hence there are comparatively few instances available in the time series that can be used for developing anomaly detection methods (Wen et al. 2021). Furthermore, this scarcity is particularly problematic for promising deep learning anomaly detection methods that require large training data sets to perform well (Pang et al. 2021). Additionally, a precise and comprehensive definition of relevant anomalies is missing (Himeur et al. 2021). Moreover, there is a lack of openly available energy time series with labeled anomalies or at least energy time series known to contain anomalies (Himeur et al. 2020b; Himeur et al. 2021).

In order to meet these challenges, different strategies can be applied for time series. Obviously, one can manually label anomalies in energy time series (Gulati and Arjunan 2022; Pereira and Silveira 2018; Ruff et al. 2021; Zhang et al. 2021). This strategy provides potentially very accurately labeled anomalies. However, it is limited to the anomalies contained in the time series, requires knowledge of the underlying system

---

Parts of this chapter are reproduced from

M. Turowski, M. Weber, O. Neumann, B. Heidrich, K. Phipps, H. K. Çakmak, R. Mikut, and V. Hagenmeyer (2022b). "Modeling and Generating Synthetic Anomalies for Energy and Power Time Series". In: *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22)*. ACM, pp. 471–484. DOI: 10.1145/3538637.3539760.

and typical patterns, often involves third parties such as facility managers or users, is time-consuming and costly, and potentially raises privacy concerns (Gaur et al. 2019).

Alternatively, one can apply means to define the majority of the time series as non-anomalous and the rest as anomalous, including selection (Ruff et al. 2021), rules (Himeur et al. 2020a), one-class classification (Zhao et al. 2013), statistical methods (Gaur et al. 2019), and pattern recognition methods (Zhang et al. 2021). This strategy depends less on experts. However, it can also be limited to the anomalies contained in the time series, it requires a strong notion of non-anomalous time series, anomalies may remain hidden in the time series, and a time-consuming and costly verification by an expert could still be necessary.

Another strategy is to increase the number of available time series through generation, augmentation, or sampling methods, either assuming the used time series to be anomaly-free or reproducing time series with labeled anomalies (Heidrich et al. 2023; Krawczyk 2016; Lu et al. 2021; Wen et al. 2021; Zhou et al. 2019). This strategy allows one to control the number of available time series containing or not containing anomalies, while also requiring a strong notion of non-anomalous time series or time series with labeled anomalies.

Lastly, as a special case of augmentation, one can insert synthetic anomalies into existing time series (Gaur et al. 2019; Ruff et al. 2021; Zhang et al. 2021). This strategy requires anomalies that well resemble real-world anomalies (Gaur et al. 2019; Ruff et al. 2021). Being able to control the number and location of specified anomalies provides a properly defined object of investigation for anomaly detection methods (Ruff et al. 2021) and turns an unsupervised into a supervised learning task (Steinbuss and Böhm 2021b). Since this strategy can be applied to various data sets from a domain, it can also help increase the use of available, currently underutilized unlabeled data sets to develop anomaly detection methods (Rossi and Chren 2020; Steinbuss and Böhm 2021a).

Similar to other domains like intrusion detection (Fan et al. 2001; Pham et al. 2014), security (Park et al. 2015), and performance monitoring (Wang et al. 2021), synthetic anomalies are used to develop anomaly detection methods for energy time series (e.g., Fahim et al. 2020; Jakkula and Cook 2010; Jokar et al. 2016; Luo et al. 2018c; Villar-Rodriguez et al. 2017). However, the inserted synthetic anomalies and their related parameters such as amplitude and quantity are generally not derived from real-world data and do not cover both energy and power, the typically recorded physical quantities. Furthermore, although Laptev (2018) considers the insertion of anomalies for time series in general, the implementations of methods to generate the considered synthetic anomalies are not openly available and thus cannot yet be directly applied.

Therefore, the present chapter proposes a method for generating four types of synthetic anomalies derived from real-world energy and power time series E and P for assuring the quality of to-be-developed anomaly detection methods. As a first step towards well-defined anomalies derived from real-world data, we identify anomalies in real-world energy and power time series E and P that are likely to be technical faults caused by the metering infrastructure and that may violate the underlying distribution corresponding to normal behavior. We then model the identified anomalies with parameters according to their

characteristics observed in the considered real-world time series. Given the modeled anomalies, we are able to insert them as synthetic anomalies into an arbitrary energy time series E or power time series P. We evaluate the identified and modeled anomalies in two ways. First, we examine whether inserted synthetic anomalies resemble the anomalies identified in real-world time series. Second, we show the benefit of inserted anomalies for training standard supervised anomaly detection methods as an initial analysis.

With the proposed method, we answer research question [RQ1] introduced in Section 1.1 that addresses how anomalies in energy time series can be modeled and generated to improve anomaly detection. By answering research question [RQ1], the proposed method provides the basis for the development and evaluation of the subsequent anomaly detection in the pipeline for managing anomalies (see Figure 3.1).

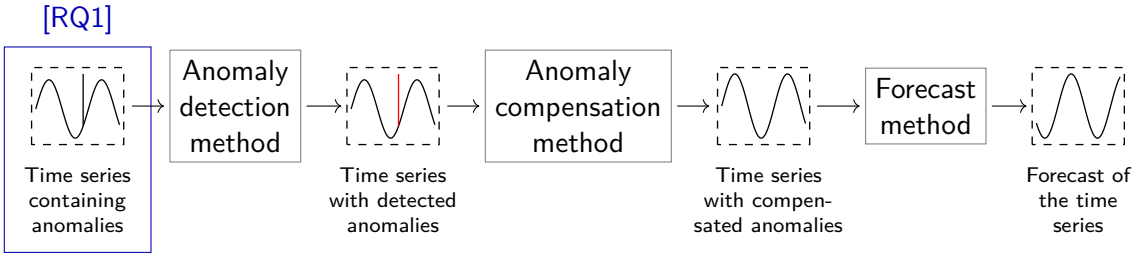


Figure 3.1 By answering research question [RQ1], the proposed method for generating synthetic anomalies derived from real-world anomalies that can be inserted into energy and power time series E and P provides the basis for the development and evaluation of the subsequent anomaly detection in the pipeline for managing anomalies.

The remainder of the present chapter is structured as follows. Section 3.1 introduces the anomalies identified in real-world time series, before Section 3.2 describes how the identified anomalies are modeled and generated to insert them as synthetic anomalies into arbitrary time series. In Section 3.3, we describe the experimental setting of the performed evaluation. In Section 3.4, we evaluate the generated synthetic anomalies. In Section 3.5, we discuss the results and our method, before Section 3.6 concludes the chapter.

3.1 Identifying Real-World Anomalies

In order to be able to generate realistic synthetic anomalies, we first derive anomalies from real-world energy and power time series E and P.

To this end, we consider electrical energy and power data collected on the Campus North of the Karlsruhe Institute of Technology (KIT), which is a subset of the data described in Wang et al. (2017) and whose publication is in preparation. This subset contains approximately 600 smart meter readings with a quarter-hourly resolution over a period of several years. Since these smart meters are installed in a variety of locations such as office buildings,

industrial facilities, gas motors, and photovoltaic panels, their recorded data presents typical patterns and anomalies of consumers and producers found in an ordinary city district. For each smart meter, an energy time series E and a power time series P are available.

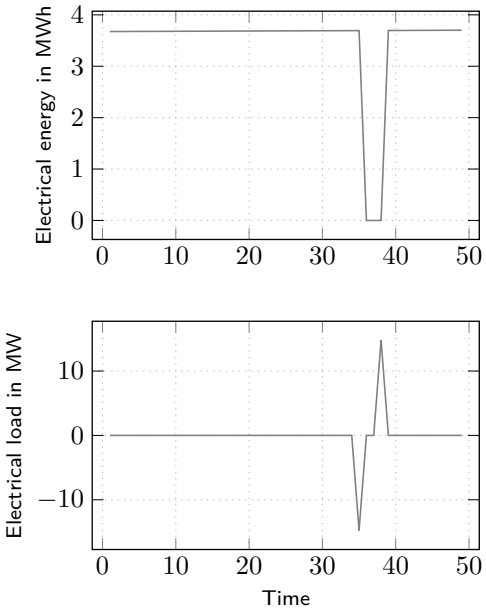
By carefully visually examining these time series, we are able to (i) find typical patterns for each smart meter and (ii) identify unusual patterns across all smart meters. In addition to both, we make use of knowledge about the facilities recorded by the smart meters to distinguish between anomalies and normal behavior in each time series. For example, we expect the power consumption of an office building not to drop to zero while an automatic lighting system may have no power consumption at all during the day. During this examination, we focus on anomalies that are likely to be technical faults caused by the metering infrastructure and that may violate the underlying distribution corresponding to normal behavior with very low or high values.

Across all smart meters, we find that many of the observed anomalies can be assigned to one of four anomaly types (see Figure 3.2) that also match general classes of anomalies described in the literature (see Table 3.1). For each identified anomaly type, we shortly describe its characteristics in an energy time series E and a power time series P and provide a potential explanation considering the metering infrastructure in the following.

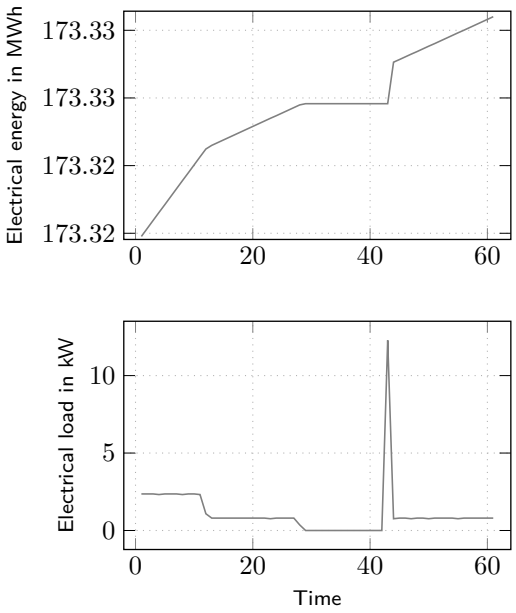
Anomaly Type 1 Anomalies of type 1 are characterized by a drop to zero for at least one time step in the energy time series E. After the zero values, the energy time series E jumps back to a plausible new value (see Figure 3.2a). The corresponding power time series P is characterized by a negative spike potentially followed by multiple zero power values and finally a positive spike. Anomalies of this type are likely to be caused by missing values in the measured energy time series E that are filled with zeros when saving the recorded values.

Anomaly Type 2 Anomalies of type 2 are characterized by a noticeable decrease in the gradient of the energy time series E. For one time step, there can be a decrease in the gradient that can be followed by constant energy values and that ends with a sharp increase in the gradient until a plausible new value is reached. Alternatively, there may be immediately constant energy values ending with an increased gradient (see Figure 3.2b). In the corresponding power time series P, there is a drop followed by a positive spike. If the energy time series E contains constant energy values, the power values drop to zero. In most of the observed cases, the height of the power spike is closely related to the length of the anomaly, suggesting that the power values of the constant sequence accumulate at one time step and thus form the spike. Anomalies of this type could be due to an interruption in the transmission of smart meter readings.

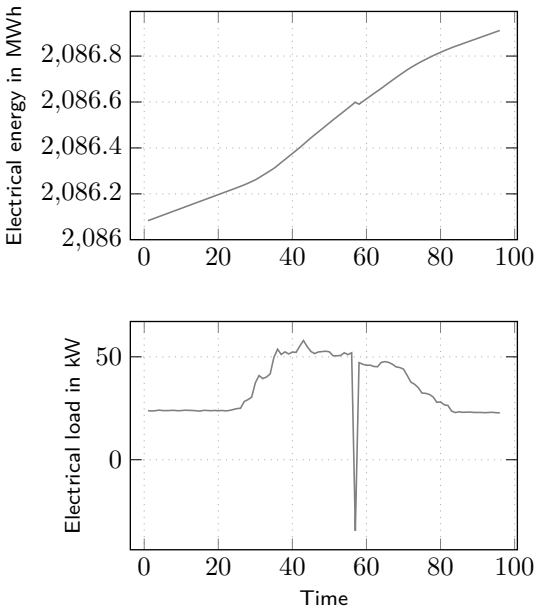
Anomaly Type 3 Anomalies of type 3 are characterized by a sudden dip in the energy time series E (see Figure 3.2c). In the corresponding power time series P, there is a negative power spike at one time step. Since this spike is rather small in some occurrences and rather strong in others, we observe two cases, i. e., a slight and an extreme negative



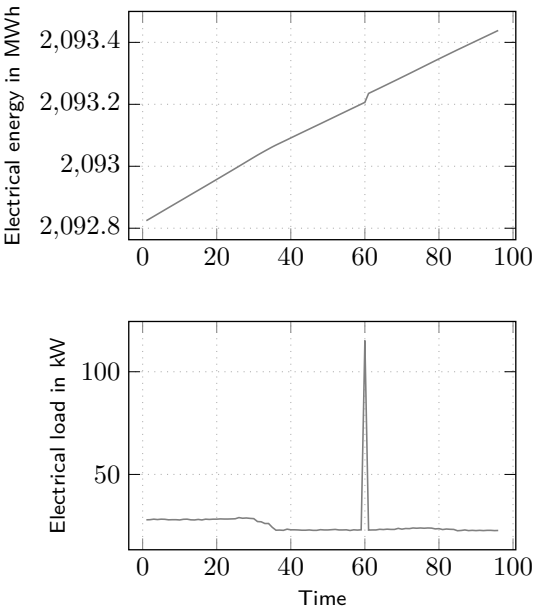
(a) Anomaly type 1: The energy time series E drops to zero for at least one time step and then jumps back to a plausible new value, corresponding to a negative spike potentially followed by zero values and finally a positive spike in the power time series P.



(b) Anomaly type 2: The gradient of the energy time series E decreases and can fully stagnate for several time steps, before returning to the correct value. This corresponds to a drop to potentially zero followed by a positive spike in the power time series P.



(c) Anomaly type 3: The energy time series E dips suddenly, which corresponds to a sudden negative spike in the power time series P.



(d) Anomaly type 4: The energy time series E contains a sudden increase in gradient, corresponding to a sudden positive spike in the power time series P.

Figure 3.2 Examples of the four anomaly types identified in the selected real-world data. Note that the power time series of type 1 is in the MW scale.

Table 3.1 Overview of the anomaly types identified in the power and energy time series E and P of the considered data and exemplary matching classes in the literature.

	Time series	Matching classes in literature
Anomaly type 1	Energy	"temporary change (ST-VIIb)" and "variation change (ST-VIIf)" (Foorthuis 2021, p. 313), "CONSTANT fault" (Sharma et al. 2010, p. 23:6)
	Power	"temporary change (ST-VIIb)" and "variation change (ST-VIIf)" (Foorthuis 2021, p. 313)
Anomaly type 2	Energy	"temporary change (ST-VIIb)" and "variation change (ST-VIIf)" (Foorthuis 2021, p. 313), "stuck-at fault" (Ni et al. 2009, p. 25:19), "stuck fault" (Zhang and Yan 2001, pp. 1390–1391)
	Power	"temporary change (ST-VIIb)" and "variation change (ST-VIIf)" (Foorthuis 2021, p. 313)
Anomaly type 3	Energy	"level shift (ST-VIIc)" (Foorthuis 2021, p. 313)
	Power	"local additive (ST-IVe)" (Foorthuis 2021, p. 309), "outlier fault" (Ni et al. 2009, pp. 25:16-25:17), "SHORT fault" (Sharma et al. 2010, p. 23:6), "spike fault" (Zhang and Yan 2001, pp. 1390–1391)
Anomaly type 4	Energy	"level shift (ST-VIIc)" (Foorthuis 2021, p. 313)
	Power	"local additive (ST-IVe)" (Foorthuis 2021, p. 309), "outlier fault" (Ni et al. 2009, pp. 25:16-25:17), "SHORT fault" (Sharma et al. 2010, p. 23:6), "spike fault" (Zhang and Yan 2001, pp. 1390–1391)

power spike. Anomalies of this type could occur due to an external adjustment of a smart meter such as a recalibration that aims to match the readings from multiple smart meters with a specific amount of energy. Anomalies with an extreme negative power spike could be caused by a reset of the respective smart meter.

Anomaly Type 4 Anomalies of type 4 are characterized by a sudden increase in the gradient of the energy time series E relative to the quarter-hourly resolution (see Figure 3.2d). In the corresponding power time series P, there is a positive power spike at one time step. Since this spike is also rather small in some occurrences and rather strong in others, we again observe two cases, i. e., a slight and an extreme positive power spike. Anomalies of this type can be caused by, for example, the change from daylight saving time to standard time. Because of this clock change by one hour, the consumption or generation within that hour is allocated to a single time step. This type of anomaly can also be observed in combination with anomalies of type 3, indicating an external adjustment of the smart meter.

We label anomalies of these four identified types in 50 one-year energy and one-year power time series E and P. Although it is theoretically possible to derive energy time series E to obtain the power time series P, we simultaneously label anomalies in both time series to eliminate possible sources of error and guarantee reliable labels. To obtain the 50 one-year time series from the selected data, we consider 2016 and 2017. We randomly select 23 smart meters from 2016 and 21 from 2017. Furthermore, we choose three smart meters that are present in both 2016

and 2017, resulting in six additional one-year time series. As shown in Tables A.1 and A.2 in Appendix A, the 50 related energy and power time series E and P of the selected smart meters are reasonably diverse, which is consistent with the fact that the used data set comprises smart meters at various locations.

3.2 Modeling Real-World Anomalies for Generating Synthetic Anomalies

To be able to generate the identified real-world anomalies as synthetic anomalies, we need to model them and to design a respective insertion method. Modeling anomalies of the different types requires several parameters. Before describing the specific modeling of each anomaly type, we briefly introduce the used parameters and how to set them.

For each previously identified anomaly type, we describe the necessary changes in the time series values — despite their proportional physical relationship — independently of each other for a given arbitrary time series $E = e_1, e_2, \ldots, e_N$ containing energy measurements and a given arbitrary time series $P = p_1, p_2, \ldots, p_N$ containing power measurements. With the described changes, we replicate an anomaly $\hat{e}_{j,i}$ or $\hat{p}_{j,i}$ of type j with start index i.

While anomalies of types 1 and 2 have a length l, anomalies of types 3 and 4 affect all entries after the time series entry i in an energy time series E and only the entry i itself in a power time series P. More precisely, for anomalies of types 1 and 2, we assume the length $l \sim \mathcal{U}_{[l_{min}, l_{max}]}$ to be from a uniform distribution in an interval $[l_{min}, l_{max}]$. For anomalies of types 1 and 3 for a power time series P, we additionally consider the fact that the amount of energy at a given time step in the power time series P in terms of the constant offset k is lost when deriving a power time series P from an energy time series E. For anomalies of these types, we thus explicitly consider the constant k, which has to be identical for all anomalies inserted into the same power time series P, to better represent the characteristics of a typical power time series P. Moreover, anomalies of types 3 and 4 comprise the random value r that determines the amplitude of their spike. We assume to be from a uniform distribution in an interval $[r_{min}, r_{max}]$, i. e., $r \sim \mathcal{U}_{[r_{min}, r_{max}]}$.

To generate anomalies of the modeled types, all these described parameters need to be set. For this, they can either be determined from available labeled data (as, for example, done in Section 3.3.2) or from values reported in literature (e.g., in Table 3.2).

Anomaly Type 1 We reproduce anomalies of type 1 with length l in an energy time series E by setting its entries $\hat{e}_i$ to $\hat{e}_{i+l-1}$ to zero. We model anomalies of this type as

$$\hat{e}_{1,i+n} = 0, \quad 0 \leqslant n < l, \tag{3.1}$$

where l is the length of the anomaly.

Table 3.2 Summary of the values determined from the 50 one-year power time series of the selected smart meters for the offset k, number, minimum length, maximum length, r_{min}, and r_{max} as presented in Table A.3. These values can be used as parameters to generate synthetic anomalies of the four modeled types for power time series. Note that anomalies of types 3 and 4 always have a length of one and that these types comprise two cases.

	Parameter	Case	Value
	k		$[177, 431796]$
Anomaly type 1	#		$[5, 18]$
	Min		3
	Max		$[3, 4465]$
Anomaly type 2	#		$[0, 15]$
	Min		$[2, 1731]$
	Max		$[2, 7434]$
Anomaly type 3	#		$[0, 2]$
	r_{min}	Slight	0.61
		Extreme	-
	r_{max}	Slight	1.62
		Extreme	-
Anomaly type 4	#		[0,4]
	r_{min}	Slight	1.15
		Extreme	11.01
	r_{max}	Slight	8.1
		Extreme	13

In order to insert anomalies of type 1 with length l into a power time series P, we set the first anomalous entry $\hat{p}_i$ to the negative value of the power aggregated up to this time step i. The next $l-2$ entries are set to zero and the last entry of the anomaly $\hat{p}_{i+l-1}$ to the sum of the power aggregated up to time step $i+l-1$ corresponding to the jump in the energy time series E. Formally, we describe this as

$$\hat{p}_{1,i+n} = \begin{cases} -1 \cdot (\sum_{t=1}^{i-1} p_t) - k, & n = 0 \\ 0, & 0 < n < l-1 \\ (\sum_{t=1}^{i+l-1} p_t) + k, & n = l-1, \end{cases} \tag{3.2}$$

where $l \geqslant 2$ is the anomaly's length and k is the constant offset.

Anomaly Type 2 To replicate anomalies of type 2 with length l in an energy time series E, we determine the first anomalous value $\hat{e}_i$ as the average of the observed value at index i, e_i, and the previous value e_{i-1} weighted by the random number $r \sim \mathcal{U}_{[0,1)}$. All following $l-1$ anomalous entries are then set to this first anomalous value $\hat{e}_i$. Anomalies of this type can be described by

$$\hat{e}_{2,i+n} = r \cdot e_i + (1-r) \cdot e_{i-1}, \quad 0 \leqslant n < l, \tag{3.3}$$

where l is the length of the anomaly and $r \sim \mathcal{U}_{[0,1)}$ can be assumed from a uniform distribution and the same for all entries. Note that, in the special case of $r = 0$, the anomaly directly starts with the value of the previous time step.

To insert anomalies of type 2 with length l into a power time series P, we scale down the first anomalous entry $\hat{p}_i$ using a random number $r \sim \mathcal{U}_{[0,1)}$ and set the subsequent $l-2$ entries to zero. In order to form the observed peak at the last entry of the anomaly, we set the last entry $\hat{p}_{i+l-1}$ to the sum of the original values of the previously manipulated entries and subtract the first manipulated value $\hat{p}_i$. Formally, the anomaly can be described as

$$\hat{p}_{2,i+n} = \begin{cases} r \cdot p_i, & n = 0 \\ 0, & 0 < n < l-1 \\ (1-r) \cdot p_i + (\sum_{t=i+1}^{i+l-1} p_t), & n = l-1, \end{cases} \tag{3.4}$$

where $l \geqslant 2$ is the length of the anomaly and $r \sim \mathcal{U}_{[0,1)}$. Analogously to the energy time series E, in the special case of $r = 0$, the manipulated entries directly start with a zero.

Anomaly Type 3 We reproduce anomalies of type 3 in an energy time series E by subtracting a certain amount of energy from every time series entry with an index greater than or equal to i. The amount of energy to be subtracted depends on the observed case, i. e., the slight and the extreme negative spike. For the slight negative power spike, we insert anomalies of type 3 by subtracting a value based on the energy difference between the anomalous entry e_i and its predecessor e_{i-1} multiplied by a random value r. The anomaly can formally be described by

$$\hat{e}_{3,i+n} = e_{i+n} - r \cdot |e_i - e_{i-1}|, \quad n \geqslant 0, \tag{3.5}$$

where r is the random value defined above.

For the extreme negative power spike that is likely caused by a smart meter reset to zero, we subtract the value of the entry e_i from all subsequent time series entries, i. e.,

$$\hat{e}_{3,i+n} = e_{i+n} - e_i, \quad n \geqslant 0. \tag{3.6}$$

In order to insert anomalies of type 3 into a power time series P, we generate anomalies for the slight negative peak, by setting the anomalous entry $\hat{p}_i$ to the previous value p_{i-1} multiplied by a random value r defined above. Formally, we describe this as

$$\hat{p}_{3,i} = -r \cdot p_{i-1}, \tag{3.7}$$

where r is the random value defined above.

For the extreme negative spike corresponding to a drop of the energy time series E to zero, we use the same model as for the starting value of anomalies from type 1 in Equation (3.2), i. e.,

$$\hat{p}_{3,i} = -1 \cdot (\sum_{t=1}^{i-1} p_t) - k, \tag{3.8}$$

where k is the previously defined constant offset.

Anomaly Type 4 To replicate anomalies of type 4 in an energy time series E, we apply a similar manipulation as for anomalies of type 3. To cover both the observed slight and extreme cases, we use two different sampling intervals for r. The anomaly is thus defined as

$$\hat{e}_{4,i+n} = e_{i+n} + r \cdot |e_i - e_{i-1}|, \quad n \geqslant 0, \tag{3.9}$$

where r is the random value defined above sampled twice to represent the slight and the extreme case.

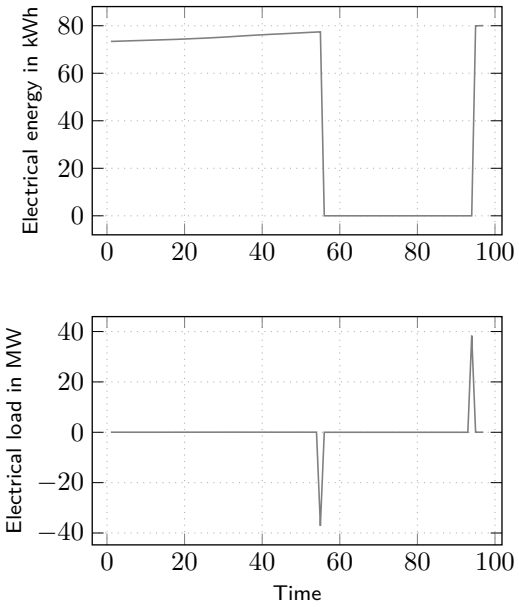
To insert anomalies of type 4 into a power time series P, we model the observed positive spike $\hat{p}_i$ by multiplying its predecessor p_{i-1} with a random value r defined above sampled from two different intervals. Again, to cover both the observed slight case and the observed extreme case, we use two different sampling intervals for r. Formally, we define this anomaly as

$$\hat{p}_{4,i} = r \cdot p_{i-1}, \tag{3.10}$$

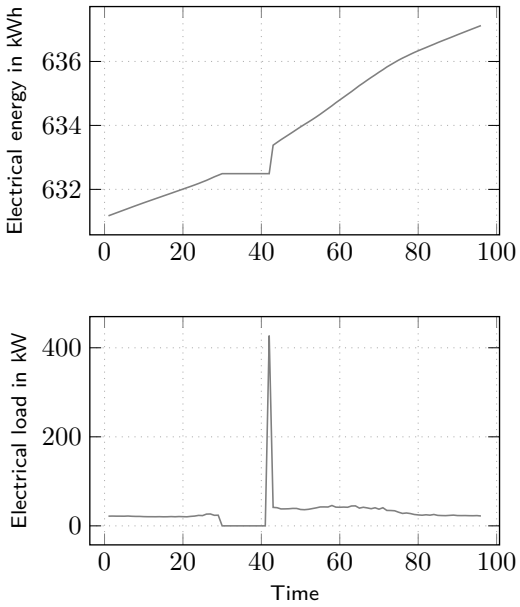
where r is the random value defined above sampled twice to represent the slight and the extreme case.

When generating anomalies of these four types, we need to consider the potential interaction between the modeled anomaly types. With regard to an energy time series E, one first has to insert anomalies of types 3 and 4 before inserting anomalies of types 1 and 2, because anomalies of types 3 and 4 affect all values after their occurrence and thus potentially influence anomalies of the other types. Concerning a power time series P, one can, however, insert anomalies in the ascending order of the type. To avoid overlapping anomalies, we use a sequential approach. For each anomaly to be generated, we firstly search for an anomaly-free sequence $(x_i, ..., x_{i+l})$ in the time series X before we insert the anomaly. Figure 3.3 shows a synthetic anomaly of each anomaly type generated with this implementation.

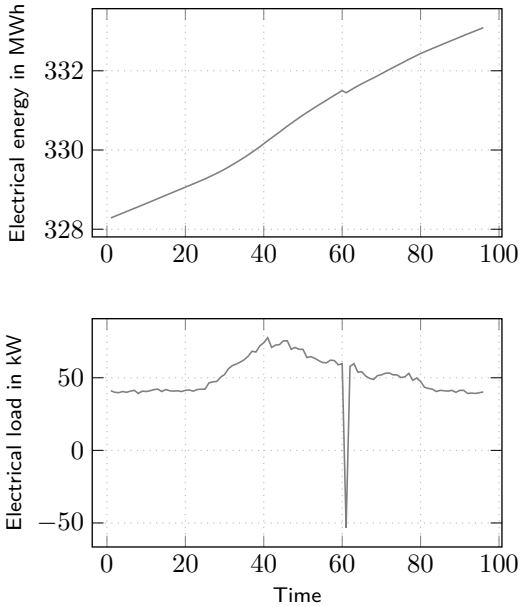
To be able to reproducibly generate anomalies of the four types for an energy time series E or power time series P, we implement a publicly available pipeline in Python



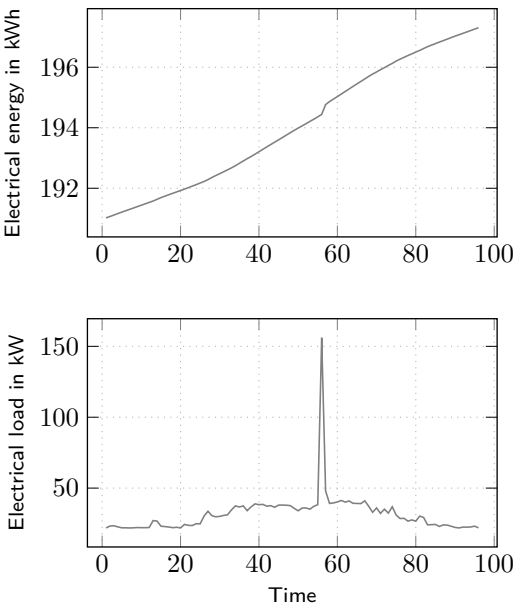
(a) Synthetic anomalies of type 1: The characteristic drop to zero in the energy time series E and the first negative, then positive spikes in the power time series P are clearly visible.



(b) Synthetic anomalies of type 2: We observe the characteristic stagnation in the gradient and following spring in the energy time series E, as well as the drop and subsequent positive spike in the power time series P.



(c) Synthetic anomalies of type 3 (extreme case): The characteristic decrease in the gradient of the energy time series E and corresponding negative spike in the power time series P are clearly observable.



(d) Synthetic anomalies of type 4 (extreme case): We observe the characteristic sudden increase in the gradient of the energy time series e and corresponding positive spike in the power time series P.

Figure 3.3 Examples of generated synthetic anomalies of the four modeled anomaly types. Note that the power time series of type 1 is in the MW scale.

using pyWATTS[1] (Heidrich et al. 2021).[2] It allows to control the types, the quantities, and the parameters of the synthetic anomalies that are inserted into an arbitrary energy time series E or power time series P.

3.3 Experimental Setting

In this section, we present how we evaluate the modeled anomalies. After describing the selected data and the calculation of the parameters used for the evaluated generation method, we introduce the applied evaluation methods, the selected metrics, and the used hard- and software.

3.3.1 Used Data

For the evaluation, we use the previously introduced electrical energy and power data collected on the KIT Campus North. More specifically, we again consider the previously labeled 50 time series for the evaluation because of the available labels for the related energy time series E_i and power time series P_i, where i is the number of the considered one-year time series. Since an energy time series E_i is typically monotonically rising and thus non-stationary, one would usually apply differencing to make it stationary and thus useful for time series analyses (Hyndman and Athanasopoulos 2021). As each already available power time series P_i is exactly the result of such a differencing due to the proportional physical relationship between energy and power, we focus on them in the following.

To obtain an anomaly-free power time series P_i for the following analyses, we first use the corresponding manually labeled 50 one-year energy time series E_i. More precisely, we mark the labeled anomalies in these time series as missing values and apply the Copy-Paste Imputation (CPI) method (Weber et al. 2021). The CPI method has shown a strong performance in imputing missing values with realistic patterns while preserving the amount of energy associated with the missing values. After imputing the anomalies marked as missing values in each energy time series E_i, we calculate their derivative to obtain the corresponding anomaly-free power time series P_i. We use the resulting anomaly-free power time series P_i as the basis for inserting the generated synthetic anomalies used in the evaluation.

For the application of the selected evaluation methods, we finally create overlapping samples with a size of 96 from all considered power time series, namely the power time series P_i containing identified anomalies, the power time series $\hat{P}_i$ reproducing the identified anomalies with synthetic anomalies, and the power time series $\breve{P}_i$ containing more synthetic anomalies than anomalies identified in the original data.

1 https://github.com/KIT-IAI/pyWATTS
2 https://github.com/KIT-IAI/GeneratingSyntheticEnergyPowerAnomalies

3.3.2 Used Anomaly Generation Parameters

We determine the parameters required for the generation of the desired synthetic anomalies in each power time series $\hat{P}_i$ from the labeled power time series P_i. From this time series, we can directly determine the number of anomalies of all four types as well as the minimum and maximum length l_{min} and l_{max} of type 1 and 2 anomalies. For k, we use the first value in the corresponding available energy times E_i for the comparison between synthetic and identified anomalies or set it to zero when evaluating the benefit of synthetic anomalies for the training of detection methods.[3]

Lastly, we determine r_{min} and r_{max} for the slight and the extreme case for anomaly type 3 and 4 using DBSCAN (Ester et al. 1996). For both anomaly types, we calculate the random value r for all labeled anomalies of this type in the power time series P_i with

$$r = \frac{p_i}{\overline{p}}, \tag{3.11}$$

where p_i is the considered anomaly and $\overline{p}$ is the corresponding local average.[4] For the local average $\overline{p}$, we arbitrarily choose a sufficiently small range of 10, i. e.,

$$\overline{p} = \frac{\sum_{t=i-5}^{i+5} p_t - p_i}{10}, \tag{3.12}$$

where $p_t \neq p_i,\ \forall t \neq i$. We cluster the result into two classes. For both types, we assume that the class with the majority of the considered anomalies represents the slight power spike case and the other the extreme spike case. For anomaly type 3, we thus select the smallest and the largest value in the majority class as r_{min} and r_{max}. For anomaly type 4, we select the smallest and the largest value from each class as r_{min} and r_{max} for the corresponding case.

Using this calculation, we aim to reproduce the anomalies contained in the original power time series P_i with the parameters reported in Table A.3 in Appendix A to examine whether the synthetic anomalies in the power time series $\hat{P}_i$ resemble the anomalies identified in the real-world power time series P_i. To evaluate the benefit of synthetic anomalies for training supervised anomaly detection methods, we additionally increase the number of anomalies compared to the original power time series P_i and by doubling the number of anomalies to obtain the power time series $\breve{P}_i$ that contains more synthetic anomalies than anomalies identified in the original data (see Table A.4 in Appendix A). Note that, for both evaluations, we define valid intervals for the minimum and maximum lengths of anomaly types 1 and 2 to consider the imbalanced distributions of these lengths. We limit the minimum length to the interval $[3, 92]$ and the maximum length to the interval $[3, 96]$

3 If an energy time series E_i is not available, one could sum the power over a year of data and multiply it by the presumed number of years the smart meter has been in service.

4 Given the energy time series E_i, one could analogously calculate $r = \frac{e_{i-1} - e_i}{e_{i-1} - e_{i-2}}$ for anomalies of type 3 and $r = \frac{e_i - e_{i-1}}{e_{i-1} - e_{i-2}}$ for anomalies of type 4.

for type 1 and the minimum length to the interval $[2, 44]$ and the maximum length to $[2, 48]$ for type 2. Additionally, we insert only anomalies of the extreme case of anomaly types 3 and 4 as soon as one exists in the corresponding labeled power time series.

3.3.3 Applied Methods

In the evaluation, we apply four different methods, which we describe in the following.

To examine whether the synthetic anomalies resemble the identified anomalies, we apply a statistical visualization and a discriminator method. As visualization method, we use the t-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton 2008). The t-SNE visualizes high-dimensional data in a two-dimensional map such that similar data points are likely to appear close together and dissimilar data points far apart. As discriminator method, we implement a simple three-layered fully-connected Neural Network (NN) with ten neurons in the hidden layer and ReLU as activation function. In this NN, all neurons are interconnected across the layers and the neuron in the output layer indicates with a binary encoding whether the input data is an identified real anomaly (0) or a synthetic anomaly (1). For training the NN, we use the binary cross-entropy as loss and RMSprop (Hinton et al. 2012) as optimizer.

To evaluate the benefit of synthetic anomalies for their training, we apply two supervised anomaly detection methods. More precisely, we select a k-Nearest Neighbor (kNN) classifier and a decision tree classifier. The kNN classifier uses a proximity measure to classify a test sample based on the similarity of training instances (Cover and Hart 1967). In comparison, as a non-parametric method, the decision tree learns simple decision rules inferred from data features (Breiman et al. 1984).

3.3.4 Metrics

The evaluation is based on the two following metrics.

For the discriminator method, we use the discriminative score. It is defined as

$$\text{Discriminative Score} = |\,\text{Accuracy} - 0.5|, \tag{3.13}$$

where Accuracy is the result from the applied discriminator method.

For the supervised anomaly detection methods, we apply the commonly used F1-Score. It is the harmonic mean between precision and recall and is defined as

$$\text{F1-Score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2} \cdot (\text{FP} + \text{FN})}, \tag{3.14}$$

where TP are the true positives, FP the false positives, and FN the false negatives in the considered classification.

3.3.5 Hardware and Software

Throughout the evaluation, we apply a standard computer with a four-core i7 CPU and 16 GB of RAM. Moreover, all applied methods are implemented in Python. The t-SNE, the decision tree, and the kNN are implemented with scikit-learn[5] (Pedregosa et al. 2011) and the fully-connected NN with Keras[6] (Chollet et al. 2015). The evaluation is automated with pyWATTS[7] (Heidrich et al. 2021) using these implementations.

3.4 Results

To evaluate the modeled anomalies, we perform a twofold evaluation. First, we examine whether generated synthetic anomalies resemble the anomalies identified in real-world data. Second, we evaluate the benefit of synthetic anomalies for training standard supervised anomaly detection methods.

3.4.1 Comparing Identified and Synthetic Anomalies

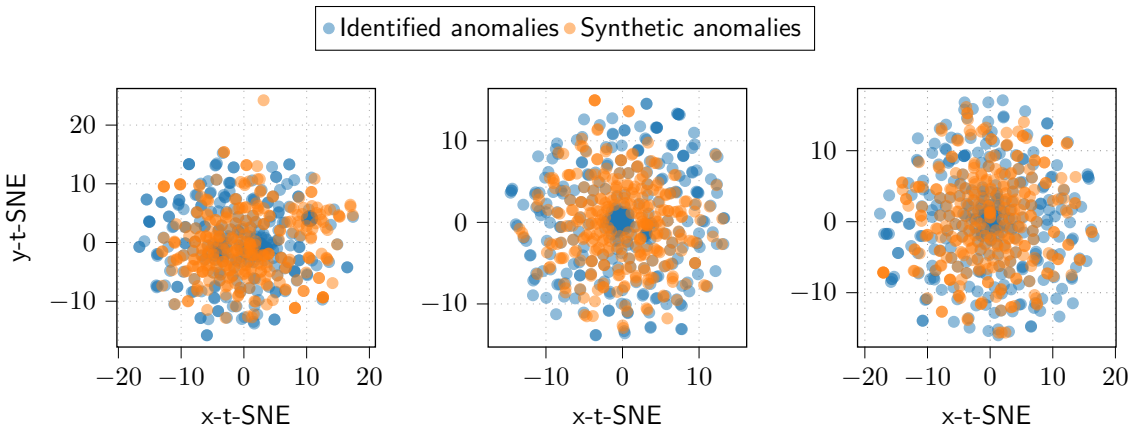
For the synthetic anomalies to be useful, they must resemble the identified real-world anomalies and ideally be indistinguishable from them. In this section, we first qualitatively compare identified and synthetic anomalies with the help of t-SNE visualizations, before quantitatively comparing them with the discriminator method.

Firstly, we examine the t-SNE visualizations of an identical number of samples containing identified and synthetic anomalies from three exemplary one-year time series. For this, we use the power time series P_i with identified anomalies and the power time series $\hat{P}_i$ with synthetic anomalies reproducing the identified anomalies.

As shown in Figure 3.4, we observe that, for all three time series, the samples with identified anomalies and the samples with inserted synthetic anomalies overlap in most cases. The overlap indicates that the synthetic anomalies exhibit properties similar to the identified anomalies.

Secondly, we consider the discriminative score of the discriminator method in detecting the difference between samples of identified anomalies and samples of inserted synthetic anomalies from all 50 considered one-year time series. For this, we again use the power time series P_i with identified anomalies and the power time series $\hat{P}_i$ with synthetic anomalies reproducing the identified anomalies. We combine their samples containing anomalies into a single data set, before we use the first 66 % for training the discriminator method and remaining 34 % for testing.

5 https://scikit-learn.org/
6 https://keras.io/
7 https://github.com/KIT-IAI/pyWATTS



(a) Time series 4. **(b)** Time series 32. **(c)** Time series 33.

Figure 3.4 A t-SNE visualization of an identical number of samples containing identified anomalies and synthetic anomalies from three exemplary one-year time series.

A histogram of the resulting discriminative score is shown in Figure 3.5. The discriminative score, that is rounded to one decimal digit and whose maximum is 0.5, is plotted on the x-axis to provide bins for the histogram and the number of occurrences in each bin is shown on the y-axis. We observe that the discriminative score is 0.25 or smaller for a large majority of the samples and higher for only few samples. This result indicates that the discriminator is mostly unable to differentiate between identified and synthetic anomalies.

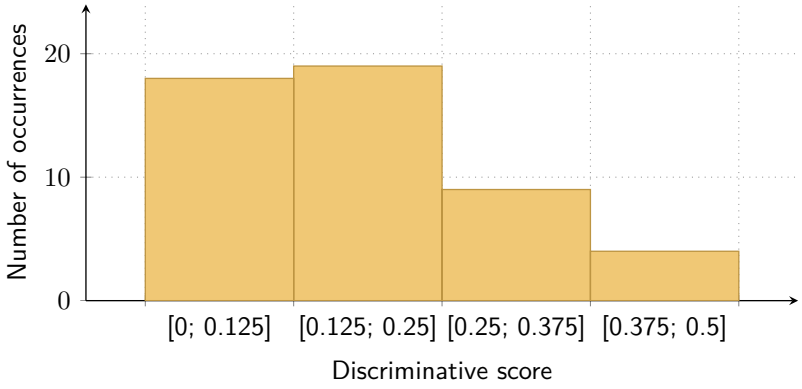


Figure 3.5 A histogram of the discriminative score of all samples containing identified or synthetic anomalies from the 50 considered one-year time series. The x-axis shows the histogram bins for the discriminative score in steps of 0.125, whereas the number of occurrences in each bin is plotted on the y-axis.

3.4.2 Benefit of Synthetic Anomalies for Anomaly Detection

To evaluate whether synthetic anomalies exhibiting real-world characteristics are beneficial, we exemplarily analyze this benefit for training standard supervised anomaly detection methods. As an initial analysis of this, we compare the detection performance of two

different training strategies based on the F1-Score. The first strategy is Train Real Test Real (TRTR), where we use the original power time series P_i with identified anomalies for both training and testing. The second strategy is Train Synthetic Test Real (TSTR)[8], where we train the anomaly detection method on the power time series $\breve{P}_i$ containing more synthetic anomalies than anomalies identified in the original data and test its performance on the original power time series P_i with identified anomalies.

Figure 3.6 shows a comparison of the detection performance of these two strategies for the kNN and the decision tree. This comparison comprises the number of wins for each strategy, whereby a strategy is considered to win whenever its detection performance is better than that of the other strategy. If both strategies provide an identical detection performance, they are considered to be equal. Independent of the considered detection method, power time series $\breve{P}_i$ containing more synthetic anomalies than anomalies identified in the original data win far more often than power time series P_i only containing identified anomalies. For the kNN, power time series $\breve{P}_i$ containing more synthetic anomalies than anomalies identified in the original data win 21 times. Power time series P_i with identified anomalies only win eight times; in 21 cases, both strategies perform equally. For the decision tree, power time series $\breve{P}_i$ containing more synthetic anomalies than anomalies identified in the original data win 25 times, whereas power time series P_i with identified anomalies only win 20 times. In 5 cases, the strategies perform equally.

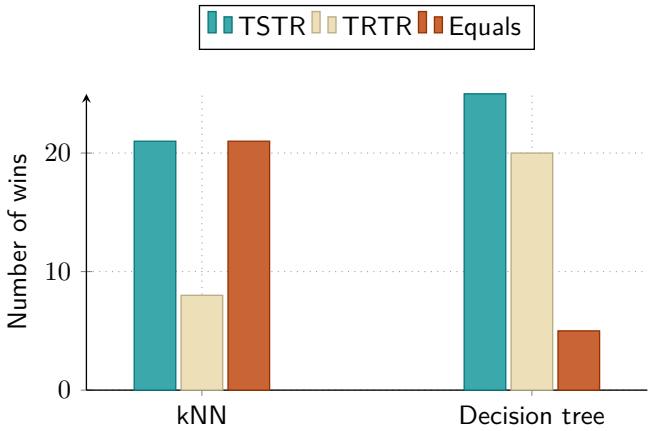


Figure 3.6 A comparison of the detection performance of the two training strategies Train Real Test Real (TRTR) and Train Synthetic Test Real (TSTR) based on the F1-Score for the two selected supervised detection methods. A strategy *wins*, if the resulting detection performs better than that of the other strategy and is considered *equal* if it performs equally.

8 Note that synthetic in this context refers to the synthetic anomalies and not the underlying data.

3.5 Discussion

This section discusses the results, the modeled anomalies, and the proposed method for modeling and generating synthetic anomalies.

In the results, the t-SNE visualizations of identified and synthetic anomalies illustrate that the generated synthetic anomalies mostly overlap with the identified anomalies. Similarly, the histogram of the discriminative scores shows that identified and synthetic anomalies are difficult to distinguish with the discriminator method. Both results confirm that our synthetic anomalies accurately replicate the identified anomalies. From this observation, we conclude that the proposed method is capable of generating synthetic anomalies with real-world properties. Furthermore, since the TSTR strategy performs better or as well as TRTR in most cases, considering these synthetic anomalies in the training of a standard supervised anomaly detection method is beneficial for its detection performance. Given this observation, the proposed anomaly generation method can be used to improve anomaly detection methods in the future.

Despite these promising initial results, we note that our experiments are limited to the considered data, the associated production and consumption, and the anomalies identified in this data. Specifically for the extreme cases of anomaly types 3 and 4, the number of occurrences in our data set are small. Therefore, the parameters selected for the synthetic anomalies are based on a small sample size and we expect that more accurate results could be achieved with more data. Moreover, we model the identified anomaly types and develop the generation method with great care and on the basis of the available data. However, despite our evaluation, we cannot completely rule out the possibility that generated synthetic anomalies are easier to detect in other real-world data based on, for example, the transitions to the real data. Furthermore, the identified anomaly types are likely to be the result of technical failures in the metering infrastructure that cause unusual values such as extreme positive or negative spikes or a series of zeros. These types of anomalies have clearly defined, often extreme characteristics and are therefore relatively easy to detect. We expect that anomalies characterized by typical patterns at uncommon levels – such as unusual consumption – are more difficult to detect and, therefore, synthetic anomalies that reflect these characteristics could further improve to-be-developed anomaly detection methods.

We also note that our method currently inserts synthetic anomalies for energy time series E and power time series P separately. Since most applications only consider either energy time series E or power time series P, we believe this limitation to be not critical. However, due to the physical relationship between energy and power, simultaneously inserting multivariate synthetic anomalies for both energy time series E and power time series P could be beneficial in some cases.

3.6 Contribution and Future Work

The present chapter investigates how anomalies in energy time series can be modeled and generated to improve anomaly detection, thus answering research question [RQ1]. For this, we firstly analyze real-world energy time series E and power time series P to identify four commonly occurring anomaly types. Given these identified anomaly types, we formally model each type and prepare a generation method to insert a chosen number of synthetic anomalies of each type into an arbitrary energy time series E or a power time series P.

With this approach, the present chapter provides the following contributions:

- We introduce a method for generating four types of synthetic anomalies derived from real-world anomalies that can be inserted into arbitrary energy time series E or power time series P.
- We demonstrate that the introduced method is capable of generating realistic synthetic anomalies.
- We also show that the inserted artificial anomalies are beneficial for training supervised anomaly detection methods.

Based on the introduced generation method, there are several follow-up questions for future work. For example, future work could consider further time series, especially those that contain anomalies characterized by unusual consumption. Furthermore, to model the physical relationship between energy and power, future work should consider the simultaneous multivariate generation of synthetic anomalies for energy time series E and power time series P. Lastly, future work could include fuzziness into the generation to increase the variation of the generated synthetic anomalies.

4 Detecting Anomalies in Energy Time Series

As described in the introduction, anomaly management requires anomaly detection methods that perform well in identifying the relevant anomalies contained in an energy time series. Due to its importance for various applications such as load analysis, load forecasting, and load management, detecting anomalies in recorded energy time series is in general an important recent issue in energy systems (Himeur et al. 2021). To detect anomalies of various types, a large variety of methods, often categorized as supervised or unsupervised methods, are employed (Himeur et al. 2021; Schmidl et al. 2022). These methods are typically applied directly or after scaling to the data containing anomalies.

However, for other tasks, machine learning methods recently demonstrated promising performance when directly applied to the so-called latent space representation of the data. This latent space is an abstract multi-dimensional space containing a meaningful representation of features that is often not directly interpretable. Such latent space data representations have been successfully applied in forecasting (Kim and Cho 2021; Nguyen and Quanz 2021), offline reinforcement learning (Rafailov et al. 2021), photo upsampling (Menon et al. 2020), path planning (Hung et al. 2022), and trajectory adjustment (Kutsuzawa et al. 2019). Furthermore, with regards to anomaly detection, there is evidence from a medical application that the latent space better separates the representation of anomalies and non-anomalous data (Pereira and Silveira 2019).

In order to separate anomalous and non-anomalous data in energy time series, a latent space that follows a known and traceable latent space distribution could be particularly useful. If this distribution has clearly defined mathematical properties, as is the case with the Gaussian distribution, these properties will help to define how anomalies are

---

Parts of this chapter are reproduced from

M. Turowski, B. Heidrich, K. Phipps, K. Schmieder, O. Neumann, R. Mikut, and V. Hagenmeyer (2022a). "Enhancing Anomaly Detection Methods for Energy Time Series Using Latent Space Data Representations". In: *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22)*. ACM, pp. 208–227. DOI: 10.1145/3538637.3538851.

represented. Given these considerations, we can then use the representation of data in the latent space to enhance anomaly detection.

The present chapter, therefore, proposes a novel approach to generally enhance anomaly detection methods for energy time series by taking advantage of their latent space representation. For this approach, we first train a generative method to learn a mapping from the original data to the latent space. Given the learned mapping, the generative model is used to create the latent space representation of an input time series containing anomalies. The resulting latent space data representation serves then as an input for an arbitrary existing supervised or unsupervised anomaly detection method.

To evaluate the proposed approach, we firstly qualitatively examine its benefit by visualizing how latent space data representations and common data representations separate anomalies and non-anomalous data. Secondly, we quantitatively evaluate how the proposed approach improves the detection performance. For this purpose, we apply a selection of existing supervised and unsupervised detection methods to real-world load data, where we insert synthetic anomalies of two groups. Anomalies of the first group represent technical faults derived from real-world data that violate the underlying distribution corresponding to normal behavior and can be easily recognized by a human. Anomalies of the second group comprise unusual consumption that remains in the underlying distribution and are hard to recognize. We additionally apply the considered supervised and unsupervised detection methods to real-world data with labeled technical faults.

With the proposed approach, we answer research question [RQ2] described in Section 1.1 that addresses how anomaly detection methods for energy time series can be enhanced. By answering research question [RQ2], the proposed approach detects anomalies with a high accuracy and thus provides a solid basis for compensating the detected anomalies within the subsequent anomaly compensation in the pipeline for managing anomalies (see Figure 4.1).

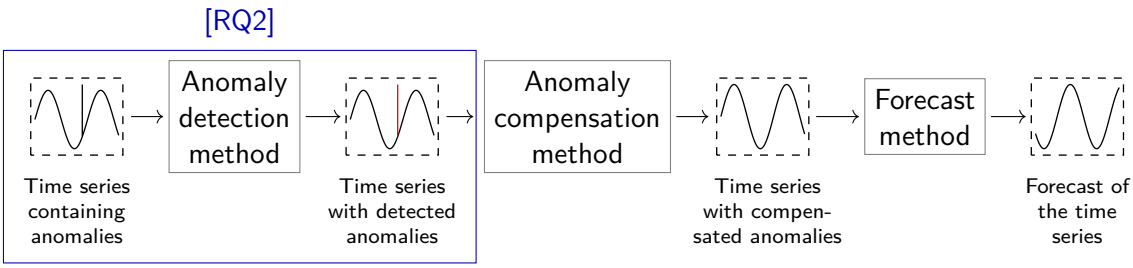


Figure 4.1 By answering research question [RQ2], the proposed approach for enhancing anomaly detection methods for energy time series by taking advantage of their latent space representation detects anomalies with a high accuracy and thus provides a solid basis for compensating the detected anomalies within the subsequent anomaly compensation in the pipeline for managing anomalies.

The remainder of the present chapter is organized as follows. Section 4.1 introduces the proposed approach to enhance anomaly detection method by directly using the

latent space data representation created with a generative model. In Section 4.2, we describe the experimental setting of the performed evaluation. In Section 4.3, we then report the evaluation results. Finally, we discuss the results and proposed approach in Section 4.4 and conclude the chapter in Section 4.5.

4.1 Anomaly Detection Using Latent Space Data Representations

This section explains how latent space data representations can be used to enhance anomaly detection methods.[1] First, we describe how latent space data representations of time series can be created with a generative method and how this method is trained in both supervised and unsupervised anomaly detection settings. We then present how the trained generative method is applied to detect anomalies contained in a time series using an arbitrary anomaly detection method.

4.1.1 Create Latent Space Data Representations With a Generative Method

To create a latent space representation of a time series $\mathbf{z} \in \mathbb{Z}$, we need to realize a mapping $f : \mathbb{X} \to \mathbb{Z}$ from the original realization space $\mathbb{X}$ to the latent space $\mathbb{Z}$. To ensure this mapping represents a known and tractable latent space distribution P_Z in the latent space, it can be realized with a Variational Autoencoder (VAE) (Kingma and Welling 2014) or an Invertible Neural Network (INN) (Kingma and Dhariwal 2018). Both methods can be extended with a conditioning mechanism to a conditional Variational Autoencoder (cVAE) (Sohn et al. 2015) or a conditional Invertible Neural Network (cINN) (Ardizzone et al. 2019), allowing them to process conditional inputs. With calendar and statistical information as conditional inputs, these methods can consider typical properties of energy time series, i. e., daily, weekly, and seasonal patterns. While both methods realize the mapping f, they differ in their structure. CVAEs consist of a jointly trained encoder and decoder with the encoder realizing the mapping f and the decoder reconstructing the original representation of the time series, i. e., the mapping $g : \mathbb{Z} \to \mathbb{X}$. In contrast, cINNs only comprise a single bijective mapping $f^{-1} = g$ that realizes both the encoding and decoding.[2]

1 A Python implementation of the proposed approach is publicly available at https://github.com/KIT-IAI/EnhancingAnomalyDetectionMethods.

2 Note that a standard Generative Adversarial Network (GAN) (Goodfellow et al. 2014) comprising a generator and a discriminator cannot be used. The generator realizes the mapping g but the discriminator only distinguishes real from synthetic data and thus does not realize the mapping f.

However, since we are only interested in the latent space representation and not the reconstructed representation, we only focus on learning the mapping f. To this means, we use a cVAE or cINN to create this latent space representation using the mapping

$$f : \mathbb{X} \to \mathbb{Z},\ \mathbf{x} \mapsto f(\mathbf{x}; \mathbf{d}, \mathbf{s}, \theta) = \mathbf{z}, \tag{4.1}$$

where $\mathbf{x} \in \mathbb{X}$ is the time series with arbitrary but fixed length L, $\mathbf{d}$ is calendar information of length L, $\mathbf{s}$ is statistical information of arbitrary but fixed length, and θ is the set of all trainable parameters.

4.1.2 Training of the Generative Method

As shown in Figure 4.2, the training of the selected generative method is based on samples of an input time series as well as calendar and statistical information. The training process itself differs for supervised and unsupervised anomaly detection, which we detail in the following.

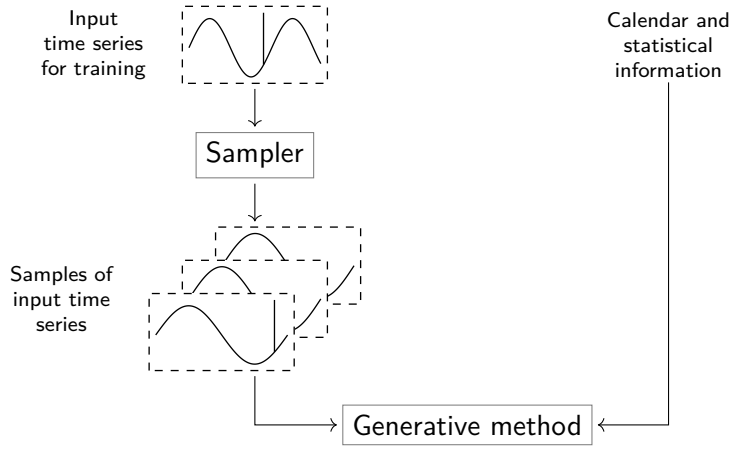


Figure 4.2 The proposed approach uses samples of an input time series as well as calendar and statistical information to train the generative method.

Supervised Anomaly Detection For supervised anomaly detection, we take advantage of the labeled anomalies and train the selected generative method with fixed-length samples of an anomaly-free time series, where we assume that all contained anomalies are labeled. For each fixed-length sample, we calculate the loss $\mathcal{L}_i$, which varies depending on the generative method selected. For a cINN, we use a maximum likelihood optimization based on the *change of variable formula*. This results in the maximum likelihood loss for a sample $\mathbf{x}_i$ defined as

$$\mathcal{L}_i = \frac{\| f(\mathbf{x}_i; \mathbf{d}_i, \mathbf{s}_i, \theta) \|_2^2}{2} - \log | J_i |, \tag{4.2}$$

where $J_i = \det(\partial f / \partial \mathbf{x}|_{\mathbf{x}_i})$ is the determinant of the Jacobian evaluated for the i-th sample (Ardizzone et al. 2019). For a cVAE, we use a reconstruction loss with regularization, resulting in the loss for a sample $\mathbf{x}_i$

$$\mathcal{L}_i = \mathbb{E}\left[(\hat{\mathbf{x}}_i - \mathbf{x}_i)^2\right] + \mathbb{KL}\left(\mathbf{x_i}, P_Z\right), \tag{4.3}$$

where $\hat{\mathbf{x}}_i$ is the reconstructed time series sample from the cVAE, and $\mathbb{KL}$ is the Kullback-Leibler divergence (Kullback and Leibler 1951). This loss function $\mathcal{L}_i$ ensures that the generative methods learn a standard normal distribution of a non-anomalous time series as latent space distribution P_Z. Therefore, when we apply the generative method to a time series containing anomalies, the contained anomalies are likely to be mapped to the outer regions of the latent space distribution and thus are more easy to be detected.

Unsupervised Anomaly Detection For unsupervised anomaly detection, the training process of the selected generative method has to cope with non-existent anomaly labels for the data points. This is realized on the fair assumption that the minority of the used training data is anomalous and that the training errors are higher for anomalous data points than for non-anomalous data points.

We take advantage of these expected higher errors for anomalies by defining a contamination parameter c for the training process. The contamination c represents the assumed share of anomalous data points in the considered time series and is used to calculate the threshold quantile $\mathcal{Q}_c$ for the training errors of each sample of a batch in the training process. Each sample with a training error above this threshold quantile $\mathcal{Q}_c$ is excluded from the loss function $\mathcal{L}_i$. The resulting adapted loss for a sample x_i is

$$\mathcal{L}_i' = \begin{cases} \mathcal{L}_i, & \mathcal{L}_i < \mathcal{Q}_c \\ 0, & \text{else,} \end{cases} \tag{4.4}$$

where $\mathcal{L}_i$ is the loss from Equation (4.2) or Equation (4.3), depending on the generative method used. Using this loss ensures that the selected generative model is also capable of learning an anomaly-free latent space data representation with the latent space distribution P_Z in an unsupervised manner.

4.1.3 Detecting Anomalies in Time Series Using the Latent Space Data Representation

Given the trained generative method, anomalies contained in a time series are detected as shown in Figure 4.3. Firstly, from an input time series containing anomalies, a sampler draws samples which serve as input for the trained generative method. As additional inputs, the trained generative method uses calendar and statistical information associated with this time series. Secondly, given these inputs, the trained generative method creates a latent space representation of the input time series' samples. Thirdly,

an arbitrary anomaly detection method is directly applied to the created latent space data representation to detect the samples that contain anomalies.

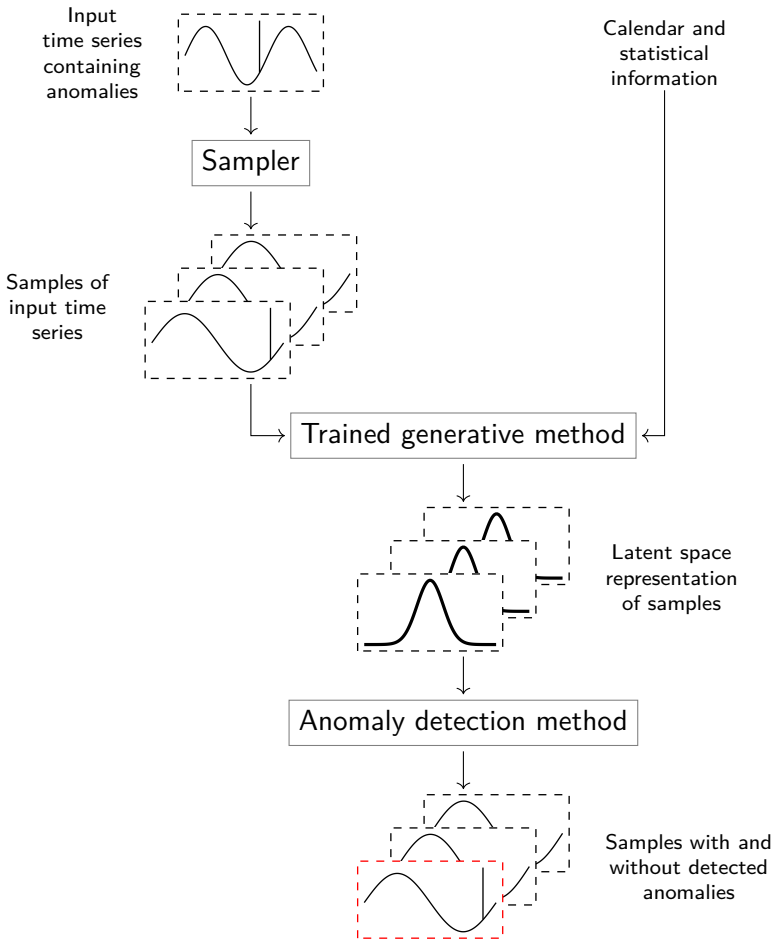


Figure 4.3 In the proposed approach, the previously trained generative method uses samples of an input time series containing anomalies as well as calendar and statistical information as inputs. Based on these inputs, the trained generative method provides the latent space representation of the samples. Using this latent space data representation, an arbitrary anomaly detection method detects the samples of the considered time series that contain anomalies.

This approach differs for supervised and unsupervised anomaly detection methods. For a supervised anomaly detection method, two steps are involved: Firstly, it is trained on the created latent space data representation using a training set with labeled anomalies. Secondly, it classifies the samples from a test set. An unsupervised anomaly detection methods, however, is applied to a complete data set without labeled anomalies.

4.2 Experimental Setting

In this section, we present how we evaluate the proposed approach. After describing the data set and the inserted synthetic anomalies, we introduce the used generative methods, the compared data representations, and the applied anomaly detection methods. Finally, we describe the evaluation criteria and the used hard- and software.

4.2.1 Data Sets With Anomalies

For the evaluation, we use two data sets. Both contain real-world data but differ in the observed anomalies. While we insert synthetic anomalies into the first, the second already contains labeled anomalies.

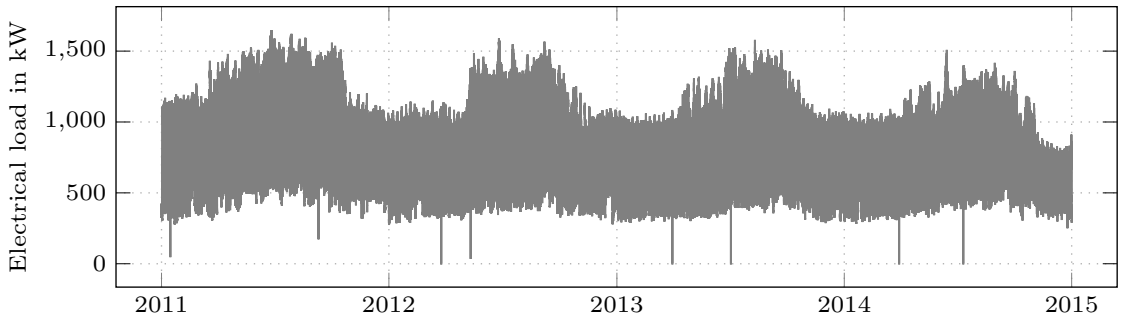
Data With Synthetic Anomalies As the first data set, we choose the publicly available "ElectricityLoadDiagrams20112014 Data Set"[3] from the UCI Machine Learning Repository (Dua and Graff 2019). This data set has a quarter-hourly temporal resolution and contains electrical power time series of 370 clients with different consumption behaviors (Rodrigues and Trindade 2018), which are mostly available for the period from the beginning of 2011 until the end of 2014. To cover the complete period of four years, to consider the electrical load of a typical client, and to use comparatively anomaly-free time series, we select the time series containing power measurements `MT_200` for the evaluation (see Figure 4.4a).

Since the selected time series does not contain labeled anomalies, we insert synthetic anomalies into it (see e. g., Figure 4.4b). For the insertion, we use anomalies of two common groups (see e.g., Wang et al. 2020), namely technical faults in the metering infrastructure and unusual consumption. In the following, we briefly introduce the anomaly types of each group and describe the relevant parameters and the corresponding manipulation for an anomaly $\hat{p}_{j,i}$ of type j with start index i in a given power time series $P = p_1, p_2, ...p_N$ with length N.

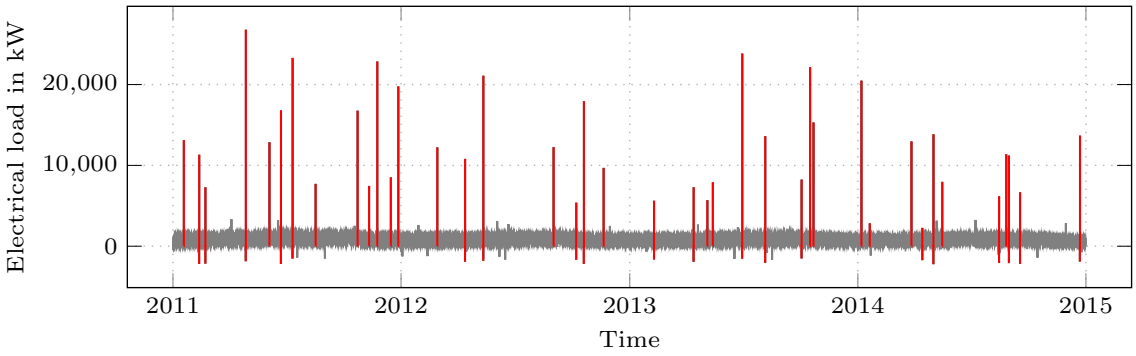
As synthetic anomalies of the first group, we select the previously introduced four types of anomalies that are identified in real-world power time series in Turowski et al. (2022b) and that violate the underlying distribution corresponding to normal behavior (see Figure 4.5). While the values of anomaly types 1 and 3 are not in the valid range, anomaly types 2 and 4 comprise values from the valid range that are not part of typical patterns in power time series.

- Anomaly type 1 refers to a negative power spike followed by zero power values and a positive spike (see Figure 4.5a). This characteristic can be based on a power time

3 https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014



(a) Without inserted synthetic anomalies.



(b) With 20 inserted synthetic anomalies of each of the four types from the group of technical faults. The anomalies are plotted in red.

Figure 4.4 Overview of the selected data from the first data set without inserted synthetic anomalies and with inserted synthetic anomalies of all four types from the group of technical faults.

series whose values are derived from an energy time series containing missing values due to a communication error. For the insertion, we model anomalies of this type as

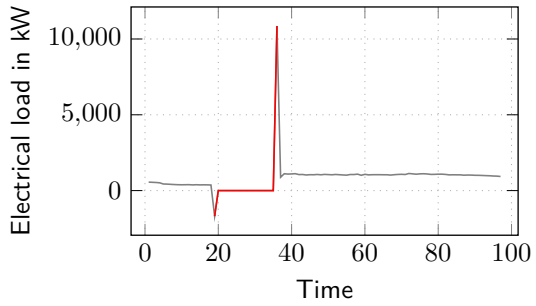
$$\hat{p}_{1,i+n} = \begin{cases} -1 \cdot mean(P) + r_s \cdot std(P), & n = 0 \\ 0, & 0 < n < l-1 \\ \sum_{t=1}^{i+l-1} p_t, & n = l-1, \end{cases} \tag{4.5}$$

where the length $l \sim \mathcal{U}_{[5,24]}$ and the random scaling factor $r_s = 2 + r \cdot 3$ with $r \sim \mathcal{U}_{[0,1]}$.

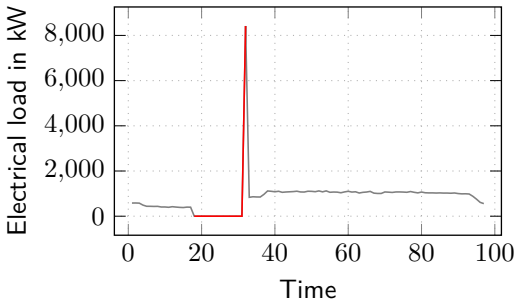
- Anomaly type 2 comprises several zero power values followed by a positive spike (see Figure 4.5b). This characteristic can be a result of an interruption in the transmission of power values from smart meters. For the insertion, we model anomalies of this type as

$$\hat{p}_{2,i+n} = \begin{cases} 0, & 0 \leqslant n < l-1 \\ \sum_{t=i}^{i+l-1} p_t, & n = l-1, \end{cases} \tag{4.6}$$

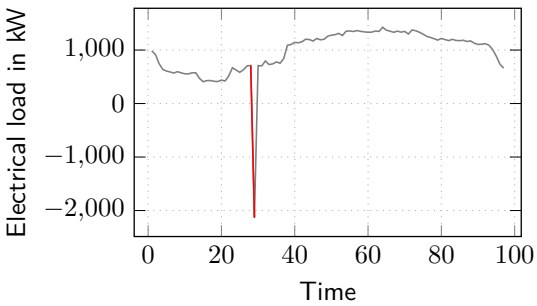
where the length $l \sim \mathcal{U}_{[5,24]}$.



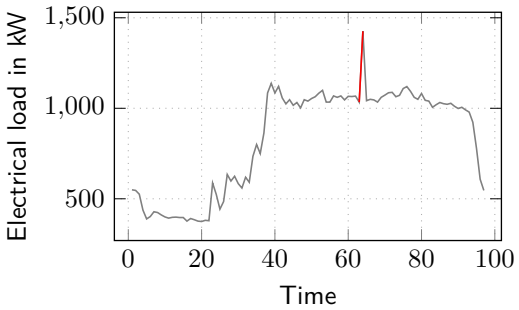
(a) Anomaly type 1: negative power spike followed by zero values and positive spike.



(b) Anomaly type 2: zero power values followed by a positive spike.



(c) Anomaly type 3: negative power spike.



(d) Anomaly type 4: positive power spike.

Figure 4.5 Examples of the anomaly types 1 to 4 from the technical faults that we insert as synthetic anomalies into the selected first data set. The anomalies are plotted in red. Note that the anomalies of types 3 and 4 actually have a length of one but are marked together with their previous value to be recognizable.

- Anomaly type 3 is a negative power spike (see Figure 4.5c). It could be caused by external recalibration of a smart meter reading so that, together with the readings of other smart meters, the meter reading matches a certain amount of load. For the insertion, we model anomalies of this type as

$$\hat{p}_{3,i} = -r_s \cdot mean(P), \tag{4.7}$$

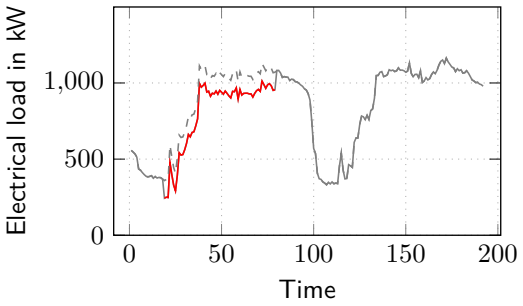
  where the random scaling factor $r_s = 0.01 + r \cdot 3.99$ with $r \sim \mathcal{U}_{[0,1]}$.

- Anomaly type 4 is a positive power spike (see Figure 4.5d). It may be due to, for example, the change from daylight saving time to standard time, where power values of five time steps are recorded as the value of one time step. For the insertion, we model anomalies of this type as

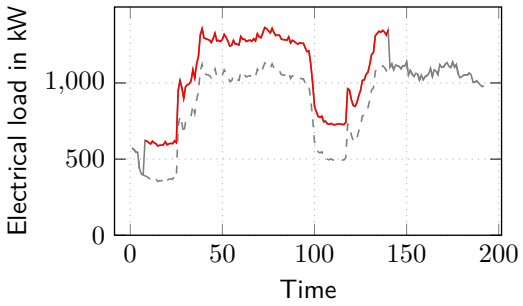
$$\hat{p}_{4,i} = r \cdot mean(P), \tag{4.8}$$

  where the random scaling factor $r_s = 3 + r \cdot 5$ with $r \sim \mathcal{U}_{[0,1]}$.

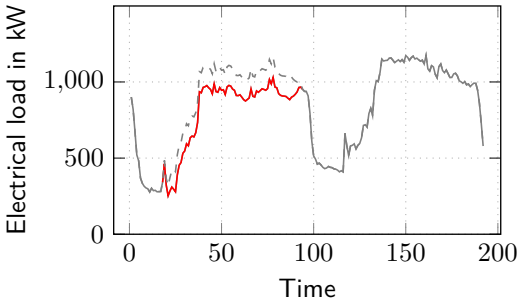
As synthetic anomalies of the second group, we insert four types of anomalies representing unusual behavior: Anomaly types 5 and 7 represent unusually low power consumption, while anomaly types 6 and 8 illustrate unusually high power consumption (see Figure 4.6). These four anomaly types comprise values from the valid range and represent in themselves typical patterns.



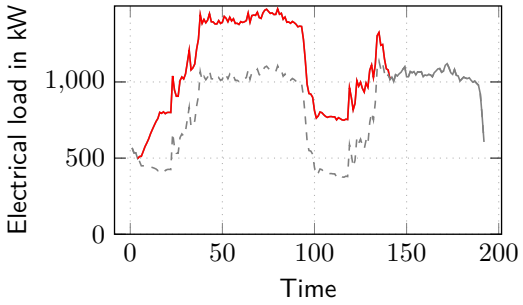
(a) Anomaly type 5: abrupt small temporary reduction in the power values.



(b) Anomaly type 6: abrupt small temporary increase in the power values.



(c) Anomaly type 7: small temporary reduction in the power values with a gradual start and end.



(d) Anomaly type 8: small temporary increase in the power values with a gradual start and end.

Figure 4.6 Examples of the anomaly types 5 to 8 from the unusual consumption that we insert as synthetic anomalies into the selected first data set. The anomalies are plotted in red.

- Anomaly type 5 is an abrupt small temporary reduction in the power values (see Figure 4.6a). This characteristic can be caused by a large device temporarily shutting down, resulting directly in lower consumption. For the insertion, we model anomalies of this type as

$$\hat{p}_{5,i+n} = p_i - r \cdot p_{\text{min}}, \quad 0 < n < l - 1, \tag{4.9}$$

  where the length $l \sim \mathcal{U}_{[48,144]}$, the random scaling factor $r \sim \mathcal{U}_{[0.3,0.8]}$, and the minimum power value $p_{\text{min}} = \min\{p_i, p_{i+1}, \ldots, p_{i+l-1}\}$.

- Anomaly type 6 is an abrupt small temporary increase in the power values (see Figure 4.6b). This characteristic can be the result of switching on a rarely used large device for a short period of time. For the insertion, we model anomalies of this type as

$$\hat{p}_{6,i+n} = p_i + r \cdot p_{\text{min}}, \quad 0 < n < l - 1, \tag{4.10}$$

where the length $l \sim \mathcal{U}_{[48,144]}$, the random scaling factor $r \sim \mathcal{U}_{[0.5,1]}$, and the minimum power value $p_{\text{min}} = \min\{p_i, p_{i+1}, \ldots, p_{i+l-1}\}$.

- Anomaly type 7 is also a period of temporary reduction in the power values, however with a gradual start and end (see Figure 4.6c). It could be caused by a large device in an unusual operating mode gradually requiring less power for a period of time, before slowly returning to its usual performance. For the insertion, we model anomalies of this type as

$$\hat{p}_{7,i} = \begin{cases} p_i - r \cdot p_{\text{min}} \cdot \frac{l}{10} \cdot i, & 0 < n < \frac{l}{10} \\ p_i - r \cdot p_{\text{min}}, & \frac{l}{10} \leqslant n \leqslant 1 - \frac{l}{10} \\ p_i - r \cdot p_{\text{min}} \cdot \frac{l}{10} \cdot (1 - i), & 1 - \frac{l}{10} < n < l - 1, \end{cases} \tag{4.11}$$

  where the length $l \sim \mathcal{U}_{[48,144]}$, the random scaling factor $r \sim \mathcal{U}_{[0.3,0.8]}$, and the minimum power value $p_{\text{min}} = \min\{p_i, p_{i+1}, \ldots, p_{i+l-1}\}$.

- Anomaly type 8 is an again small temporary increase in the power values, however with a gradual start and end (see Figure 4.6d). Similar to anomaly type 7, it may be due to a device in an unusual operating mode that gradually requires more power, before slowly returning to its usual performance. For the insertion, we model anomalies of this type as

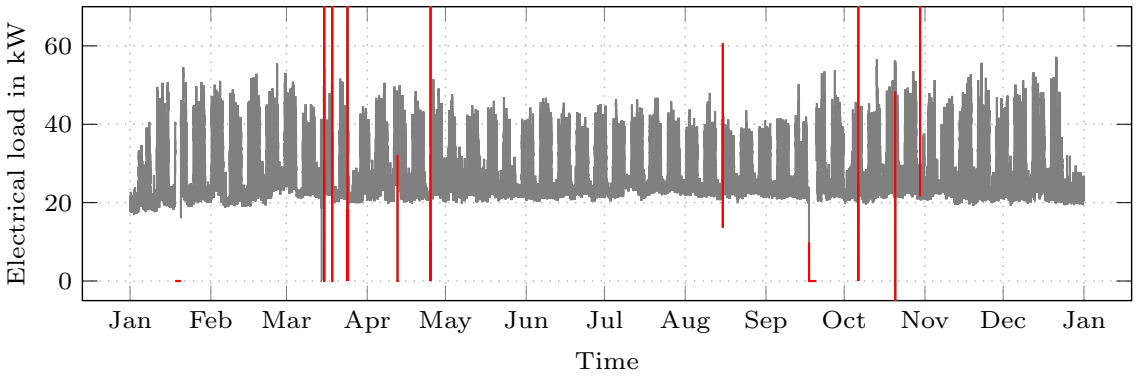
$$\hat{p}_{8,i} = \begin{cases} p_i + r \cdot p_{\text{min}} \cdot \frac{l}{10} \cdot i, & 0 < n < \frac{l}{10} \\ p_i + r \cdot p_{\text{min}}, & \frac{l}{10} \leqslant n \leqslant 1 - \frac{l}{10} \\ p_i + r \cdot p_{\text{min}} \cdot \frac{l}{10} \cdot (1 - i), & 1 - \frac{l}{10} < n < l - 1, \end{cases} \tag{4.12}$$

  where the length $l \sim \mathcal{U}_{[48,144]}$, the random scaling factor $r \sim \mathcal{U}_{[0.5,1]}$, and the minimum power value $p_{\text{min}} = \min\{p_i, p_{i+1}, \ldots, p_{i+l-1}\}$.

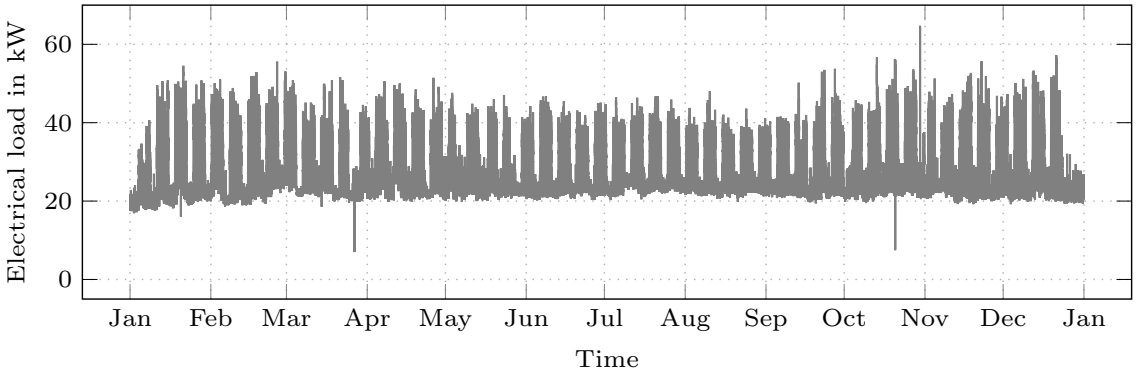
For the evaluation, we insert 10, 20, 30, 40, and 50 anomalies of each anomaly type into the selected time series. This corresponds to 3, 5, 8, 10, and 12 % of the data for the technical faults and 6, 11, 16, 21, and 26 % of the data for the unusual consumption.

Data With Labeled Anomalies As the second data set, we consider the electrical data collected on the Campus North of the Karlsruhe Institute of Technology (KIT), which we also use in Chapter 3 and in Turowski et al. (2022b) respectively to identify and model real-world anomalies. More precisely, we choose one of the 50 one-year power time series in which anomalies of the four identified anomaly types from the group of technical faults are labeled (see Section 3.1). The selected time series contains power consumption measurements from a typical mid-campus office building in 2016, is comparatively anomaly free and is shown in Figure 4.7a.

The selected time series contains 19 labeled anomalies of all four anomaly types from the group of technical faults that are described in Chapter 3. The 19 labeled anomalies



(a) With labeled anomalies of the four identified anomaly types from the group of technical faults. The labeled anomalies are plotted in red. Note that the labeled anomalies with a short length are not recognizable due to their length and that the y-axis is truncated for better graphical clarity.



(b) With anomalies compensated by imputation using the Copy-Paste Imputation (CPI) method (Weber et al. 2021).

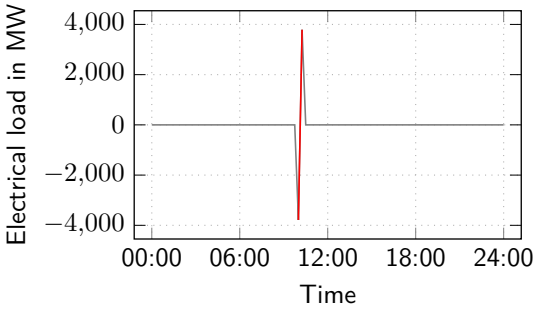
Figure 4.7 Overview of the selected one-year power time series P from the second data set with labeled anomalies and with compensated anomalies.

correspond to a 6 % share of the data. Table 4.1 lists all anomalies labeled in that time time series and Figure 4.8 shows an exemplary labeled anomaly of all four types.

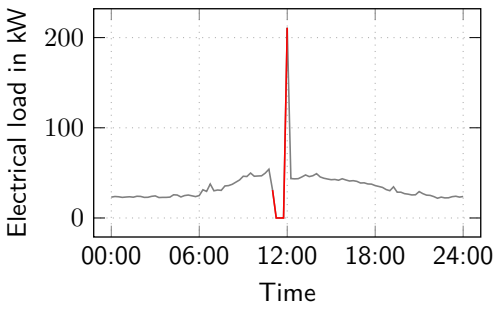
Since the selected time series already contains anomalies and the proposed approach requires anomaly-free training data for applying supervised anomaly detection methods, we perform the same procedure as in Section 3.3. To obtain the anomaly-free power time series P, we use the corresponding manually labeled one-year energy time series E. In this time series, we mark the labeled anomalies as missing values and apply the Copy-Paste Imputation (CPI) method (Weber et al. 2021), that imputes missing values with realistic patterns while preserving the amount of energy associated with the missing values. The CPI method directly provides the anomaly-free power time series P, which is shown in Figure 4.7b.

Table 4.1 Overview of the anomalies labeled in the selected time series from the electrical data collected at the Campus North of the KIT.

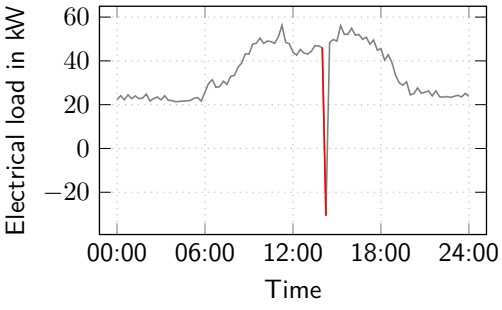
Number	Begin	End	Labeled anomaly type
1	18.01.2016 08:30	20.01.2016 12:15	Type 1
2	21.01.2016 11:45	21.01.2016 12:30	Type 1
3	21.01.2016 15:30	21.01.2016 15:45	Type 1
4	14.03.2016 12:45	14.03.2016 15:45	Type 2
5	14.03.2016 15:30	14.03.2016 15:45	Type 1
6	15.03.2016 11:00	15.03.2016 12:00	Type 2
7	18.03.2016 09:15	18.03.2016 10:15	Type 1
8	18.03.2016 12:00	18.03.2016 13:45	Type 2
9	24.03.2016 10:00	24.03.2016 10:30	Type 2
10	27.03.2016 02:00	27.03.2016 03:00	Type 1
11	12.04.2016 15:00	13.04.2016 01:30	Type 2
12	25.04.2016 07:15	25.04.2016 07:45	Type 2
13	15.08.2016 11:00	15.08.2016 11:15	Type 2
14	17.09.2016 12:30	20.09.2016 07:45	Type 2
15	27.09.2016 10:00	27.09.2016 10:15	Type 1
16	06.10.2016 09:30	06.10.2016 10:00	Type 2
17	20.10.2016 14:15	20.10.2016 14:15	Type 3
18	30.10.2016 03:00	30.10.2016 03:00	Type 4
19	07.11.2016 11:45	07.11.2016 12:00	Type 1



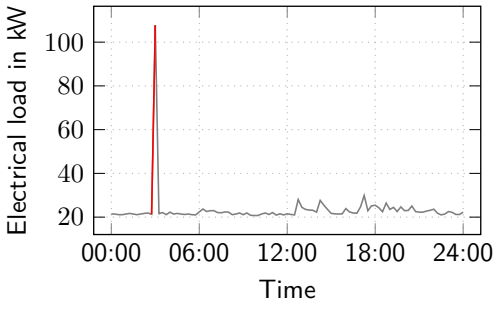
(a) Anomaly of type 1 on 27.09.2016.



(b) Anomaly of type 2 on 15.03.2016.



(c) Anomaly of type 3 on 20.10.2016.



(d) Anomaly of type 4 on 30.10.2016.

Figure 4.8 Examples of the labeled anomalies of types 1 to 4 from the technical faults in the selected power time series with labeled anomalies P. The anomalies are plotted in red. Note that the power time series of type 1 is in the MW scale and that the anomalies of types 3 and 4 actually have a length of one but are marked together with their previous value to be recognizable.

4.2.2 cINN and cVAE as Used Generative Methods

Since cINNs and cVAEs can be used as generative method in the proposed latent space-based approach, we perform the evaluation with a representative implementation from both generative methods. After introducing the selected implementations of the cINN and the cVAE, we describe the input data used for both generative methods.

cINN The selected cINN consists of 10 GLOW coupling layers (Kingma and Dhariwal 2018) that implement a type of generative flow. Each of them is followed by a random permutation and contains a subnetwork that allows the coupling layer to learn. We use a fully connected network as subnetwork. To account for conditional information, we use a conditioning network as proposed by Ardizzone et al. (2019). The conditioning network processes the conditional information and is also a fully connected network as proposed in Heidrich et al. (2023) (for details, see Table 4.2). For the training of the cINN, we apply a batch size of 512, the Adam optimizer (Kingma and Ba 2015), and a maximum of 50 epochs.

Table 4.2 Implementation details of the subnetwork and the conditioning network q in the used cINN.

(a) Subnetwork

Layer	Description
Input	[Output of previous coupling layer, conditional information]
1	Dense 32 neurons; activation: tanh
2	Dense horizon neurons; activation: linear

(b) Conditioning network

Layer	Description
Input	[Calendar information, statistical information]
1	Dense 8 neurons; activation: tanh
2	Dense 4 neurons; activation: linear

cVAE The selected cVAE comprises an encoder and a decoder. Both are fully connected networks (for details, see Table 4.4). For the training of the cVAE, we use a batch size of 512, the Adam optimizer (Kingma and Ba 2015), and a maximum of 100 epochs.

Input Data To train both generative methods for supervised anomaly detection, we use the first 15,000 data points of the selected time series of a data set. For unsupervised anomaly detection with both generative methods, we choose an appropriate contamination value, which we specify accordingly. Regardless of the supervised or unsupervised anomaly detection, both generative methods get standardized data points of the selected time series of a data set as samples with a size of 96. They also use the information contained in the time stamps of the considered time series as calendar information. It comprises

the hour of the day, the month of the year, and the weekday. As statistical information, the generative methods get the mean of the considered time series sample.

Table 4.4 Implementation details of the encoder and decoder of the used cVAE.

(a) Encoder

Layer	Description
Input	[Normal data, conditional information]
1	Dense 64 neurons; activation: tanh
2	Dense 32 neurons; activation: tanh
3	μ: dense latent dimension; activation: linear
4	σ: dense latent dimension; activation: linear

(b) Decoder

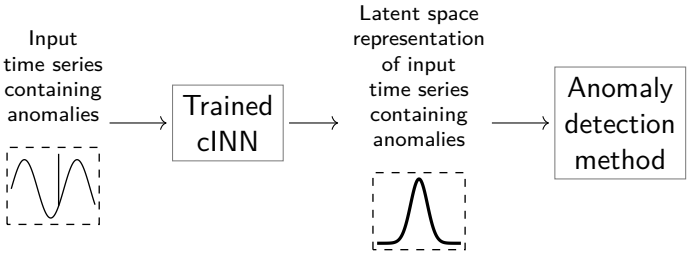
Layer	Description
Input	[Latent data, conditional information]
1	Dense 32 neurons; activation: tanh
2	Dense 64 neurons; activation: tanh
3	Dense horizon neurons; activation: linear

4.2.3 Data Representations for Comparison

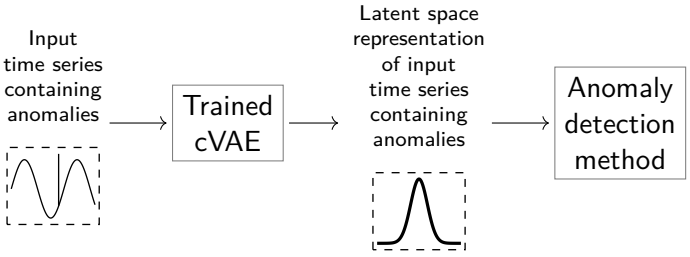
We consider different data representations in our experiments to compare the proposed approach of applying anomaly detection methods directly to the latent space representation to the common approach of applying anomaly detection methods directly or after scaling to the data containing anomalies (see Figure 4.9). More precisely, we use the selected time series of each data set to create four data representations. Two data representations are latent space representations of the considered data generated by the cINN and the cVAE as proposed in our approach. The other two data representations are the scaled and unscaled data representations that correspond to the common approach of applying anomaly detection methods directly or after scaling to the given data. These latter two data representations serve as benchmark data representations.

The first latent space data representation is from the cINN. For this, we standardize the selected time series and create overlapping samples with a size of 96 beginning every hour. These samples serve as an input for the trained cINN that generates the resulting latent space data representation. The second latent space data representation is from the cVAE. It is created in the same way as for the cINN.

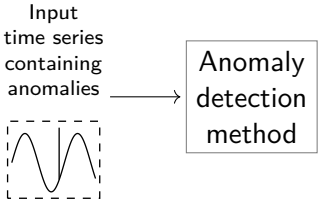
For the first benchmark data representation, we standardize the selected time series to obtain a scaled data representation, before creating overlapping samples with a size of 96 beginning every hour. For the second benchmark data representation, we use the unaltered time series as an unscaled data representation, from which we create overlapping samples with a size of 96 beginning every hour.



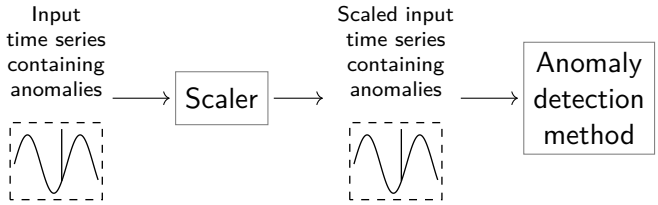
(a) A trained cINN creates the cINN latent space data representation of an input time series containing anomalies. Following the proposed approach, the cINN latent space data representation then serves as input to an anomaly detection method.



(b) A trained cVAE creates the cVAE latent space data representation of an input time series containing anomalies. Following the proposed approach, the cVAE latent space data representation then serves as input to an anomaly detection method.



(c) The unscaled data representation corresponds to the common approach of directly applying an anomaly detection method to an input time series containing anomalies.



(d) The scaled data representation corresponds to the common approach of applying an anomaly detection method to an input time series containing anomalies after scaling.

Figure 4.9 The four considered data representations to compare the proposed approach of applying anomaly detection methods directly to the latent space to the common approach of applying these methods directly or after scaling to the data containing anomalies. These data representations include the latent space data representations created by the cINN and the cVAE as well as the scaled and unscaled benchmark data representations.

4.2.4 Applied Anomaly Detection Methods

For the evaluation of our approach, we select existing anomaly detection methods and apply them to the four described data representations to detect anomalous and non-anomalous data. To consider different learning assumptions, i. e., inductive biases (Mitchell 1997), we select seven supervised and four unsupervised anomaly detection methods, which we briefly present below including their application.

Supervised Methods The selected supervised methods consider anomaly detection as a binary classification problem, where each data point is assigned the label anomaly or non-anomalous data.

As the first supervised detection method, we choose the k-Nearest Neighbor (kNN) method. It uses a proximity measure to classify a test sample based on the similarity of training instances (Cover and Hart 1967). As second method, we select the Logistic Regression (LogR). In the LogR for binary outcomes, the posterior probabilities of the outcomes are modeled with a logistic function (Hastie et al. 2009). We select the Multi-Layer Perceptron (MLP) as the third method. As an Artificial Neural Network, it approximates an arbitrary function through multiple hidden layers of interconnected nodes and applying activation functions between the layers (e.g., Werbos 1974; Mitchell 1997). The fourth method we choose is the Gaussian Naïve Bayes (NB). The NB estimates a conditional probability by assuming the conditional independence of the input features, given the prior probability of the output variable (Tan et al. 2019). As fifth method, we select the Random Forest (RF). The RF uses bagging to reduce the variance of an estimated prediction by combining the predictions of multiple decision trees (Breiman 2001; Hastie et al. 2009; Tan et al. 2019). As sixth method, we choose a Support Vector Machine for Classification (SVC) that maximizes the hyperplane between the binary classes to classify test samples (Vapnik 2000). We select XGBoost as the seventh method. It is a gradient boosting machine and optimizes a regularized objective function using gradient decent (Chen and Guestrin 2016).

Unsupervised Methods The selected unsupervised methods analyze the data to uncover the underlying normal behavior and then identify anomalous data points that violate this behavior.

As first unsupervised detection method, we select an Autoencoder (AE). It learns a mapping to the latent representation of the data and a mapping back to the reconstruction of the input (Rumelhart et al. 1986). The AE identifies samples as anomalous if their reconstruction error is above a threshold that we set to the quantile resulting from the contamination of the cINN or cVAE used. The second method is the Isolation Forest (iForest). It is an ensemble of isolation trees that randomly partitions samples in randomly selected features. The averaging path length of samples in different isolation trees serves as the indicator for anomalous data (Liu et al. 2008). As third method, we select the Local Outlier Factor (LOF). The LOF measures the distances of a sample to its k-nearest

neighbors to estimate the local density. By comparing the local density to the local densities of its neighborhood, the LOF is able to identify samples as non-anomalous or anomalous (Breunig et al. 2000). The fourth method is a Variational Autoencoder (VAE). It learns the probability distribution of the data in the latent space to reconstruct its input (Kingma and Welling 2014). The VAE also compares the reconstruction error to a threshold that we set to the quantile resulting from the contamination of the used cINN or cVAE to identify samples as non-anomalous or anomalous.

Application To apply the unsupervised detection methods, we use the complete selected time series of each data set. To apply the supervised detection methods, however, we split the selected time series of each data set to obtain a training and a test set. We use the first 5,000 data points as training set. As test set, we use all data points except the first 15,000 data points because these 15,000 data points are used for the training of the generative methods.

To both sets, we apply the selected supervised detection methods with default hyperparameters, i. e., the hyperparameters set as default in the available implementation, and with the best-performing hyperparameters. To determine the best-performing hyperparameters, we choose hyperparameters and select corresponding values for each method (see Table B.1 in Appendix B). Over the resulting hyperparameter grid, we perform a cross-validated grid search on the training set. We choose the parameters that yield the best F1-Score (Equation (4.13)) for a data representation and group of anomalies as best-performing hyperparameters for this data representation and group of anomalies (see Tables B.2 to B.13 for the data with synthetic anomalies and Tables B.14 to B.19 for the data with labeled anomalies in Appendix B).

4.2.5 Evaluation Criteria

To quantitatively evaluate the selected anomaly detection methods on the four data representations, we use three evaluation criteria.

The first evaluation criterion is the detection performance of the methods. For this criterion, we choose the commonly used F1-Score as metric, which is the harmonic mean of precision and recall. It is calculated on the samples and defined as

$$\text{F1-Score} = \frac{\text{TP}}{\text{TP} + \frac{1}{2} \cdot (\text{FP} + \text{FN})}, \tag{4.13}$$

where TP are the true positives, FP the false positives, and FN the false negatives in relation to the inserted synthetic or labeled anomalies. In the calculation of the F1-Score, a sample is considered as an anomaly as soon as one of its data points is an inserted synthetic or labeled anomaly.

The second evaluation criterion is the robustness of the methods' detection performance. To assess the detection robustness, we calculate the F1-Score for different shares of inserted synthetic anomalies.

The third evaluation criterion is the computational cost of the detection. To assess the computational cost, we measure run-times. For the supervised anomaly detection methods, we measure the run-times for training the supervised cINN and cVAE, finding the methods' best hyperparameters, and for training them given the best-performing hyperparameters. Similarly, for the unsupervised anomaly detection methods, we determine the run-times for training the unsupervised cINN and cVAE and fitting the methods.

4.2.6 Hard- and Software

For a better comparability of the results, we use the same hardware throughout the evaluation, namely a 48 core system with 256 GB RAM, where each core has 2.1 GHz. Furthermore, all selected detection methods are implemented in Python. More specifically, for XGBoost, we use its available implementation[4] (Chen and Guestrin 2016) and Keras[5] (Chollet et al. 2015) for the AE and VAE; for all other anomaly detection methods, scikit-learn[6] (Pedregosa et al. 2011). The cINN is implemented with FrEIA[7] and PyTorch[8] (Paszke et al. 2019) and the cVAE with PyTorch (Paszke et al. 2019). To automate the evaluation with these implementations, we additionally use pyWATTS[9] (Heidrich et al. 2021).

4.3 Results

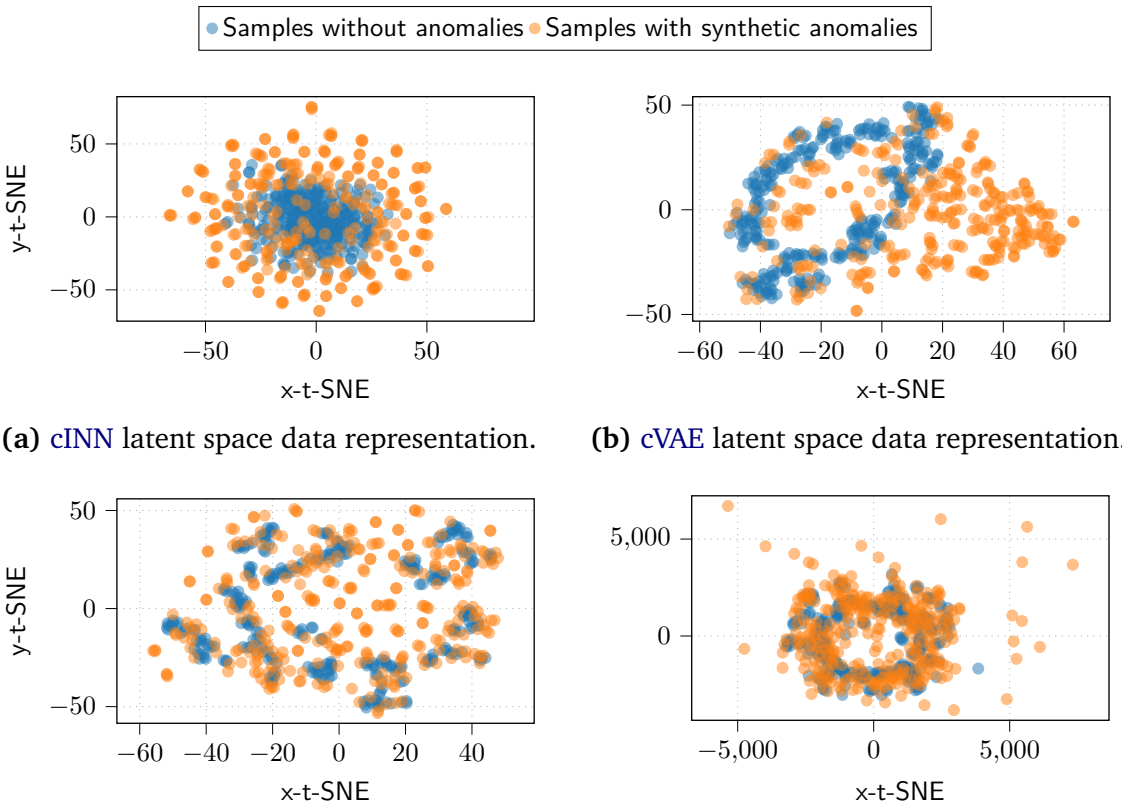
To qualitatively evaluate the benefit of using the latent space data representation in anomaly detection, we first visualize how the four data representations separate anomalies and non-anomalous data. Second, we report the quantitative evaluation criteria when applying supervised and unsupervised anomaly detection methods to the data representations containing inserted synthetic anomalies. For the supervised and unsupervised anomaly detection methods, we also present the results for the data containing labeled anomalies. Since the scaled and unscaled data representations perform similarly, we focus on the scaled data representation in the following.

4 https://xgboost.ai/
5 https://keras.io/
6 https://scikit-learn.org/
7 https://github.com/VLL-HD/FrEIA
8 https://pytorch.org/
9 https://github.com/KIT-IAI/pyWATTS

4.3.1 Visualization of Anomalies in Data Representations

To analyze how the four data representations separate anomalies and non-anomalous data, we randomly choose 300 samples without anomalies and 300 samples with anomalies from the supervised detection methods' test set of the data with inserted synthetic anomalies. We visualize the cINN latent space, cVAE latent space, scaled, and unscaled data representations of the chosen samples with a t-distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton 2008) using two dimensions. For an optimal anomaly detection, samples with anomalies and samples without anomalies should be clearly separated without overlapping.

Figures 4.10 and 4.11 show the resulting t-SNE visualizations of samples with anomalies of technical faults and unusual consumption and samples without anomalies for the four data representations. For both groups of anomalies, the t-SNE visualizes less overlap between samples with anomalies and samples without anomalies for the cINN latent space data representation than for the scaled data representation. Furthermore, for the cINN latent space data representation, the samples with anomalies are grouped around the main cluster of samples without anomalies.



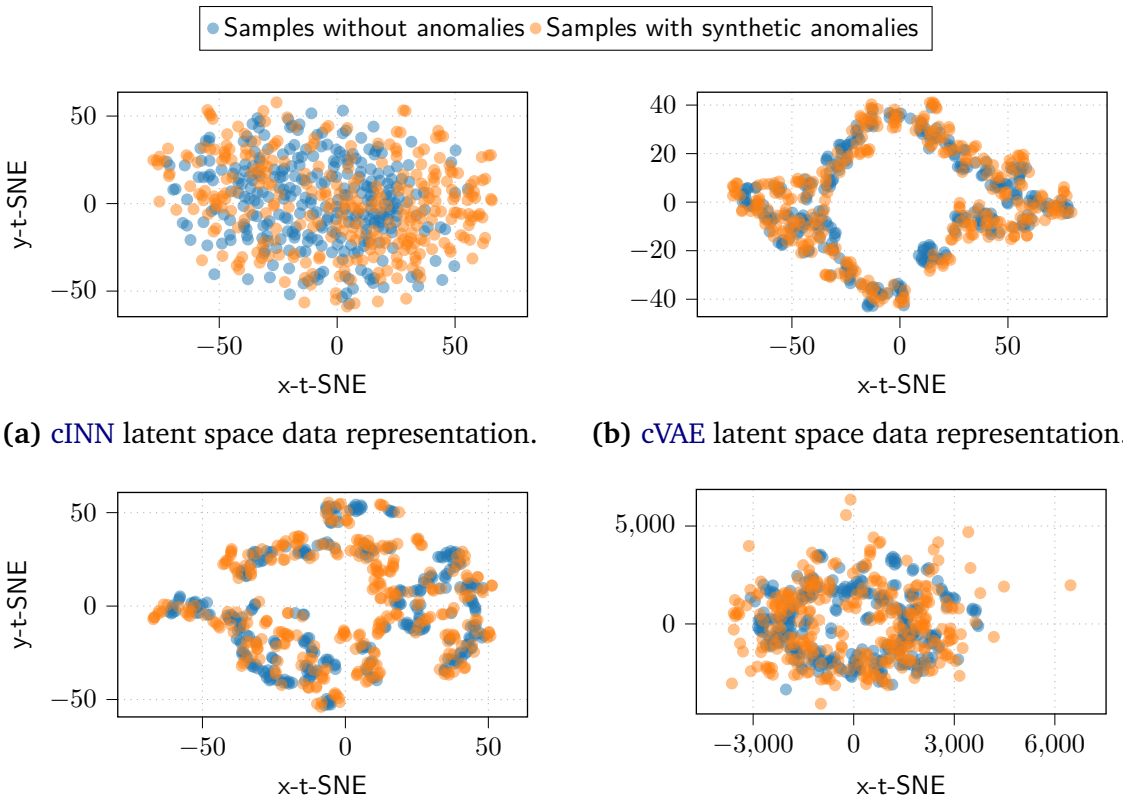
(a) cINN latent space data representation.

(b) cVAE latent space data representation.

(c) Scaled data representation.

(d) Unscaled data representation.

Figure 4.10 t-SNE visualizations of 300 random samples without anomalies and 300 random samples with synthetic anomalies in the cINN latent space, the cVAE latent space, the scaled, and the unscaled data representations with 20 inserted synthetic anomalies of technical faults.



(a) cINN latent space data representation.

(b) cVAE latent space data representation.

(c) Scaled data representation.

(d) Unscaled data representation.

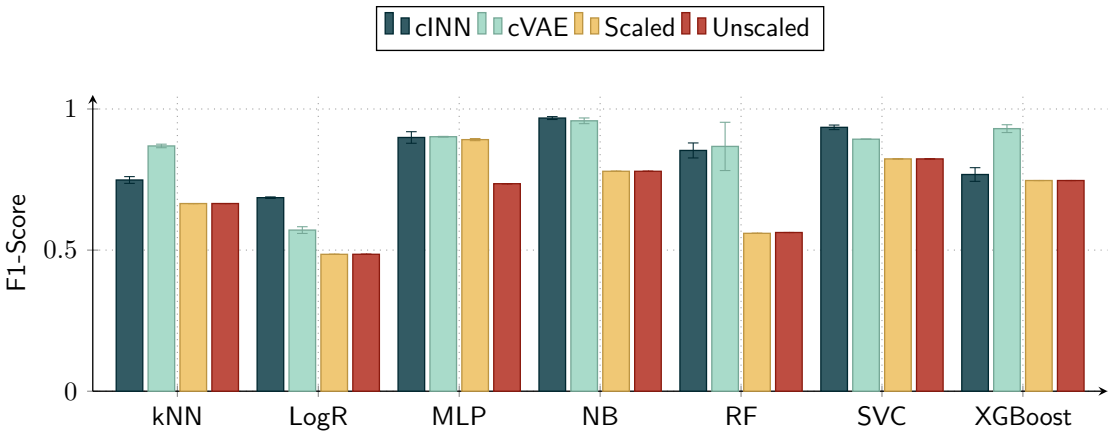
Figure 4.11 t-SNE visualizations of 300 random samples without anomalies and 300 random samples with synthetic anomalies in the cINN latent space, the cVAE latent space, the scaled, and the unscaled data representations with 20 inserted synthetic anomalies of unusual consumption.

4.3.2 Data Representations in Supervised Anomaly Detection

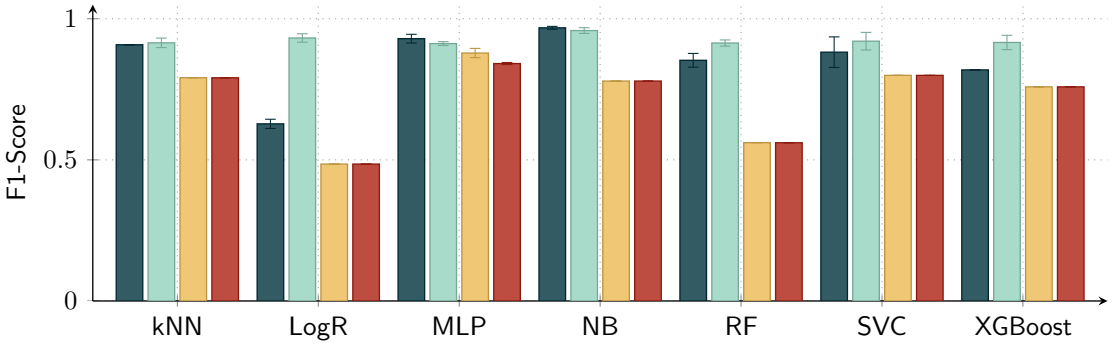
We first report the three evaluation criteria for the data with inserted synthetic anomalies, before showing the results for the data with labeled anomalies.

Detection Performance We evaluate the detection performance of the supervised anomaly detection methods with both default and best-performing hyperparameters for technical faults and unusual consumption. For both groups of anomalies, we insert 20 anomalies of each type belonging to this group. Figure 4.12a and Figure 4.12b show the resulting F1-Scores for the technical faults and Figure 4.13a and Figure 4.13b for the unusual consumption. For each supervised method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst observed F1-Scores in multiple runs using varying random initialization for the cINN, cVAE, and the detection methods.

With default hyperparameters, all evaluated methods yield the best F1-Scores for both groups of anomalies when using a latent space data representation. Compared to the scaled



(a) Technical faults with default hyperparameters.

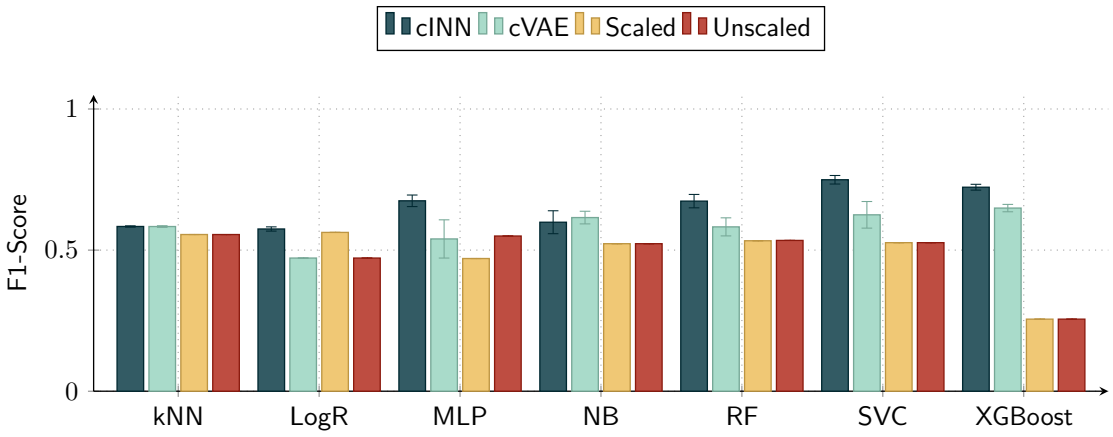


(b) Technical faults with the best performing hyperparameters.

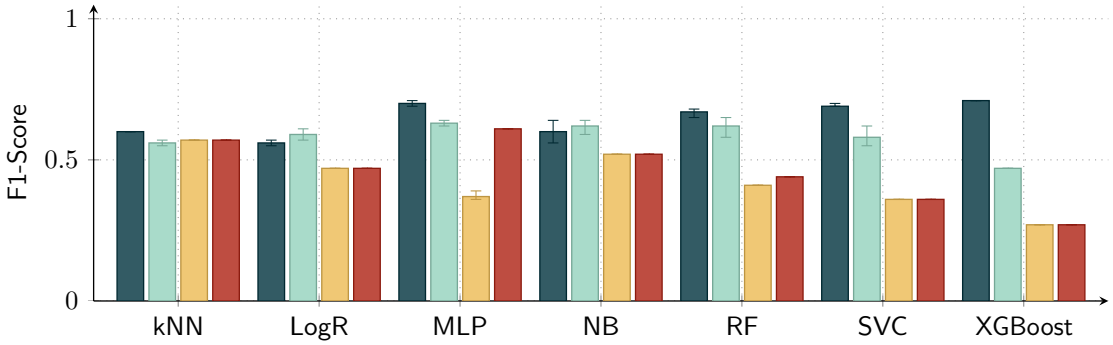
Figure 4.12 The F1-Scores of the seven supervised detection methods applied to the data with 20 synthetic anomalies of each type from the technical faults. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst observed F1-Scores.

data representation, the F1-Scores of the cINN latent space representation are 21 % better on average, ranging from 1 % for the MLP to 52 % for the RF. The F1-Scores of the cVAE latent space representation are 23 % better on average, ranging from 1 % for the MLP to 55 % for the RF. Note that, despite the general improvement through using a latent space data representation, the F1-Scores strongly vary between the evaluated methods. For example, considering the cINN latent space data representation and the technical faults, the LogR yields a F1-Score of 0.69, while the NB achieves a F1-Score of 0.97.

With the best-performing hyperparameters, all evaluated methods also perform best using the latent space data representation for both groups of anomalies. Compared to the scaled data representation, the F1-Scores of the cINN latent space representation for the technical faults are 21 % better on average, ranging from 6 % for the MLP to 52 % for the RF. The F1-Scores of the cVAE latent space representation are 34 % better on average, ranging from 4 % for the MLP to 92 % for the LogR. Note that we again observe highly varying F1-Scores across all evaluated methods.



(a) Unusual consumption with default hyperparameters.

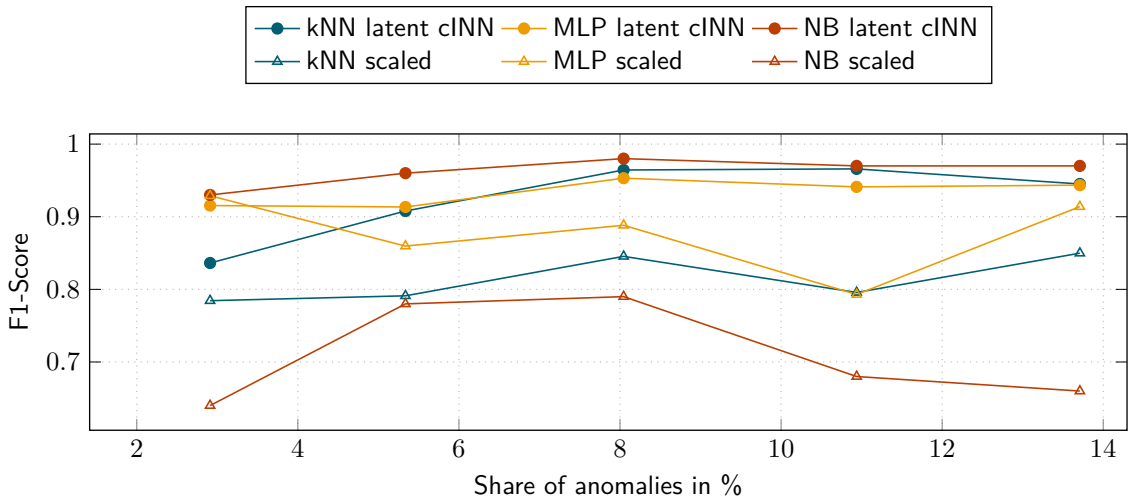


(b) Unusual consumption with the best performing hyperparameters.

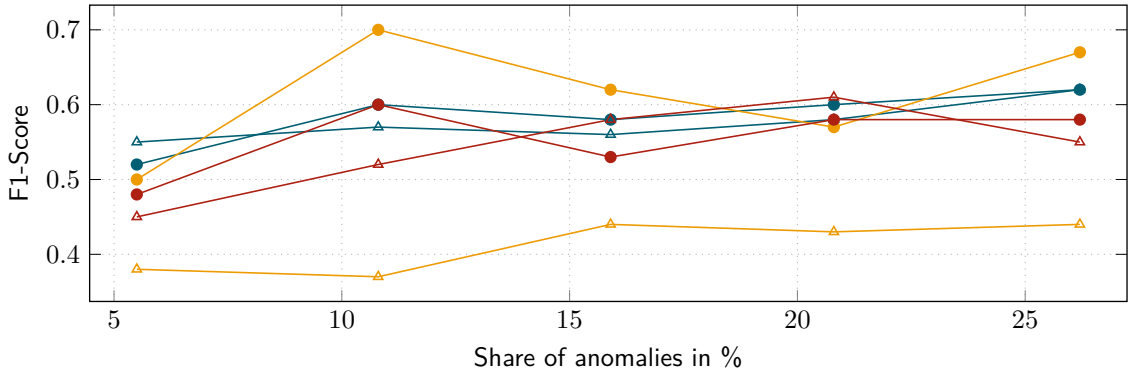
Figure 4.13 The F1-Scores of the seven supervised detection methods applied to the data with 20 synthetic anomalies of each type from the unusual consumption. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst observed F1-Scores.

Detection Robustness Regarding different shares of anomalies, we examine the kNN, MLP, and NB as the three best methods when using best-performing hyperparameters determined for 20 anomalies of each type from technical faults and unusual consumption respectively. For the sake of brevity, we only consider the cINN latent space and the scaled data representations. The average F1-Scores for different shares of anomalies is shown in Figure 4.14a for the technical faults and in Figure 4.14b for the unusual consumption (for all other methods, see Figure B.1 in Appendix B).

For the technical faults, the F1-Scores based on the cINN latent space data representation are consistently higher than those for the scaled data representation across all shares of anomalies. Furthermore, all anomaly detection methods perform more consistently when using the cINN latent space data representation, showing less variation than the scaled data representations. For unusual consumption, the F1-Scores when using the cINN latent space data representation are noticeably better for the MLP when compared to the scaled data representation, and similar for the kNN and NB.



(a) Technical faults



(b) Unusual consumption

Figure 4.14 The F1-Scores of the three best-performing supervised detection methods applied to the data with different shares of synthetic anomalies from technical faults and unusual consumption using the best-performing hyperparameters. For each method, one line each indicates the resulting F1-Score for the cINN latent space and scaled data representations.

Computational Cost Concerning the computational cost reported in Table 4.6, we first compare the run-times required to train the supervised cINN and cVAE with the run-times to find the best-performing hyperparameters and to train the supervised methods given selected hyperparameters. Afterward, we compare the run-times of the hyperparameter optimization and of the training of the supervised methods with respect to the four data representations.

The supervised training of the used cINN and cVAE takes considerably less time than the hyperparameter optimization of the MLP, SVC, and XGBoost, about the same as the hyperparameter optimization of the kNN, and more time than the hyperparameter optimization of the LogR and RF. Compared to the training of the methods on all data representations, the supervised training of the cINN and cVAE, however, generally requires some more time.

Table 4.6 The required run-times in seconds to train the supervised cINN and cVAE, to find the best-performing hyperparameters of the supervised detection methods, and to train them given the best-performing hyperparameters for the four data representations.

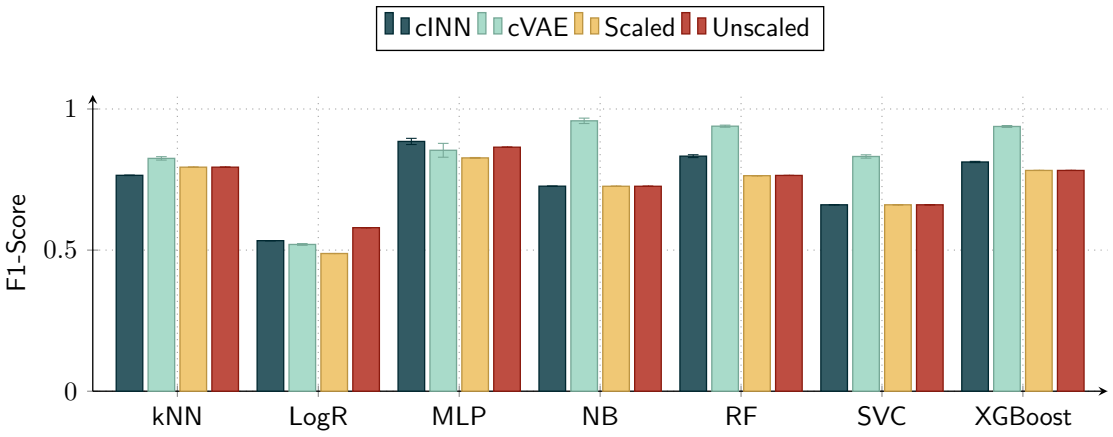
		cINN	cVAE	Scaled	Unscaled
Supervised training		60.77	22.24	0	0
Detection method's hyperparameter optimization	kNN	44.43	10.46	42.56	41.25
	LogR	6.29	2.55	6.35	12.61
	MLP	2694.54	9116.88	9286.41	3125.75
	RF	6.04	3.71	5.20	5.00
	SVC	5906.27	4.86	899.70	11863.58
	XGBoost	1912.33	1048.95	1631.59	1631.59
Detection method's training	kNN	0.00	0.00	0.00	0.00
	LogR	0.66	0.91	0.66	0.84
	MLP	3.24	7.37	7.58	1.64
	RF	0.83	0.48	0.86	0.86
	SVC	0.11	0.13	0.29	0.30
	XGBoost	1.48	1.10	1.47	1.70

The hyperparameter optimization itself requires different amounts of time depending on the data representation. On the cINN latent space data representation and compared to the scaled data representation, the optimization takes noticeably less time for the MLP, about the same for the kNN, LogR, and RF, and more time for the SVC and XGBoost. On the cVAE latent space data representation, the optimization takes less time for the kNN, SVC, and XGBoost, about the same time for the LogR and RF, and longer for the MLP. Note, however, that the run-time required for the hyperparameter optimization varies greatly across the methods.

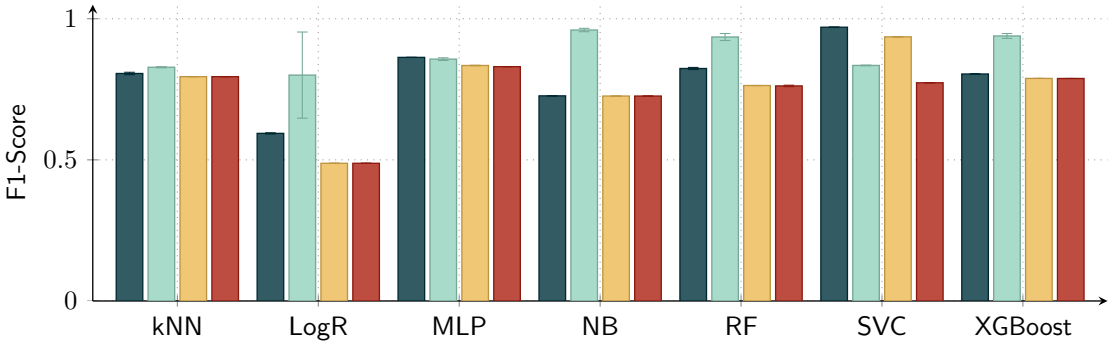
The run-times for training the supervised methods also depend on the data representation. Using the latent space data representations for the training requires less or about the same time as using the scaled data representation for most supervised methods.

Data With Labeled Anomalies For the data with labeled anomalies, we focus on the detection performance since the considered data already contains real-world technical faults. For this data, we also evaluate the detection performance of the supervised anomaly detection methods with both default and best-performing hyperparameters. Figure 4.15 shows the resulting F1-Scores for the data with labeled anomalies which are all technical faults. For each supervised method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst observed F1-Scores.

All evaluated methods achieve F1-Scores with default hyperparameters using a latent space data representation that are at least as good as those obtained with the scaled data representation. Compared to the scaled data representation, the F1-Scores of the cINN latent space representation are 4 % better on average, ranging from −4 % for the



(a) Technical faults with default hyperparameters.



(b) Technical faults with the best performing hyperparameters.

Figure 4.15 The F1-Scores of the seven supervised detection methods applied to the data with labeled technical faults. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst observed F1-Scores.

kNN to 9 % for the LogR and RF. The F1-Scores of the cVAE latent space representation are 16 % better on average, ranging from 3 % for the MLP to 32 % for the NB.

Similarly, all evaluated methods also obtain F1-Score with the best-performing hyperparameters using a latent space data representation that are at least as good as those obtained with the scaled data representation. Compared to the scaled data representation, the F1-Scores of the cINN latent space representation are 6 % better on average, ranging from 0 % for the NB to 22 % for the LogR. The F1-Scores of the cVAE latent space representation are 19 % better on average, ranging from −11 % for the SVC to 64 % for the LogR. Note that we also observe for this data varying F1-Scores across all evaluated methods.

4.3.3 Data Representations in Unsupervised Anomaly Detection

For the unsupervised anomaly detection methods, we again first report the three evaluation criteria for the data with inserted synthetic anomalies, before presenting the results for the data with labeled anomalies.

Detection Performance To evaluate the detection performance of the unsupervised anomaly detection methods, we first use latent space data representations from an unsupervised cINN and cVAE trained with a contamination of 0.05 for the technical faults and 0.1 for the unusual consumption. Afterwards, we examine the effect of different contamination values for both groups of anomalies.

For the unsupervised cINN and cVAE using data with 20 inserted anomalies of each type from an anomaly group, Figure 4.16 presents the F1-Scores of the unsupervised detection methods. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst observed F1-Scores.

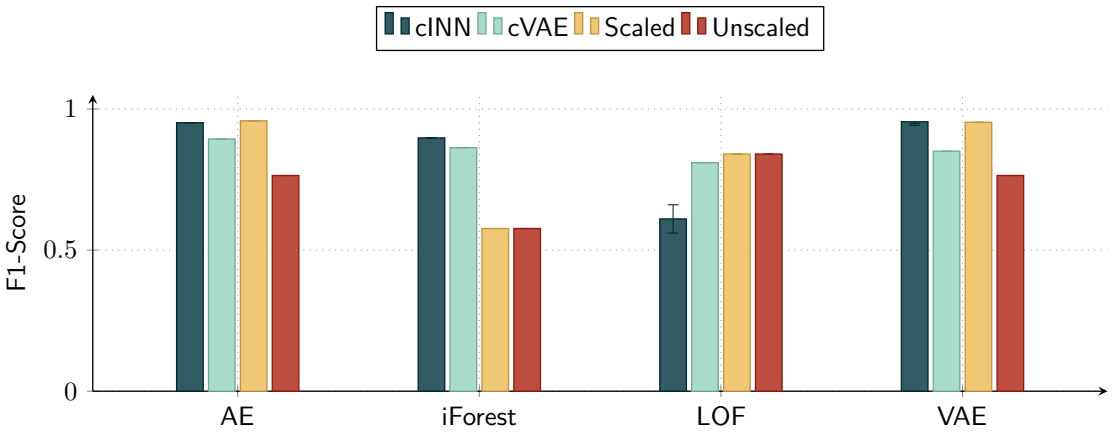
We observe that unsupervised detection methods perform differently when using the cINN and cVAE latent space data representations. Compared to the scaled data representation, the F1-Scores for the technical faults and the cINN latent space data representation show an improvement for the iForest, a similar performance for the AE and VAE, and a worse performance for the LOF. Furthermore, for the unusual consumption, the cINN and cVAE latent space data representations results in similar or lower F1-Scores than those from unsupervised anomaly detection methods using the scaled data representation.

For a contamination of 0.05, 0.1, 0.15, 0.2, and 0.25, Figures 4.17a and 4.17b show the average F1-Scores of the unsupervised detection methods on the cINN and cVAE latent space data representations for the technical faults and unusual consumption respectively.

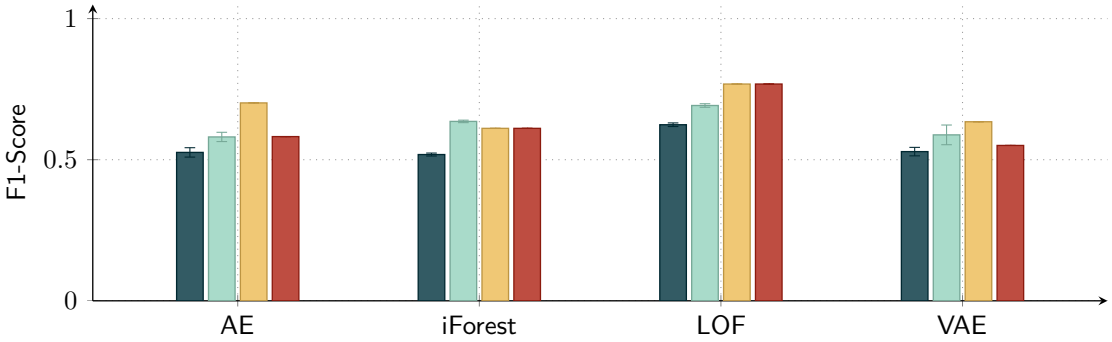
For technical faults, the detection methods achieve varying F1-Scores across the different contamination values, with the best F1-Scores for all methods, except for the LOF, occurring with a contamination of 0.05. The performance for unusual consumption varies more, with the best F1-Scores being achieved with a contamination of 0.05, 0.1, or 0.15, depending on the considered method.

Detection Robustness Regarding the analysis of different shares of anomalies from technical faults and unusual consumption, we again only consider the cINN latent space and the scaled data representations. Figure 4.18 shows the average F1-Scores of the detection methods for these data representations. For each share of anomalies, a corresponding contamination is used for the training of the cINN.

Compared to the scaled data representation, the F1-Scores of the cINN latent space data representation for technical faults are higher for the iForest, similar for the AE and VAE, and lower for the LOF. When considering unusual consumption, the latent space data representation results in lower F1-Scores across all shares of anomalies.



(a) Technical faults



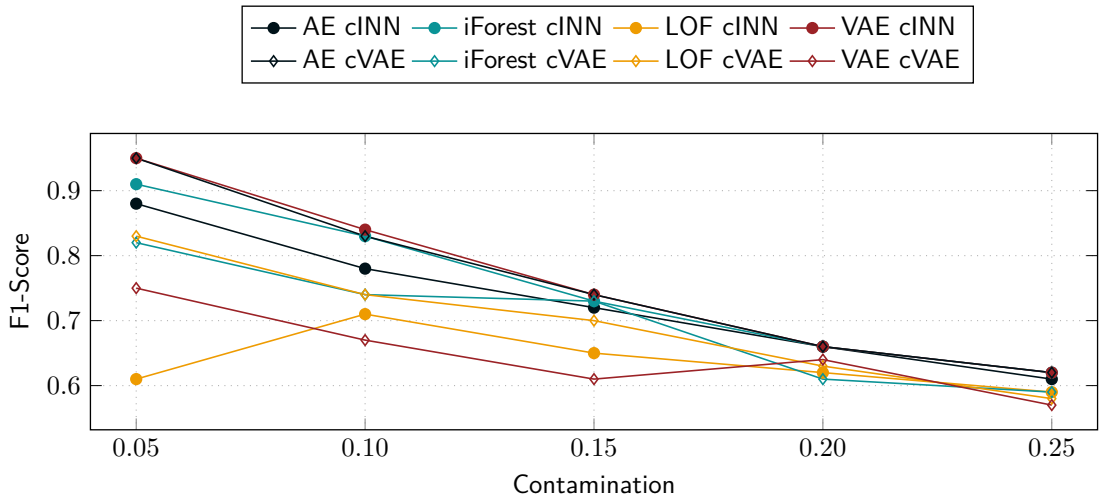
(b) Unusual consumption

Figure 4.16 The F1-Scores of the four unsupervised detection methods applied to the data with 20 synthetic anomalies of each type from technical faults and unusual consumption. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst F1-Scores.

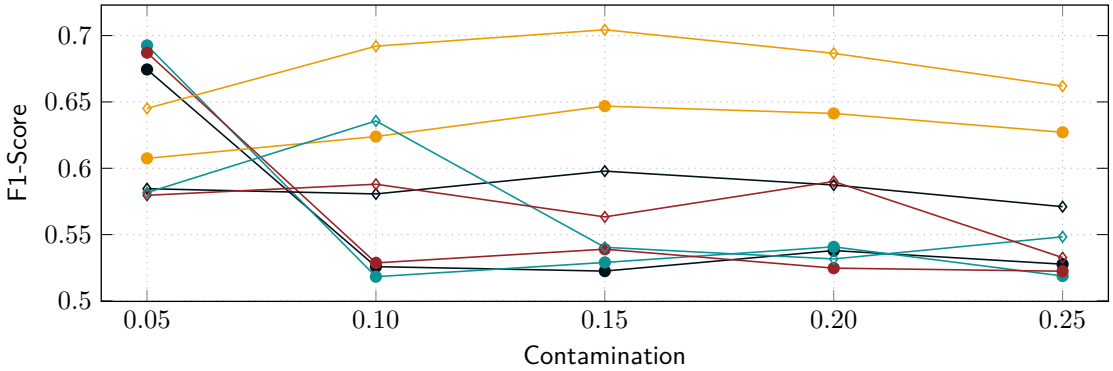
Computational Cost Regarding the four data representations, the run-times required to train the unsupervised cINN and cVAE and to fit the unsupervised methods are reported in Table 4.7.

The unsupervised training of the cINN and cVAE requires considerably more time than the fitting of most of the unsupervised methods. Additionally, fitting the AE and VAE is quicker, the iForest is similar, and the LOF is slower on the latent space compared to the scaled data representation.

Data With Labeled Anomalies For the data with labeled anomalies, we again focus on the detection performance because it already contains real-world technical faults. For this data, we first also evaluate the detection performance of the unsupervised anomaly detection methods using latent space data representations from an



(a) Technical faults

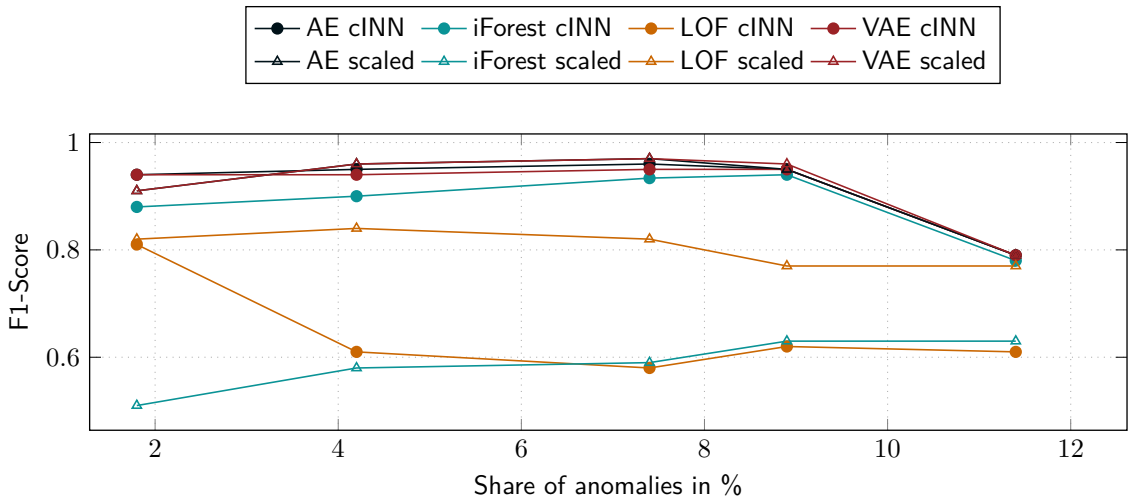


(b) Unusual consumption

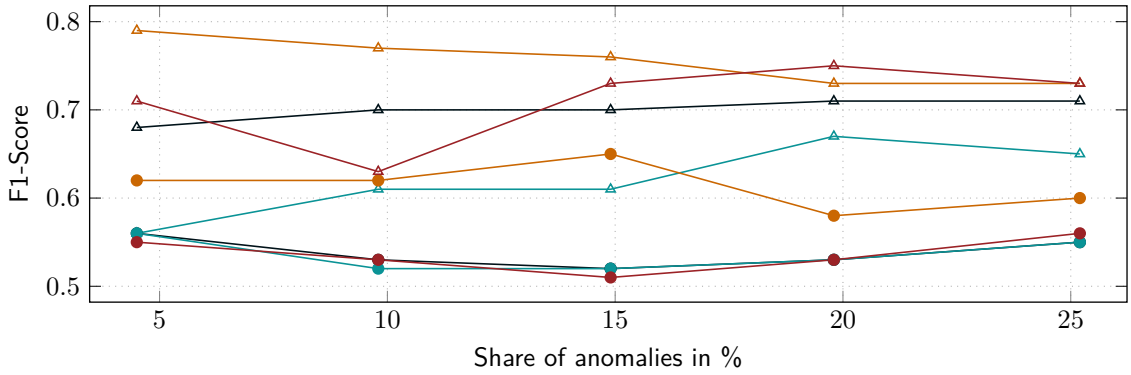
Figure 4.17 The F1-Scores of the four unsupervised detection methods applied to the latent space data representations created by an unsupervised cINN and cVAE with different contamination values. The data contains 20 synthetic anomalies of each type from technical faults and unusual consumption, which corresponds to 5 % of the data for technical faults and 11 % for unusual consumption.

Table 4.7 The required run-times in seconds to train the unsupervised cINN and cVAE and to fit the unsupervised detection methods regarding the four data representations.

		cINN	cVAE	Scaled	Unscaled
Unsupervised training		632.96	510.22	0	0
Detection method's fitting	AE	443.71	99.76	579.79	320.27
	iForest	16.21	3.84	29.53	23.33
	LOF	370.23	269.11	207.98	206.81
	VAE	856.12	76.00	4555.05	416.99



(a) Technical faults



(b) Unusual consumption

Figure 4.18 The F1-Scores of the four unsupervised detection methods applied to the data with different shares of synthetic anomalies from technical faults and unusual consumption. For each method, one line each indicates the resulting F1-Score for the cINN latent space, scaled, and unscaled data representations. For the latent space data representation, the unsupervised cINN is trained with a contamination corresponding to the share of anomalies in the data.

unsupervised cINN and cVAE trained with a contamination of 0.05. We then also examine the effect of different contamination values.

For the unsupervised cINN and cVAE, Figure 4.19 shows the F1-Scores of the unsupervised detection methods for the data with labeled anomalies which are all technical faults. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst observed F1-Scores.

All unsupervised detection methods except the LOF perform better compared to the scaled data representation when using a latent space data representation. The LOF using the

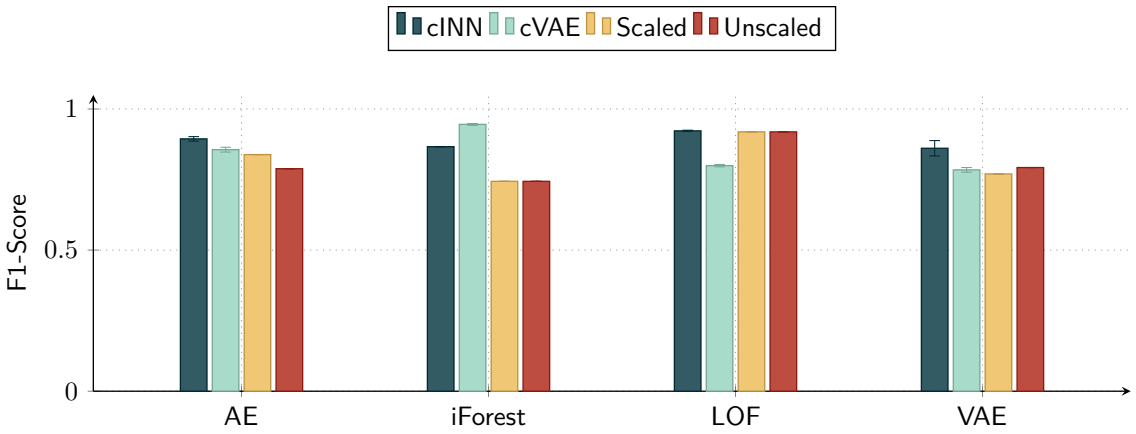


Figure 4.19 The F1-Scores of the four unsupervised detection methods applied to the data with labeled technical faults. For each method, the bars indicate the average F1-Score for the cINN latent space, cVAE latent space, scaled, and unscaled data representations. The error bars show the best and the worst F1-Scores.

cINN latent space data representation shows a similar performance as with the scaled data representation but a worse performance for the cVAE latent space data representation.

For a contamination of 0.025, 0.05, 0.1, 0.2, and 0.25, Figure 4.20 shows the average F1-Scores of the unsupervised detection methods on the cINN and cVAE latent space data representations. The detection methods achieve different F1-Scores across the considered contamination values, with the best F1-Scores for all methods, except for the VAE and the cVAE latent space data representation, obtained at a contamination of 0.05.

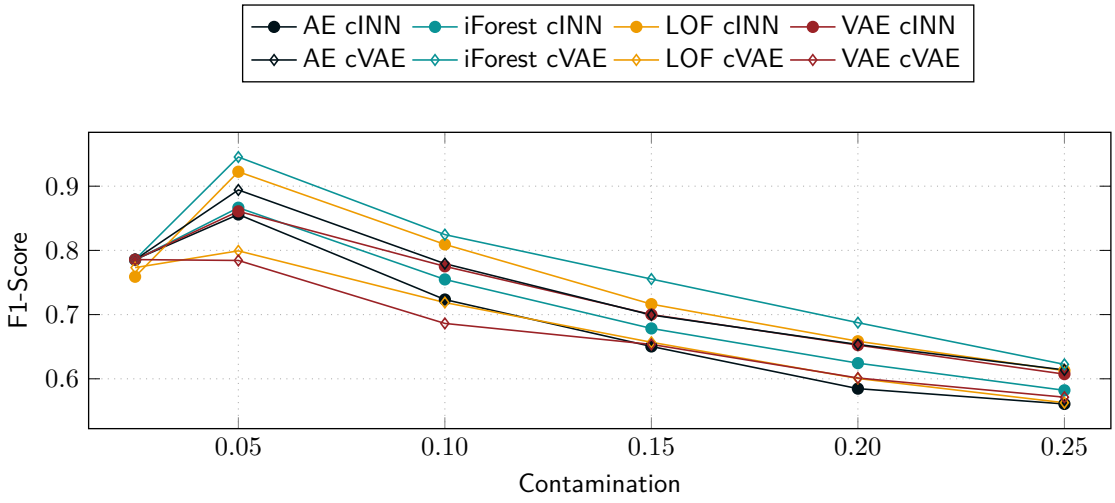


Figure 4.20 The F1-Scores of the four unsupervised detection methods applied to the latent space data representations created by an unsupervised cINN and cVAE with different contamination values. The data contains labeled technical faults, which corresponds to 5 % of the data.

4.4 Discussion

In this section, we discuss the reported results and the benefits of the proposed approach to enhance anomaly detection methods. First, we focus on the visualization of anomalies in different data representations, before considering the three evaluation criteria for the supervised and unsupervised anomaly detection methods and the data with inserted synthetic anomalies as well as the detection performance on the data with labeled anomalies. Finally, we discuss limitations and the overall benefits of the proposed approach.

From the initial t-SNE visualization of synthetic anomalies in different data representations in Figures 4.10 and 4.11, we conclude that the latent space data representation helps to better separate anomalies and non-anomalous data. This separation is clearer for technical faults, but still noticeable for unusual consumption. The visualization thus confirms that using the latent space data representation as in the proposed approach is beneficial.

The detection performance and robustness of the selected supervised and unsupervised detection methods also support this observation. For the evaluated detection methods and the data with inserted synthetic anomalies, the results show that directly using a latent space data representation improves supervised anomaly detection methods for both technical faults and unusual consumption, and partly improves unsupervised anomaly detection methods when considering technical faults. For the supervised methods, this improvement occurs even without performing hyperparameter optimization and is independent of the share of inserted synthetic anomalies. For unsupervised methods, the improvement is only partly noticeable for technical faults, with unusual consumption performing similarly or worse. However, since the used anomalies of unusual consumption are similarly shaped and difficult to detect even for a human, a better detection performance for such anomalies might only be possible if they are labeled beforehand. Furthermore, the LOF performs worse on the latent space data representation for both technical faults and unusual consumption. This suggests that the latent space data representation may not be suited for density-based anomaly detection methods and this phenomenon should be investigated further in future work. For the evaluated detection methods and the data containing labeled technical faults, however, using a cINN or cVAE latent space data representation generally improves the detection performance of all considered supervised and unsupervised anomaly detection methods. Nevertheless, the one year of considered data containing labeled technical faults is limited, since the limited number of 19 labeled anomalies is rather unevenly distributed over time. Therefore, future work should investigate whether our observations can be verified by further, more evenly distributed data.

With regards to computational time, for some supervised detection methods, the proposed anomaly detection method reduces the time required for hyperparameter optimization and the methods' training. At the same time, for the unsupervised detection methods, the proposed anomaly detection method does not reduce the fitting time. Therefore, the proposed approach can also be beneficial for the hyperparameter optimization and the methods' training.

Nevertheless, these improvements in the detection performance come with limitations. One limitation is the computational cost for the required trained cINN or cVAE. For the supervised cINN or cVAE, the required training time is considerably smaller than the time needed for hyperparameter optimization for some supervised detection methods; for the others, the time needed is in the same order of magnitude. For the unsupervised cINN or cVAE, the training takes noticeably longer than the fitting of the evaluated unsupervised detection methods, possibly due to the calculation of the quantile and the filtering of the errors for each batch. Additionally, unlike the supervised cINN and cVAE, the unsupervised cINN and cVAE are trained on all available data. Furthermore, since all available data are used, we must choose a suitable contamination to enable the cINN and cVAE to provide a beneficial data representation for unsupervised detection methods.

Considering the mentioned aspects, applying supervised detection methods without hyperparameter optimization directly on the latent space data representation as proposed is advantageous. For these methods, the training time of the cINN or cVAE is often considerably shorter than the time required to optimize their hyperparameters. At the same time, their detection performance remains high, even without hyperparameter optimization.

Another limitation of the proposed approach is the use of samples. By using samples, the proposed approach can only determine whether a sample contains an anomaly or not; it is not able to locate the exact position and length of an anomaly. If the amount of available data is sufficient, samples containing anomalies can be excluded for down-stream applications. For the case of limited data, future work should explore how the proposed approach can be extended to precisely locate anomalies in a sample.

A further limitation of the proposed approach is the assumption used in the training of the generative method. For the supervised anomaly detection, the proposed approach assumes that all anomalies contained in the used data are labeled. However, the used data may still include anomalies that have not been identified during labeling. For this reason, a thorough labeling process has to be performed, ideally by domain experts, to identify all contained anomalies. For the unsupervised anomaly detection, the proposed approach assumes that the minority of the used training data is anomalous and that the training errors are higher for anomalous data points than for non-anomalous data points. While this assumption is true for distinct anomalies such as the used technical faults, it is not necessarily true for anomalies that comprise typical patterns at uncommon levels, such as the considered unusual consumption. As a result, the used generative method could learn a mapping that is unfavorable for the subsequently applied unsupervised anomaly detection method, which could explain the comparatively worse performance of the proposed approach for unsupervised methods and unusual consumption. Therefore, future work could investigate how the proposed approach could be adapted to improve the detection of unusual consumption with unsupervised anomaly detection methods.

Overall, the proposed approach provides several benefits. The most important one is that it generally considerably enhances the detection performance of supervised and unsupervised detection methods. This way, the latent space data representation created by a cINN or cVAE can serve as a beneficial input for any existing detection method at only

moderate computational cost. This performance improvement is particularly noticeable for synthetic and labeled technical faults in both supervised and unsupervised anomaly detection methods; for unusual consumption only in supervised methods. However, the considered technical faults are assumed to have more impact on down-stream applications and thus should be prioritized. Therefore, the high performance for technical faults and improved supervised performance for unusual consumption imply that our approach is suitable to enhance anomaly detection methods.

4.5 Contribution and Future Work

In the present chapter, we examine how anomaly detection methods for energy time series can be enhanced. To answer the related research question [RQ2], we qualitatively examine the latent space data representations created with a cINN and cVAE by visualizing the separation of synthetic anomalies and non-anomalous data. We also quantitatively evaluate the anomaly detection performance using this latent space data representation by applying selected supervised and unsupervised anomaly detection methods to real-world electrical power time series containing inserted synthetic anomalies of two groups, namely technical faults and unusual consumption. We additionally apply the selected detection methods to real-world data containing labeled technical faults.

Based on this approach, the present chapter provides the following contributions:

- We propose an approach for directly using the latent space data representation to enhance anomaly detection methods.
- We show that the latent space data representation enhances anomaly detection, since it results in a clearer separation between time series samples with synthetic anomalies and samples without anomalies.
- We demonstrate that the proposed approach generally improves the detection performance of the selected supervised detection methods for synthetic and labeled technical faults as well as unusual consumption with only moderate additional computational cost. We also show that this benefit is mostly observable regardless of the share of anomalies in the considered time series.
- For unsupervised anomaly detection methods, we demonstrate that the proposed approach improves anomaly detection methods for labeled technical faults and partially for synthetic technical faults, but has difficulties with unusual consumption.

Given the proposed approach for enhancing anomaly detection methods, future work could address several follow-up questions. One follow-up question could, for example, address how the proposed approach performs with multivariate time series. Moreover, it could be interesting to integrate the creation of the latent space data representation with the training of the detection methods, and systematically evaluate hyperparameter optimization in the latent space. Furthermore, future work could extend the proposed approach to precisely detect anomalies in a sample that contains an anomaly or focus on improving unsupervised methods for detecting unusual consumption in the latent space data representation.

5 Compensating Anomalies in Energy Time Series

As established in the introduction, anomaly management aims to compensate anomalies once they are detected in an energy time series to obtain a time series with compensated anomalies that better reflects the actual normal behavior. As a first step, the detected anomalies are often labeled as missing values (Akouemo and Povinelli 2017; Alquthami et al. 2020). Although some applications are able to handle data with missing values (e.g., Taylor and Letham 2018), most applications such as load analysis, load forecasting, and load management require that these missing values be further handled.

A common method to handle missing values is imputation. Imputation replaces missing values with values that should resemble the actual data (Moritz and Bartz-Beielstein 2017). Since missing values are a common problem in real-world data sets, many imputation methods exist for time series in general: These methods range from very basic methods such as linear interpolation and the last observation carried forward (Moritz and Bartz-Beielstein 2017) over time series analysis-based methods (Akouemo and Povinelli 2014; Akouemo and Povinelli 2017) to learning-based methods (Bokde et al. 2018; Cao et al. 2018).

To further improve the imputation, it is a common approach to focus on time series from a particular domain and to consider their characteristics as additional information. In the context of smart meters in the smart grid, the recorded energy time series are typically influenced by factors such as weather, human routines, social norms (e. g., weekends or holidays) and many others (González Ordiano et al. 2018; Peppanen et al. 2016). These factors often lead to the commonly known characteristic of daily, weekly, and seasonal patterns, which can be used by imputation methods. For example, daily and weekly patterns are exploited in Friese et al. (2013). The pattern frequency of a power time series P is estimated with the auto-correlation function and the mean values of the estimated pattern frequency are used to impute missing values. Another example is the use of the similarity between days to fill larger gaps in a power time series P with the

Parts of this chapter are reproduced from

M. Weber, M. Turowski, H. K. Çakmak, R. Mikut, U. Kühnapfel, and V. Hagenmeyer (2021). "Data-Driven Copy-Paste Imputation for Energy Time Series". In: *IEEE Transactions on Smart Grid*, Vol. 12, No. 6, pp. 5409–5419. DOI: 10.1109/TSG.2021.3101831.

average values of validated reference days in Matheson et al. (2004). Very short gaps with a length of two hours or less are imputed by linear interpolation, as this often fits the very short-term characteristics of smart meter time series. Lastly, the optimally weighted average approach in Peppanen et al. (2016) also uses daily and weekly patterns as well as seasonality to select appropriate historical values. With these values, the historical averages of a power time series P are calculated, before they are combined with linear interpolation for smooth transitions between actual and imputed values.

Other methods use even more additional data or information to impute missing values in energy time series. An example is the method for imputation, de-noising, and outlier removal based on principal component pursuit in Mateos and Giannakis (2013). It uses the spatial correlations in the power load profiles of adjacent substations. In Ang et al. (2020), the energy time series measured by smart meters in a factory are used to impute missing values in other energy time series from smart meters located in the same factory with clustering and k-nearest neighbors. In Borges et al. (2020), the imputation of substation data is formulated as a forecasting problem. The forecast uses the collected power data of nearby substations as well as weather data, which often has an impact on power consumption and generation.

While all of these imputation approaches are specifically designed for energy time series, all of them except Ang et al. (2020) are limited to the imputation of a power time series P and none of the approaches uses the inherent properties of an energy time series E. Every entry e_t in an energy time series E contains the energy that has been consumed or generated up to time step t. Therefore, – unlike in a power time series P – if, for example, the entries e_t to e_{t+3} are missing in an energy time series E, the next existing entry e_{t+4} still contains the information about the total energy, which was consumed or produced between the time steps $t-1$ and $t+4$. As a consequence, a power time series P can be derived from an energy time series E with missing values but not vice versa.

Thus, in the present chapter, we propose the novel Copy-Paste Imputation (CPI) method for univariate energy time series. It uses an energy time series E as input and copies blocks of data with similar characteristics into existing gaps. By copying blocks of matching data, the inherent patterns of the time series are preserved, even in time series with pattern changes. The CPI method also uses the information about the total energy of each gap that an energy time series E contains in contrast to a power time series P, which guarantees that the total recorded energy remains unchanged during the imputation. To the best of our knowledge, the CPI method is the first method using this property of an energy time series E for imputation. Its realistic imputation that considers the total energy results in a complete power time series P, which also allows calculating a complete energy time series E.

To evaluate the proposed CPI method, we compare its performance with benchmark methods. The comparison comprises the use of matching patterns, the conservation of energy, the computational cost and its decomposition, and the relation of the use of matching patterns and the computational cost. Lastly, we present exemplary imputations

to visually illustrate the evaluation result and show an example of how the CPI method imputes data with detected anomalies considered as missing values.

With the proposed CPI method, we answer research question [RQ3] presented in Section 1.1 that addresses how anomalies detected in energy time series be compensated. By answering research question [RQ3], the proposed CPI method realistically compensates detected anomalies by imputation, and the resulting imputed time series serves as a solid foundation for the subsequent forecast method in the pipeline for managing anomalies (see Figure 5.1).

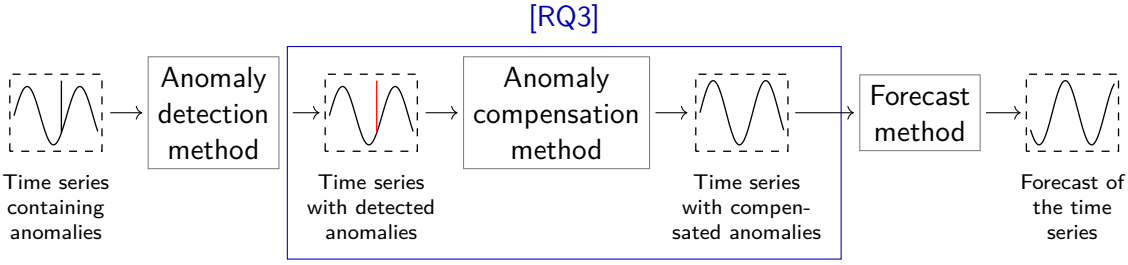


Figure 5.1 By answering research question [RQ3], the proposed CPI method realistically compensates detected anomalies by imputation, and the resulting imputed time series serves as a solid foundation for the subsequent forecast method in the pipeline for managing anomalies.

The remainder of the present chapter is structured as follows. Section 5.1 introduces the proposed CPI method in detail before Section 5.2 describes the experimental setting of the performed evaluation. Section 5.3 presents the results of the evaluation of the CPI method against three benchmark methods on real-world data sets and its exemplary imputations. Section 5.4 discusses the results and the proposed method before Section 5.5 gives concluding remarks.

5.1 Anomaly Compensation Using the Copy-Paste Imputation Method

In this section, we present the proposed CPI method to compensate detected anomalies.[1] For the compensation, the detected anomalies are regarded as missing values, so the application of an imputation method is reasonable. As shown in Figure 5.2, the CPI method comprises several steps. It uses an energy time series E with gaps, i. e., one or multiple consecutive missing values, as input and first imputes single missing values using a linear interpolation. The resulting energy time series with imputed single values E serves as the basis for estimating the energy consumption per day.[2] Next, a list of

1 A Python implementation of the CPI method is publicly available at https://github.com/KIT-IAI/CopyPasteImputation.

2 Although we refer to consumption data only, the same principles apply to generation data.

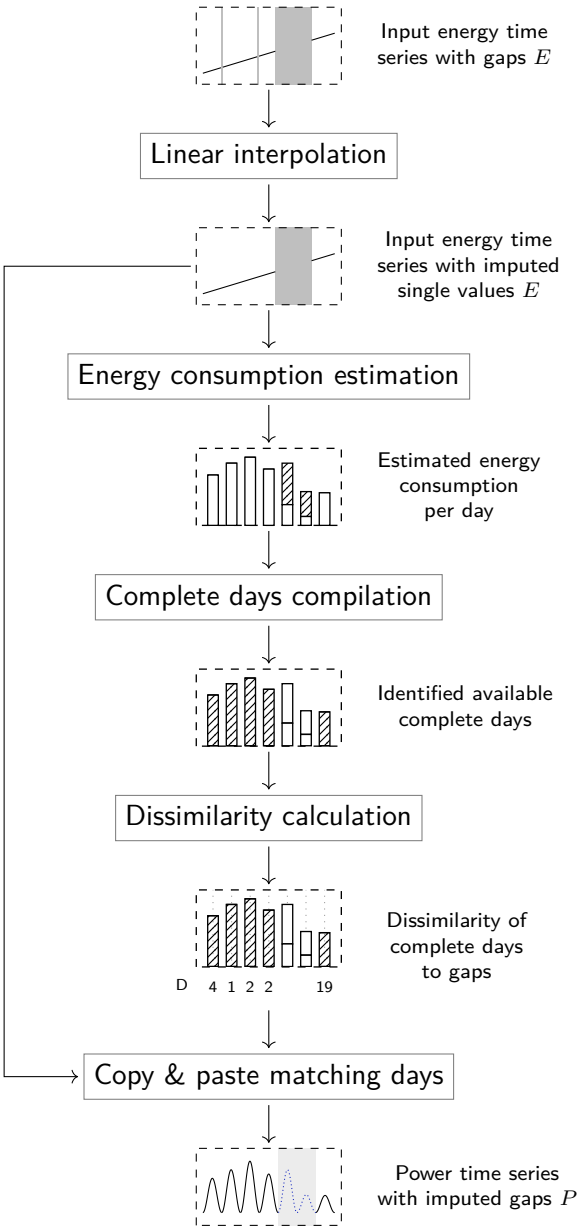


Figure 5.2 The CPI method uses an energy time series E with gaps as input. After imputing single missing values with a linear interpolation, it estimates the energy consumption per day and compiles the available complete days. For these days, the CPI method then calculates their dissimilarity to the days with gaps. Given the dissimilarity of the complete days, it finally copies matching days and pastes them into days with gaps in the power time series P that is derived from the input energy time series with imputed single values E. The CPI method finally provides an imputed power time series P.

the available complete days is compiled, before the dissimilarity between these days and the days with gaps is determined. Based on the dissimilarity, the CPI method finally imputes the missing values of the days with gaps. For this, it fills the days with gaps with the best matching days of the same time series in the power time series P that is

derived from the input energy time series with imputed single values E. The result is an imputed power time series P. In the following, we describe each of these step of the CPI method in detail and reference the corresponding lines in Algorithm 5.1.

Algorithm 5.1 Copy-Paste Imputation (CPI)

Input: energy time series E with missing values (i. e., NaNs)
Result: energy time series E without missing values, optionally power time series without missing values P

1 $E \leftarrow$ single_value_linear_interpolation(E)
2 $energy\_per\_day \leftarrow$ calculate_energy_per_day(E)
3 $non\_complete\_days \leftarrow$ determine_days_with_missing_values(E)
`// each entry in this list describes whether a day of` E `has missing values`
4 $weekly\_pattern \leftarrow$ estimate_weekly_pattern_with_prophet($energy\_per\_day, non\_complete\_days$)
`// only considers the daily energy consumption of the days without missing values`
5 $missing\_energy\_per\_day \leftarrow$ estimate_missing_energy_per_day($E, weekly\_pattern$)
6 $estimated\_energy\_per\_day \leftarrow energy\_per\_day + missing\_energy\_per\_day$
7 $complete\_days \leftarrow$ compile_list_of_complete_days($E.time, energy\_per\_day, non\_complete\_days$)
8 $P \leftarrow$ derive_power_time_series_from_energy_time_series(E)
9 **foreach** *day with missing values* **do**
10 $best\_matching\_day \leftarrow$ find_day_with_min_dissimilarity($day, complete\_days$)
11 $P[day] \leftarrow P[best\_matching\_day]$
12 **end**
13 **foreach** *gap in E* **do**
14 $scaling\_factor \leftarrow actual\_energy\_of\_gap \,/\, imputed\_energy\_of\_gap$
15 $P[gap] \leftarrow P[gap] \cdot scaling\_factor$
16 **end**
17 $E \leftarrow$ calculate_energy_time_series_from_power_time_series(P)

5.1.1 Linear Interpolation of Single Missing Values

In the first step of the CPI method, single missing values, i. e., individual meter readings, are imputed in the given energy time series E (see Line 1 in Algorithm 5.1). For this imputation, we use a linear interpolation because it provides sufficiently correct estimates for individual missing values. We consider only single missing values in this step to limit the number of consecutively linearly interpolated values and thus potentially unrealistic imputations, while still benefiting from these easily imputable values. Indeed, the resulting imputed values are considered as correct in the subsequent steps to increase the number of days without missing values that are available for copying.

5.1.2 Energy Consumption Estimation

In the second step, the CPI method estimates the energy consumption for days with gaps (see Lines 2 to 6 in Algorithm 5.1). The total energy consumption during gaps can be determined because the CPI method uses an energy time series as input. In energy time series, the first entry after a gap still contains the information about the total energy consumed during the gap. For this reason, to obtain the total energy consumption E_j of the gap j from time step t to time step $t + k$, we calculate the energy difference, i. e.,

$$E_j = e_{t+k+1} - e_{t-1}, \tag{5.1}$$

where e_{t+k+1} and e_{t-1} are the energy consumption at the time steps $t + k + 1$ and $t - 1$ respectively. This energy difference is the basis for the estimation of the missing energy in Line 5 in Algorithm 5.1.

However, for gaps longer than one day, the calculated energy consumption must be allocated to the respective days appropriately. For this purpose, the calculated energy consumption of the gap is first distributed to the respective days according to their share of missing values. Second, we consider a weekly pattern in the daily energy consumption. For this pattern, we use the weekly pattern of the input energy time series estimated by the Prophet method (Taylor and Letham 2018). It models the weekly pattern such that the values of all weekdays add up to zero. If some of these values are positive, others need to be negative. When estimating these values, the Prophet method only considers the daily energy consumption of the days without missing values, i. e., one value per day. Lastly, the estimated weekly pattern is added to all days of the gap. When adding the weekly pattern, the added energy is also summed up. The sum of the added energy is then divided by the number of days in the gap to obtain the average added energy. This average is subtracted from each day of the gap to preserve the total energy consumption of the gap.

5.1.3 Compilation of Available Complete Days

In the third step, the CPI method compiles a list of the available complete days, i. e., days without missing values (see Line 7 in Algorithm 5.1). Assuming daily, weekly, and seasonal patterns in the energy consumption, each day is listed with the following three characteristics: its total energy consumption (d_e), its weekday ($d_w \in \{1, 2, \ldots, 7\}$), and its seasonal position (d_s). Under the assumption of a yearly seasonality, i. e., 365 days or 366 days for leap years, it follows that d_s is in $\{1, 2, \ldots, 366\}$.

5.1.4 Calculation of Dissimilarity Between Days

In the fourth step, the CPI method calculates a dissimilarity criterion between each day with gaps and all complete days (see Line 10 in Algorithm 5.1), which is used to select the best matching days for filling gaps in the next step. For the dissimilarity criterion,

the CPI method uses the three previously introduced characteristics of days, namely the total energy, the weekday, and the seasonal position. Since these characteristics are already computed for all complete days, they only have to be determined for the days with missing values in this step. More precisely, the three distance measures D_e, D_w, and D_s are calculated for each day with gaps d_i and each available complete day d_j. We introduce each distance measure in the following.

The first distance measure D_e describes the distance between the total energy consumption of a day with gaps d_i and a complete day d_j. The total energy consumption can serve as a distance measure because the CPI method uses an energy time series as input and thus can calculate the energy consumed during a gap. The distance measure D_e is defined as

$$D_e(d_i, d_j) = \frac{|d_{i,e} - d_{j,e}|}{e_{max} - e_{min}}, \tag{5.2}$$

where e_{max} and e_{min} are the maximum and minimum energy consumption of a day in the time series and $d_{i,e}$ and $d_{j,e}$ are the total energy consumption of the days d_i and d_j. For the day with gaps d_i, the energy consumption estimated in the second step is used. To ensure that the distance measure D_e is in $[0, 1]$, it is divided by the difference between e_{max} and e_{min}.

The second distance measure D_w is based on the assumption of a weekly pattern in the time series and describes the distance between the weekday of a day with gaps d_i and a complete day d_j. It is defined as

$$D_w(d_i, d_j) = \begin{cases} 0.0, & \text{if } d_{i,w} = d_{j,w} \\ 0.5, & \text{if } d_{i,w} \in \{1, \ldots, 5\} \wedge d_{j,w} \in \{1, \ldots, 5\} \\ & \vee d_{i,w} \in \{6, 7\} \wedge d_{j,w} \in \{6, 7\} \\ 1.0, & \text{else,} \end{cases} \tag{5.3}$$

where $d_{i,w}$ and $d_{j,w}$ are integer representations of the weekday of the days d_i and d_j. The integers 1 to 5 represent the workdays Monday to Friday, whereas 6 and 7 represent the weekend days Saturday and Sunday. This distance measure D_w assigns smaller distances to days of the same weekday or days of the same class (i. e., workday or weekend) and higher distances to days of different weekdays or classes.

The third distance measure D_s captures the underlying seasonal pattern and describes the distance between the seasonal position of a day with gaps d_i and a complete day d_j. It is defined as

$$D_s(d_i, d_j) = \begin{cases} \frac{|d_{i,s} - d_{j,s}|}{\lfloor \frac{\mathbf{s}}{2} \rfloor}, & \text{if } |d_{i,y} - d_{j,y}| \leqslant \lfloor \frac{\mathbf{s}}{2} \rfloor \\ \frac{\mathbf{s} - |d_{i,s} - d_{j,s}|}{\lfloor \frac{\mathbf{s}}{2} \rfloor}, & \text{else,} \end{cases} \tag{5.4}$$

where $\mathbf{s}$ is the length of the seasonal cycle and $d_{i,s}$ and $d_{j,s}$ are the positions of the days d_i and d_j in this cycle. For a yearly seasonality, $\mathbf{s}$ can be set to 365 or 366 to reflect the number of days in a year. This distance measure ensures that two days from the same season are considered as more similar than two days from different

seasons. For example, January 1 and December 31 of the same year are almost one year apart but have a minimal distance D_s. In contrast, January 1 and July 1 are only half a year apart and have a maximal distance D_s.

To determine the dissimilarity between a day with gaps d_i and a complete day d_j, the three individual distance measures D_e, D_w, and D_s are combined as a weighted sum into a single dissimilarity criterion D. It is defined as

$$D = w_e D_e + w_w D_w + w_s D_s, \tag{5.5}$$

where w_e, w_w, and w_s are the weights and D_e, D_w, and D_s are the normalized distance measures. The individual distance measures are normalized to the interval [0,1] for an easier interpretation of the used weights. It is necessary to specify these weights once before applying the CPI method. To find suitable weights, one possible approach is to perform a grid search on a representative set of time series (see Section 5.2 for an exemplary grid search).

5.1.5 Copy and Paste of Matching Days

In the last step, the CPI method copies the best matching days, pastes them into the gaps (see Line 11 in Algorithm 5.1) of the derived power time series P (see Line 8 in Algorithm 5.1), and scales the imputed values to preserve the energy of the respective gaps (see Lines 13 to 16 in Algorithm 5.1).

To determine the best matching days, the CPI method uses the previously generated list of complete days. For a day with gaps d_i, it selects the day d_j with the smallest dissimilarity $D(d_i, d_j)$. Since the entire list of complete days is used in this step, days from the future of the day with gaps are also considered.

Given the selected best matching days, the CPI method performs the actual copying and pasting of the best matching days into the gaps. For this, the power time series P serves as a basis. As described in Section 2.1, it can be derived from the input energy time series E by calculating the average power p_t between the time steps $t-1$ and t, i. e.,

$$p_t = \frac{e_t - e_{t-1}}{\Delta t}, \tag{5.6}$$

where Δt is the time between the two time steps, e_t and p_t are the energy and power at time step t, and e_{t-1} is the energy at time step $t-1$. In the derived power time series P, the CPI method replaces every missing value in each day with gaps by the corresponding value of the previously determined best matching complete day.

Finally, the CPI method scales the imputed power values to preserve the actual energy of each gap. The scaling is based on the actual energy and the imputed energy. Both can be determined because the CPI method uses energy time series as input and thus can calculate the energy consumed during a gap. The actual energy E_j of the gap j is calculated according to Equation (5.1). The imputed energy E'_j is calculated by accumulating the

imputed power values. To preserve the energy, the imputed power values of gap j are multiplied with the ratio of the actual energy and the imputed energy of gap j, i. e.,

$$\hat{p}_t = \hat{p}'_t \cdot \frac{E_j}{E'_j}, \tag{5.7}$$

where $\hat{p}'_t$ is the power value calculated by the CPI method and $\hat{p}_t$ is the scaled power value.

After this scaling, one can use the imputed power time series P to calculate the energy values e_t of the corresponding imputed energy time series E. As described in Section 2.1, solving Equation (5.6) for e_t yields

$$e_t = \Delta t \cdot \sum_{i=1}^{t} p_i + k, \tag{5.8}$$

where k is a constant representing the offset of the energy time series E. With the calculated energy values, the CPI method finally returns a complete energy time series E whose initially missing values are completely imputed and for which the total energy of each gap remains unchanged.

5.2 Experimental Setting

In this section, we present how we evaluate the proposed CPI method. After describing the used data, we introduce the selected benchmark methods and the applied evaluation criteria. Finally, we determine the weights used in the CPI dissimilarity criterion and present the used hard- and software.

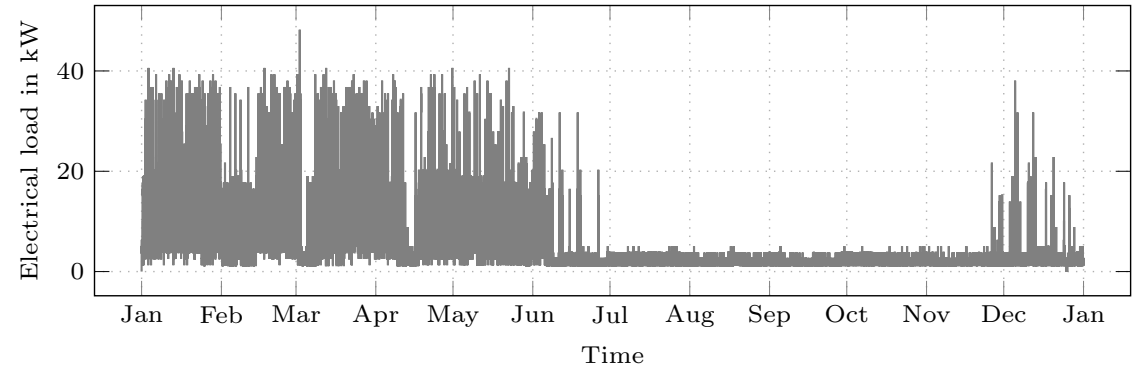
5.2.1 Data Sets With Missing Values

For the evaluation, we use two data sets. Both comprise real-world data but differ in the observed missing values. While we insert missing values – that can represent previously detected anomalies – into the first, the second already contains labeled anomalies that we regard as missing values in the evaluation.

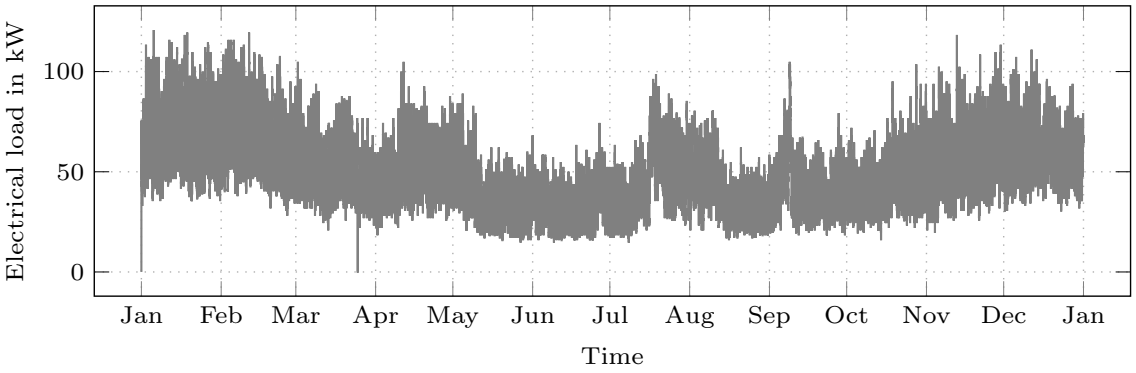
Data With Inserted Missing Values The first used data set is the "ElectricityLoadDiagrams20112014 Data Set"[3] from the UCI Machine Learning Repository (Dua and Graff 2019) that we also use in Chapter 4. The data set consists of electrical power time series from 370 clients with different consumption behaviors over a period of up to four years (Rodrigues and Trindade 2018) The time series contain quarter-hourly average power values in kW, resulting in 35,040 values per year. From these 370 time

3 https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

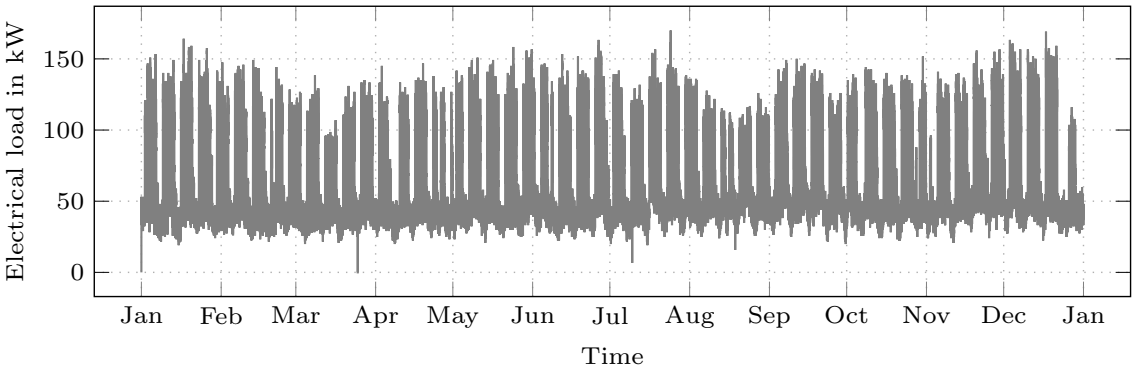
series, we select 50 time series as a representative sample (see Table C.1 in Appendix C) and use their 2012 values for the evaluation. The selected time series vary greatly in terms of seasonal, weekly, and daily patterns as illustrated in Figure 5.3. For the evaluation of the CPI method, the selected power time series P_i, that do not contain any missing values, are converted to energy time series E_i by integrating the power values in a time-discrete manner using Equation (5.8).



(a) Power time series *MT_001*.



(b) Power time series *MT_005*.



(c) Power time series *MT_013*.

Figure 5.3 Three exemplary power time series P in 2012 from the first selected data set with different daily, weekly, and seasonal patterns.

Due to their completeness, we insert missing values in the calculated energy time series E_i by replacing values with NaNs. To decide which values are replaced, we perform the

following four steps. First, we determine the longest sequence of the energy time series E_i without missing values T_c. Second, we define the number of consecutive energy values to be replaced in this sequence T_c by choosing uniformly between two and the minimum of the specified maximum number of consecutive missing values, the length of T_c, and the remaining number of values to be replaced. Third, we randomly select a starting index for the determined number of consecutive energy values to be replaced such that all values to be replaced are contained in T_c. Lastly, we replace each selected energy value with NaN. We repeat these four steps until we reach the total number of values to be replaced.

For the evaluation, we consider the number of values to be replaced in the form of shares of missing values. In the evaluation, we use six shares of missing values between 1 % and 30 %, namely 1, 2, 5, 10, 20, and 30 %. To consider both larger gaps and single missing values, 5 % of each share of missing values are single missing values. We randomly determine the indices for the single missing values after creating the larger gaps.

Data With Detected Anomalies Considered as Missing Values The second used data set is the electrical data collected on the Campus North of the Karlsruhe Institute of Technology (KIT), which we also use in Chapters 3 and 4 and in Turowski et al. (2022b). From this data, we consider the smart meter of a typical mid-campus office building. More precisely, we choose that smart meter whose power time series P from 2016 is used for the evaluation in Chapter 4. Thanks to its availability, we use the related one-year energy time series E (see Figure 5.4). It contains 19 labeled anomalies of the four identified anomaly types from the group of technical faults described in Chapter 3. The contained anomalies correspond to a 6 % share of the data. Figure 5.5 shows an exemplary labeled anomaly of all four types. For more details on the contained anomalies, a plot of the entire related power time series P, and a plot of the shown exemplary anomalies in the related power time series P, we refer to Table 4.1 and Figures 4.7a and 4.8 in Section 4.2.1.

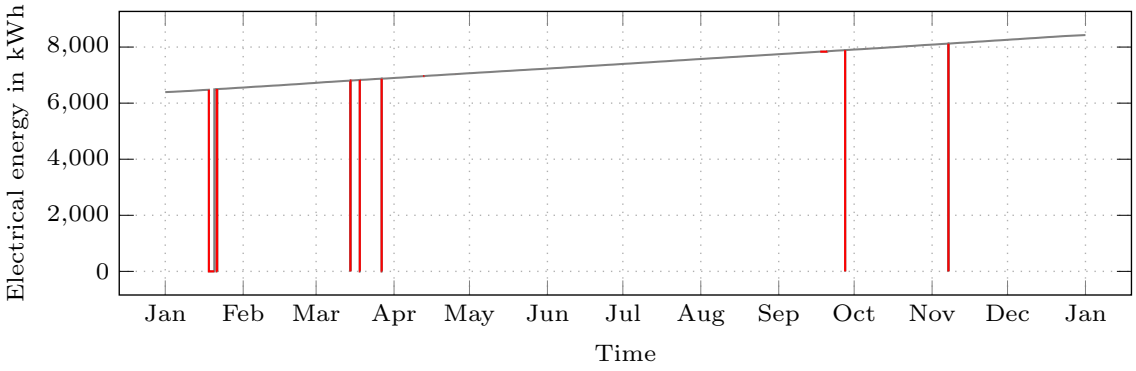
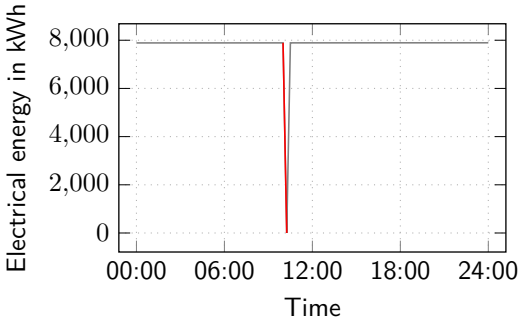
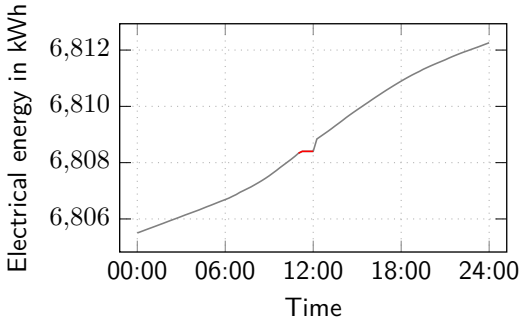


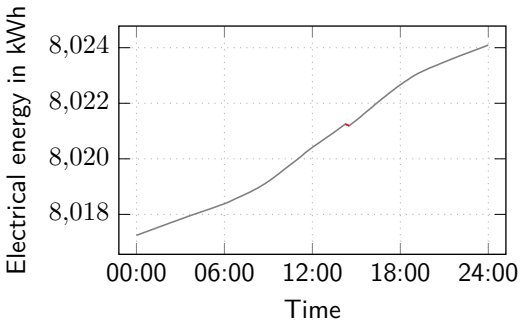
Figure 5.4 The chosen energy time series E from the second selected data set. It contains 19 labeled anomalies of the four anomaly types from the group of technical faults that are considered as missing values. The labeled anomalies are plotted in red. Note that the labeled anomalies with a short length are not recognizable due to their length.



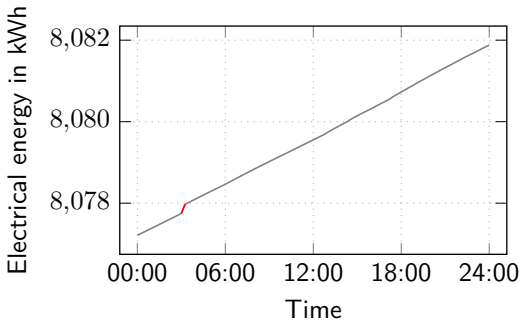
(a) Anomaly of type 1 on 27.09.2016.



(b) Anomaly of type 2 on 15.03.2016.



(c) Anomaly of type 3 on 20.10.2016.



(d) Anomaly of type 4 on 30.10.2016.

Figure 5.5 Examples of the labeled anomalies of types 1 to 4 from the technical faults in the selected energy time series with labeled anomalies E. The anomalies are plotted in red. Note that the anomalies of types 3 and 4 actually have a length of one but are marked together with the following value to be recognizable.

We consider the labeled anomalies in the selected energy time series E as missing values in order to apply the CPI method to the selected time series for the evaluation.[4] This procedure corresponds to the use case where detected anomalies should be replaced by typical patterns and values.

5.2.2 Benchmark Methods

To compare the performance of the proposed CPI method, we apply benchmark methods to the data set. As suitable benchmark methods, we generally consider all imputation methods for an energy time series E that use the time series and its characteristics only. Due to the lack of imputation methods for energy time series E – to the best of our knowledge –, we include imputation methods for power time series P and time series in general although they have the disadvantage of not using energy data. Methods requiring additional data or information such as weather data (Akouemo and Povinelli 2014; Akouemo and Povinelli

4 Note that, as with the first selected data set, one could alternatively use the related power time series P as the starting point.

2017) or validated reference days (Matheson et al. 2004) and methods designed for multivariate time series only (Borges et al. 2020; Cao et al. 2018; Mateos and Giannakis 2013) are discarded due to lack of comparability. Furthermore, during the evaluation, the method described in Bokde et al. (2018) is excluded due to its excessive run-time.

As a result, we use three methods as benchmarks in view of comparison complexity and fairness. We derive these methods from literature (Friese et al. 2013; Moritz and Bartz-Beielstein 2017; Peppanen et al. 2016; Taylor and Letham 2018) and adapt them where necessary. To establish a fair comparison, the evaluated benchmark methods receive their data input in the same way as the CPI method. The methods sequentially get the 50 time series and can use each individual time series completely, but independently from the others.

The first benchmark method is the commonly applied Linear Interpolation (Moritz and Bartz-Beielstein 2017; Peppanen et al. 2016). This method represents a lower baseline and should be outperformed in any case. It imputes missing values $\hat{p}_t$ by linearly interpolating the last known power value before a gap and the first known power value after the gap, i. e.,

$$\hat{p}_t = \frac{t - t_1}{t_2 - t_1} \cdot (p_{t_2} - p_{t_1}) + p_{t_1}, \tag{5.9}$$

where t_1 and t_2 are the time steps before and after the gap. The Linear Interpolation is thus the only evaluated method that uses two values for imputing a gap.

The second benchmark method is the Optimally Weighted Average (OWA) (Peppanen et al. 2016). Assuming a weekly pattern, this method calculates a historical average

$$\hat{p}_t^{\mathrm{HA}} = \frac{1}{|H|} \sum_{i \in H} p_i, \tag{5.10}$$

where H contains all values of the hour before and after t as well as of the same two hours of the previous and of the next week. As long as H is empty, the considered weeks are iteratively extended by one in each direction to include additional values from the same two hours in further weeks. To ensure smooth transitions between actual and imputed values, this average is combined with the linear interpolation $\hat{p}_t^{\mathrm{LI}}$ (see 5.9). The combination results in

$$\hat{p}_t = w_t \hat{p}_t^{\mathrm{LI}} + (1 - w_t) \hat{p}_t^{\mathrm{HA}}, \tag{5.11}$$

where w_t weighs the influence of the two imputation methods. The weight w_t is designed to decrease with increasing distance to the actual values, i. e.,

$$w_t = e^{-\alpha d_t}, \tag{5.12}$$

where d_t describes the distance from t to the nearest actual value in the time steps and α determines the rate of decay for w_t. Since α has a negligible influence on the imputation results in the present evaluation, we use a global $\alpha = 0.1387$ for the evaluation as determined in Peppanen et al. (2016).

The third benchmark method is based on the Prophet method for time series forecasting (Taylor and Letham 2018). The Prophet method uses a modular regression model that can be described as

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{5.13}$$

where g is a model for the trend, s for the seasonality, h for holidays, and ϵ_t for changes that are not represented in the model. The imputation method based on this model exploits Prophet's capability to estimate a time series model on irregularly spaced data (Taylor and Letham 2018) and imputes missing values with the corresponding values of the model. The model is learned on all values available in the energy time series E to be imputed. In contrast to its application in the CPI method, the Prophet-based benchmark imputation method gets, like all other benchmark methods, the aforementioned quarter-hourly values as input, i. e., 96 values per day.

5.2.3 Evaluation Criteria

To evaluate the imputation in a power time series P using the CPI method and the benchmark methods, we use three evaluation criteria.

The first evaluation criterion is the use of matching patterns to fill the gaps. More precisely, we determine how well imputed patterns match the actual patterns. For this purpose, we measure the deviation between every single actual power value and the corresponding imputed power value using the Mean Absolute Percentage Error (MAPE). It is defined as

$$\text{MAPE} = \frac{1}{|T_m|} \sum_{t \in T_m} \left| \frac{\hat{p}_t - p_t}{p_t} \right|, \tag{5.14}$$

where p_t and $\hat{p}_t$ are the actual and imputed power values at time step t and T_m is the set of time steps with missing values.

The second evaluation criterion is the conservation of the total energy in the gaps. For this, we measure the difference between the actual and imputed energy while ignoring the fine granular patterns that are used for the imputation. We determine this difference using the Weighted Absolute Percentage Error (WAPE), which is defined as

$$\text{WAPE} = \frac{\sum_{i=1}^{N} |\hat{E}_i - E_i|}{\sum_{i=1}^{N} E_i}, \tag{5.15}$$

where E_i and $\hat{E}_i$ are the actual and imputed energy of gap i in a power time series P with N gaps. In contrast to the MAPE, the weighting of the individual absolute errors is necessary in the WAPE to account for gaps of different sizes.

The third evaluation criterion is the computational cost of the imputation. For this, we measure the run-time of the evaluated methods and decompose it into model estimation and imputation.

5.2.4 Used Weights in the CPI Dissimilarity Criterion

Before copying and pasting matching days, the CPI method calculates the dissimilarity criterion between two days using Equation (5.5). For this calculation, the CPI method requires that the weights of the three distance measures regarding the total energy consumption, the weekday, and the seasonal position be determined in advance. For the CPI method used in the evaluation, we determine these weights using a grid search before the actual evaluation.

This grid search is performed on a separate data set that is used only for calibration. To compile the calibration data set, we consider the remaining 320 time series of the selected data set in which we insert missing values. Based on a visual inspection, we choose five time series for the calibration data set (see Table C.2 in Appendix C) and use their 2012 values for the calibration. The chosen time series differ from each other but have similar characteristics as the 50 time series selected for the evaluation. Each of the five time series from the calibration data set is evaluated with six different shares of missing values, ranging from 1 % to 30 %, which results in 30 time series in total.

Using this calibration data set, we test 1,000 combinations of the three weights $w = (w_e, w_w, w_s)$ with each weight in the range $[1, 2, \ldots, 10]$. Each combination of weights is evaluated using the two previously introduced evaluation criteria concerning the use of matching patterns and the conservation of the total energy. We calculate the related error measures MAPE and WAPE for all time series of the calibration data set. Based on the results, we also determine the overall minimum and maximum of these two error measures and use them to min-max normalize the values of these two error measures. The min-max normalized MAPE is calculated with

$$\mathrm{MAPE_n}(w, i) = \frac{\mathrm{MAPE}(w, i) - \min \mathrm{MAPE}}{\max \mathrm{MAPE} - \min \mathrm{MAPE}}, \tag{5.16}$$

where w is the tested weight combination and i is the identifier of the time series from the calibration data set. The min-max normalized WAPE is determined analogously to Equation (5.16). To obtain the average sum of these two normalized error measures $\overline{TE}$, we first add the normalized results of both error measures for each time series from the calibration data set. Afterward, we add this sum of all time series and divide it by the number of time series, i. e.,

$$\overline{TE}(w) = \frac{\sum_i^n \mathrm{MAPE_n}(w, i) + \mathrm{WAPE_n}(w, i)}{n}, \tag{5.17}$$

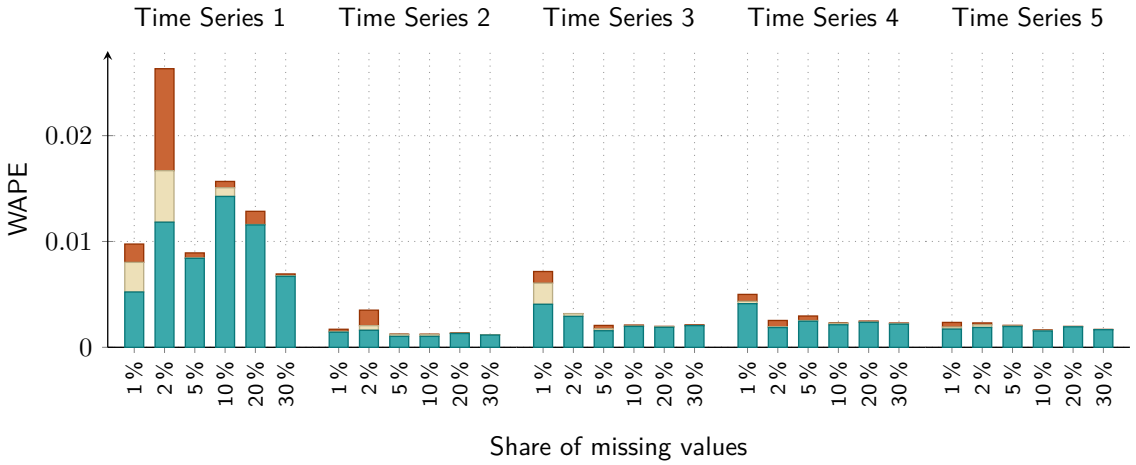
where n is the number of time series in the calibration data set. Finally, we determine the weights with the minimum average sum of both normalized error measures, i. e.,

$$w_{opt} = \arg\min_w \overline{TE}(w). \tag{5.18}$$

We refer to these weights w_{opt} in the following as best mean weights.

To evaluate the aforementioned 1,000 weight combinations, the grid search requires 91 minutes and uses a total system memory of approximately 3.5 GB. Figure 5.6 shows the resulting MAPE and WAPE for the tested time series from the calibration data set. For each time series and each share of missing values, a green bar depicts the result with the best weights, an orange bar the results with the worst weights, and a light yellow bar the result with the determined best mean weights $w_{opt} = (10, 1, 5)$.

(a) MAPE



(b) WAPE

Figure 5.6 The MAPE and WAPE of the best, worst, and overall best weights for the five time series from the calibration data set used in the grid search to determine the weights in the CPI dissimilarity criterion for the evaluation. Every time series is evaluated with six different shares of missing values ranging from 1 % to 30 %, resulting in a total of 30 time series used.

The results show that every time series has its own optimal weight combination. Nevertheless, the difference between the results with the best and the worst weights is often very small. Similarly, the difference between the results with the best weights and the best mean

weights w_{opt} is often negligible. Therefore, w_{opt} is used in the evaluation for all 50 time series from the data, where we insert missing values. We additionally use these weights for the data with detected anomalies that we consider as missing values for simplicity.

5.2.5 Hard- and Software

In the evaluation of the CPI and the benchmark methods, we ensure the comparability of the results by implementing all methods in Python and evaluating them on the same hardware. The evaluation hardware is a desktop PC running Ubuntu 20.04 with an AMD Ryzen 5 3600 processor and 16 GB of memory.

5.3 Results

To evaluate the imputation using the CPI method and the benchmark methods, we first examine the use of matching patterns and the conservation of energy, quantified by the MAPE and the WAPE respectively. We report the truncated means of these two evaluation criteria for the 50 evaluated time series and omit the two best and worst values to obtain less outlier-sensitive results. Afterward, we present the computational cost of the CPI method and the benchmark methods and show its decomposition before we put the use of matching patterns and the computational cost in relation to each other. We then present exemplary imputations to visually illustrate the evaluation results and how the CPI method copies matching days. While all the previous results are based on the data where we insert missing values, we lastly show an example of how the CPI method imputes data with detected anomalies considered as missing values.

5.3.1 Use of Matching Patterns

We evaluate the use of matching patterns by the evaluated methods using the MAPE defined in Equation (5.14). For the six different shares of inserted missing values, Figure 5.7 shows the MAPE of all evaluated methods.

For most of the shares of missing values, we observe that the CPI method performs better than the OWA method as the best benchmark method. Both methods perform overall about 10 to 12 % better than the Prophet-based method. The Linear Interpolation performs by far the worst for all shares of missing values. All methods tend to higher errors with higher shares of missing values. This trend is most distinct for the Linear Interpolation. With regard to the errors of individual time series, the benchmark methods are more prone to extreme errors with a maximum MAPE of 5.88 and above while the CPI method has a maximum MAPE of 2.37.

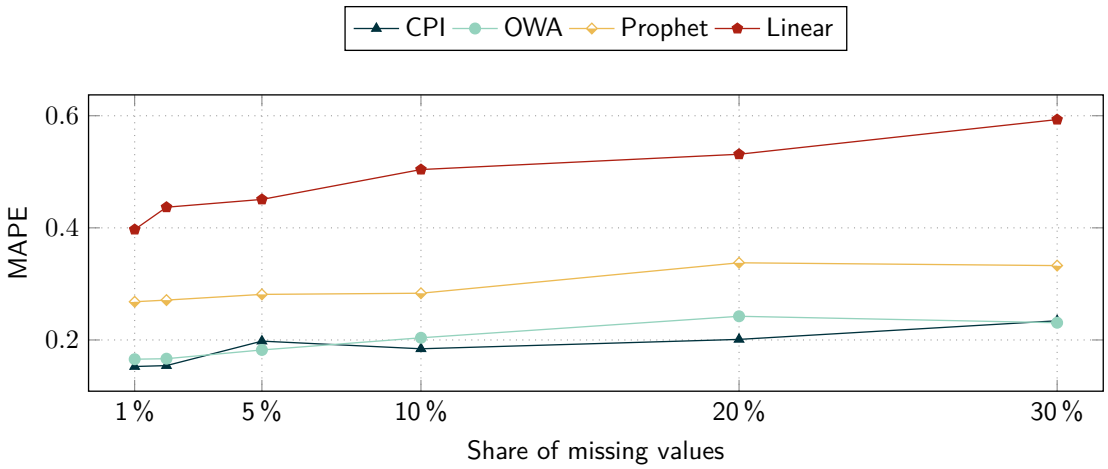


Figure 5.7 The MAPE of the CPI method and the three benchmark methods applied to the data with six different shares of inserted missing values. Since the scaling of imputed values does not noticeably affect the results of the CPI method, it is omitted in this figure.

5.3.2 Conservation of Energy

To evaluate the conservation of energy in the gaps, we use the WAPE as defined in Equation (5.15). Figure 5.8 shows the WAPE of all evaluated methods for the six different shares of inserted missing values. For a better comparability with the benchmark methods that all do not use scaling, the dashed line indicates the WAPE for the CPI method without scaling.

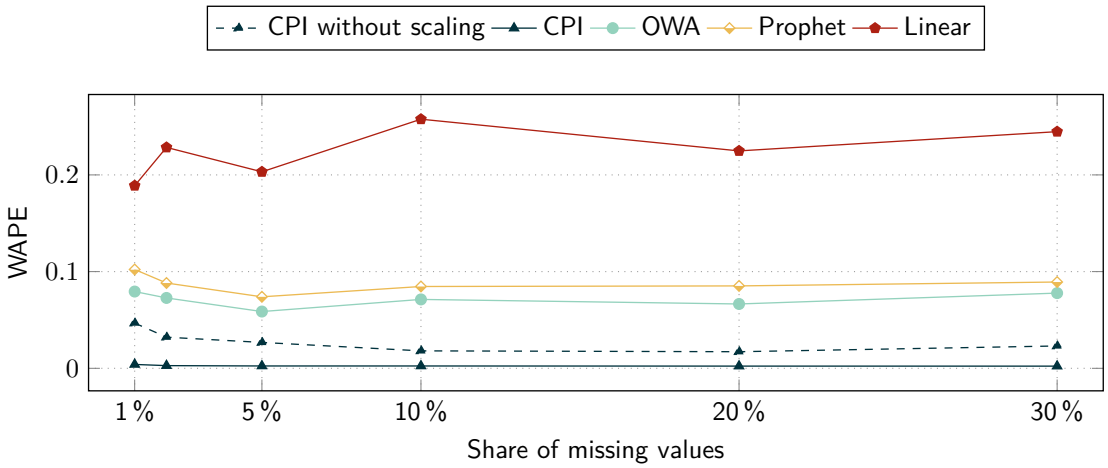


Figure 5.8 The WAPE of the CPI method and the three benchmark methods applied to the data with six different shares of inserted missing values. For better comparability with the benchmark methods that all do not use scaling, the dashed line indicates the WAPE of the CPI method that does not scale the imputed values to preserve the energy of a gap.

We observe that the CPI method performs best regardless of the share of missing values. The CPI method without scaling is the second best method and performs on average 4.4 % better than the OWA method. The Prophet-based method and the OWA method

perform very similarly, with the OWA method performing better by 1.6 % on average. The Linear Interpolation again performs worst for all shares of missing values.

5.3.3 Computational Cost

Regarding the computational cost, we first evaluate the run-times required by the evaluated methods, before we examine how the number of input values influences the run-time of the CPI method.

To evaluate the run-times, we apply the evaluated methods to the 50 selected one-year time series with 35,040 values each. Figure 5.9 shows the resulting average run-times required for the imputation of these 50 time series with different shares of inserted missing values.

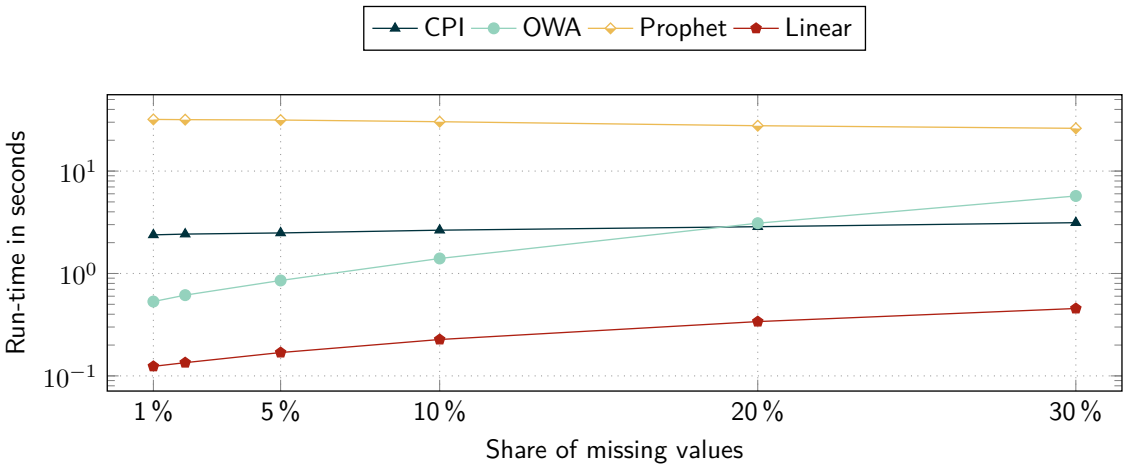


Figure 5.9 The average run-times required by the CPI method and the three benchmark methods for the imputation of the 50 selected one-year time series. Note that the logarithmic time scale visually compresses the run-time decrease of the Prophet-based method by 5.8 seconds from 1 % to 30 % of missing values.

The Linear Interpolation is by far the fastest method. The OWA method is similarly fast for small shares of missing values but increases more drastically in run-time than the other methods for increasing shares of missing values. The CPI method requires about 10 to 20 times more run-time than the Linear Interpolation but is faster than the OWA method for 20 % and 30 % of missing values. The Prophet-based method requires much more time than all other methods and is 9 to 10 times slower than the CPI method.

To examine how the number of input values influences the run-time of the CPI method, we additionally measure the run-time required to impute time series with different lengths. Figure 5.10 shows the run-time required by the CPI method for imputing time series with five different lengths from three months with 8,832 values to three years with 105,120 values. The CPI method scales approximately linearly with the number of input values and has an average run-time of 5.56 seconds for time series with 105,120 values.

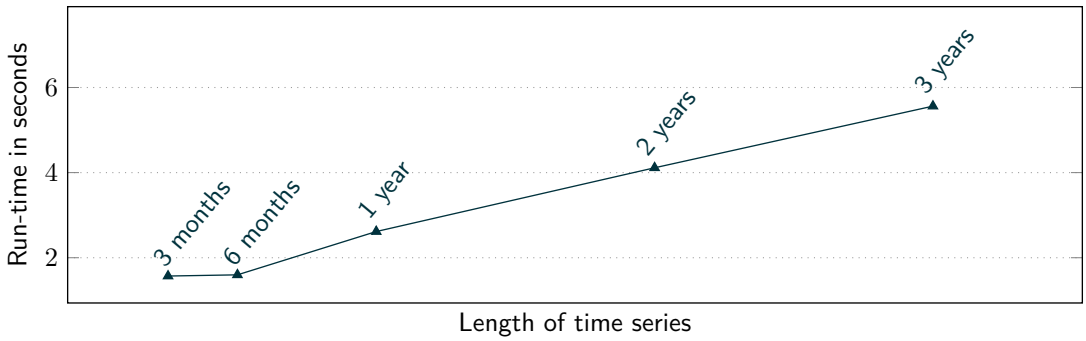


Figure 5.10 The average run-times of the CPI method for time series with five different lengths from three months with 8,832 values to three years with 105,120 values.

5.3.4 Computational Cost Decomposition

To examine the measured run-times in more detail, we first decompose the run-time into the model estimation and the imputation. Afterward, we additionally further differentiate the run-time of the CPI method with respect to its steps.

In Figure 5.11, the run-times of the evaluated methods are decomposed into model estimation and imputation for 1 % and 30 % of missing values. The model estimation including training and fitting is depicted in blue and the actual imputation is depicted in orange.

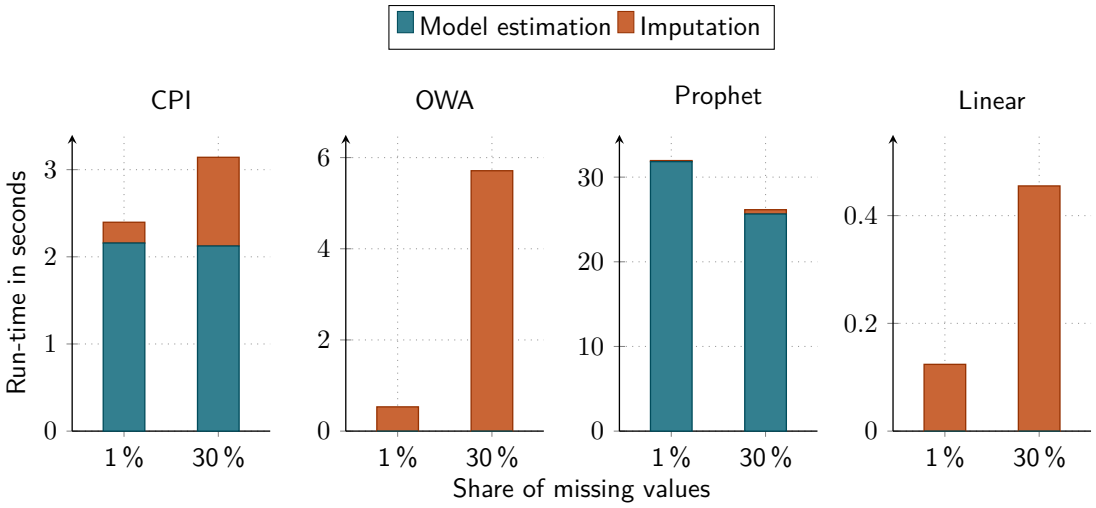


Figure 5.11 Run-time decomposition of the CPI method and the three benchmark methods into model estimation including training and fitting as well as imputation for 1 % and 30 % of missing values.

We observe that the model estimation is the dominant part of the run-time of the CPI method. Similarly, the model estimation also dominates the run-time of the Prophet-based method but decreases with a larger share of missing values. In contrast, the Linear Interpolation and the OWA method do not comprise any model estimation. Their run-times thus entirely consist of the imputation itself. For the Linear Interpolation and the OWA

method, the run-time required for the imputation increases considerably with a larger share of missing values. For the CPI method and the Prophet-based method, the run-time needed for the imputation also increases slightly with a larger share of missing values.

To further differentiate the run-time of the CPI method with respect to its steps, we divide the model estimation into the Linear Interpolation of single missing values, the energy consumption estimation, and the compilation of available complete days. We accordingly split the imputation into the matching of the most similar days, pasting the values into the gaps, scaling the imputed values, and calculating the completed energy time series. For 1 % and 30 % of inserted missing values, Figure 5.12 shows the run-times of these steps from bottom to top, where the steps related to the model estimation are depicted in blue to green and the steps related to the imputation in yellow to red.

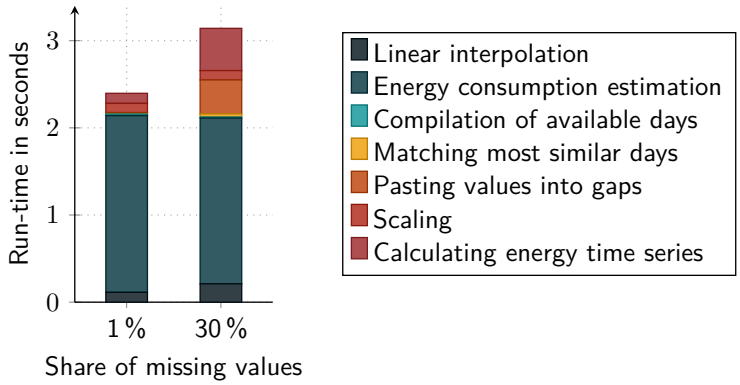


Figure 5.12 Detailed run-time decomposition of the CPI method with regard to its steps for 1 % and 30 % of inserted missing values.

We observe that the steps of the CPI method need different run-times. The energy consumption estimation for gaps, that relies on the Prophet method, requires the largest run-time, whereas all other steps need much less time. Nevertheless, the run-time of the energy consumption estimation slightly decreases with an increased share of missing values. All other steps except for the compilation of available complete days and the scaling, however, require more run-time with an increased share of missing values.

5.3.5 Use of Matching Patterns vs. Computational Cost

To examine the imputation of the evaluated methods, we also relate the use of matching patterns to the computational cost using the previous results from imputation of the 50 considered one-year time series. Figure 5.13 shows the measured average required run-times on the x-axis and the obtained MAPE on the y-axis. A decreasing distance to the origin indicates a better performance of an imputation method.

We observe that the Linear Interpolation provides fast and inaccurate results, whereas the CPI method and the OWA method provide the most accurate results with a reasonable run-time. The Prophet-based method yields medium results while taking much longer to calculate than the other methods.

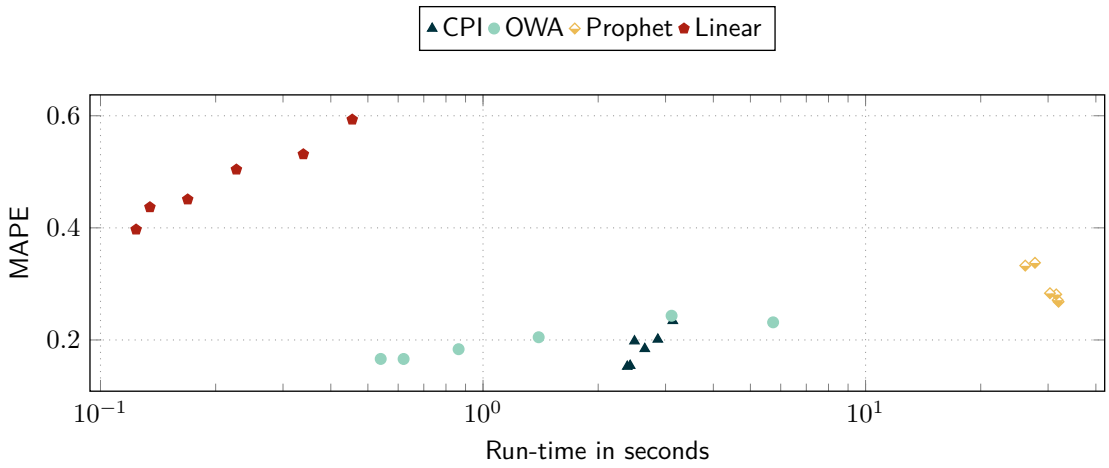


Figure 5.13 Comparison of the use of matching patterns and the computational cost needed by the CPI method and the three benchmark methods for the imputation of 50 one-year time series. The x-axis shows the required average run-times on a logarithmic scale and the y-axis the MAPE.

5.3.6 Exemplary Imputations

To visually illustrate the evaluation results, we first present the imputations of all evaluated methods for one gap, before we show an example of how the CPI method copies and pastes matching days into gaps.

To demonstrate an imputation of all evaluated methods, we choose one of the 50 one-year time series from the first data set with 20 % of inserted missing values, resulting in large gaps. When applying all evaluated methods to the selected time series, we obtain the following results of the CPI method and the best respective benchmark method for the three evaluation criteria: The CPI method has a MAPE of 0.194 and the OWA method one of 0.211. The CPI method achieves a WAPE of 0.003 and the OWA method one of 0.065. The CPI method requires 2.89 s for the imputation and the Linear Interpolation 0.35 s. To illustrate this performance, Figure 5.14 shows the actual values and the imputations of all evaluated methods for a nine-days gap from November 2012 of the selected time series.

We observe that the imputation of the Linear Interpolation fails to capture the patterns of the time series to be imputed. The imputations by the OWA method and the Prophet-based method capture the essential patterns but lack details. The imputation by the CPI method mostly fits the actual values but it shifts and increases some peaks. Despite not explicitly addressing the transitions between existing and imputed values, the CPI method also generally provides smooth transitions on both ends of gaps. Therefore, the imputation by the CPI method comes closest to the actual values compared to the three benchmark methods, which confirms the previously reported MAPE and WAPE.

To show how the CPI method copies and pastes matching days into gaps, we select another time series from the set of 50 one-year time series. Figure 5.15 shows 14 days from January 2012 of the selected time series where the contained one and a half days

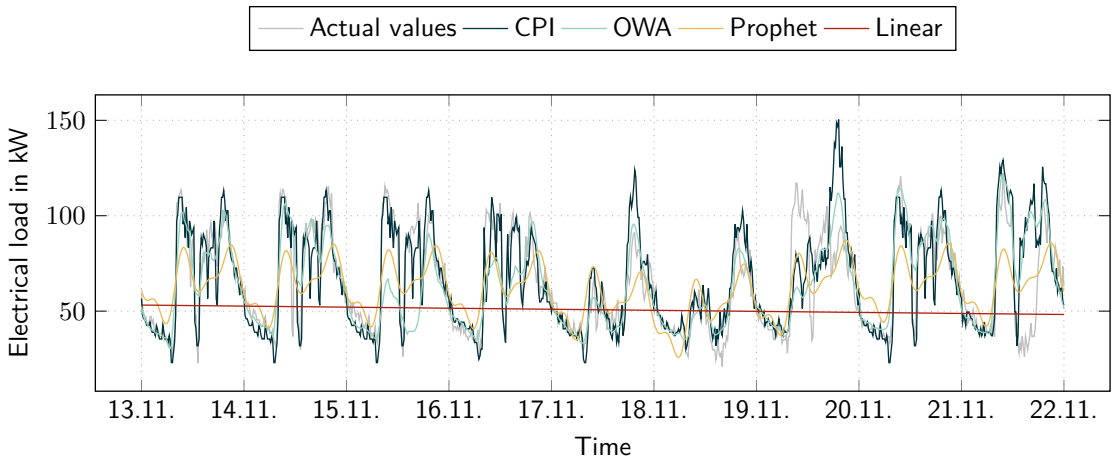


Figure 5.14 The actual values and the imputations by the CPI method and the three benchmark methods for a nine-days gap in November 2012 of an exemplary one-year time series with 20 % of missing values.

gap is imputed using the CPI method. To illustrate the imputation, arrows show which parts of days the CPI method copies and pastes into the gap.

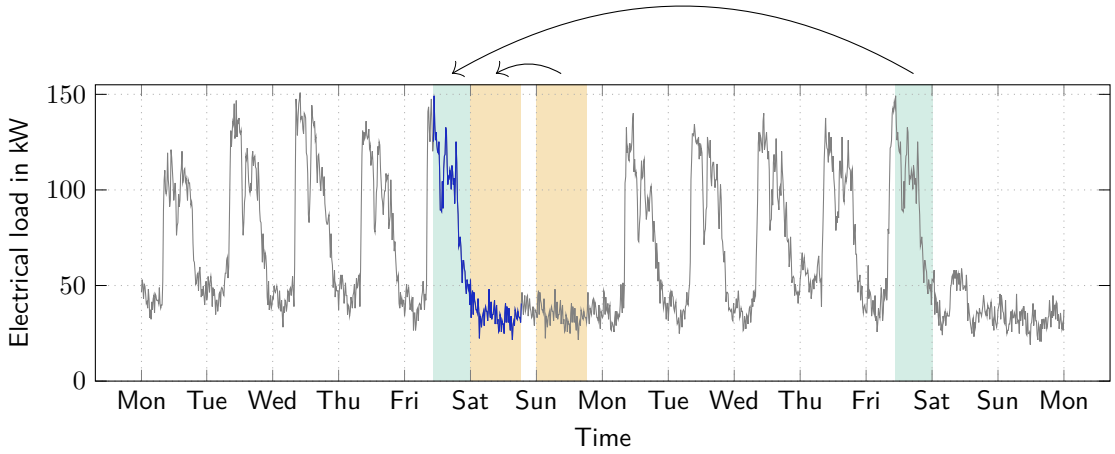


Figure 5.15 A 14-days segment from January 2012 of an exemplary time series whose gap is imputed using the CPI method. To illustrate the imputation, arrows show which parts of which days the CPI method copies and pastes into the gap.

We observe that the gap covers half a Friday and almost all of the following Saturday. The applied CPI method imputes this gap by copying a part of the matching Friday one week later and a part of the matching Sunday following the gap.

5.3.7 Exemplary Imputation of Data With Detected Anomalies

To present an example of how the CPI method imputes data with detected anomalies that are considered as missing values, we apply the CPI method to the previously introduced

energy time series E from the second data set. The time series contains 19 labeled anomalies of the four anomaly types from the group of technical faults. We first show the resulting imputed energy time series E, before presenting the related imputed power time series P and imputed exemplary anomalies contained therein.

Figure 5.16 shows the selected energy time series E with the imputations using the CPI method. We find that the CPI method replaces all zero and constant values of the labeled anomalies, resulting an monotonously increasing energy time series E.

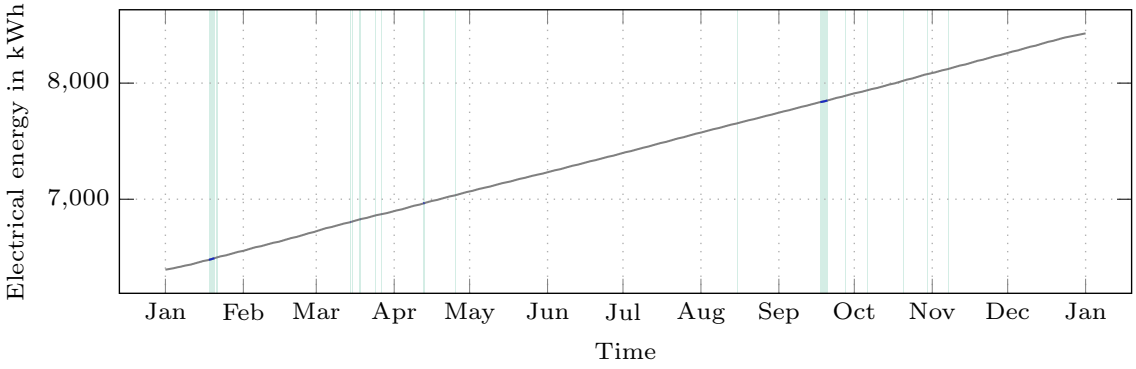


Figure 5.16 The chosen energy time series E from the second selected data set. It contains 19 labeled anomalies of the four anomaly types from the group of technical faults that are considered as missing values when applying the CPI method.

To also examine the patterns used for the imputation, we furthermore consider the power time series P that the CPI provides when imputing the energy time series E selected from the second data set. Figure 5.17 shows this entire power time series P with the imputations by the CPI method.

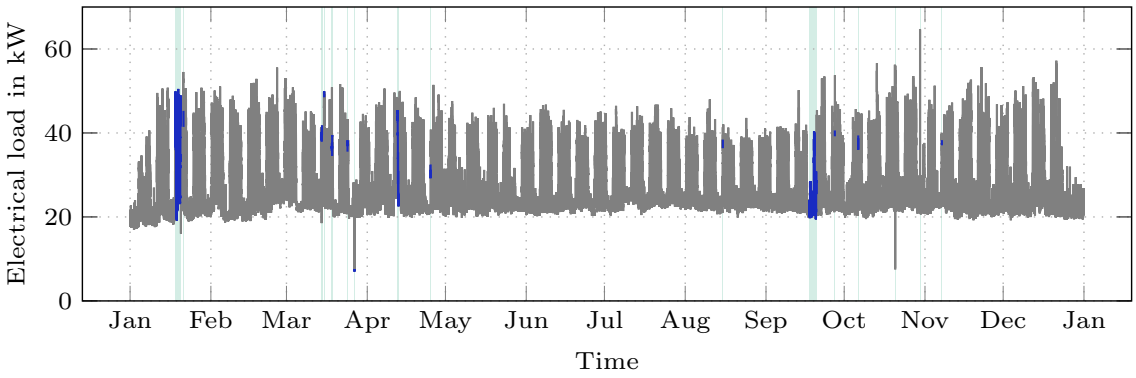
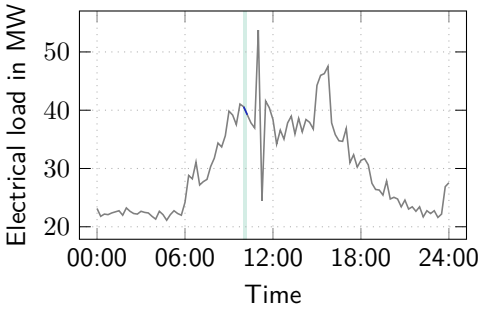


Figure 5.17 The imputed power time series P related to the chosen energy time series E containing 19 labeled anomalies from the second selected data set. The contained anomalies are considered as missing values and imputed using the CPI method.

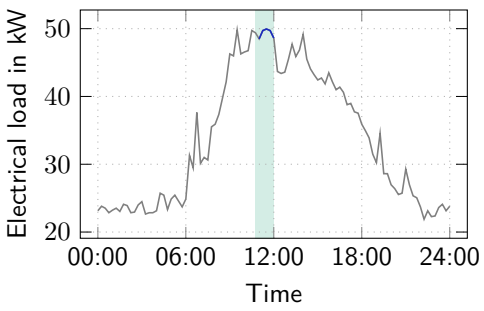
With respect to the related power time series P, we note that the CPI method imputes the labeled anomalies in such a way that the labeled positive and negative peaks as well

as the labeled zero values are all replaced by typical values. Thereby, the values of the imputed time series lie within the usual range of values of the considered time series.

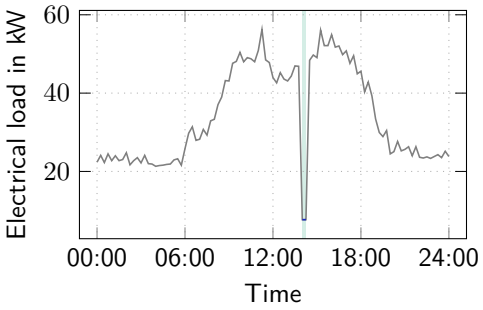
For a closer look at imputed anomalies in the power time series P, we select an exemplary labeled anomaly of all four types. More specifically, we choose the same four anomalies as considered for anomaly detection and as presented in Figure 4.8 in Section 4.2.1 for consistency. Figure 5.18 shows the imputation of the selected exemplary labeled anomalies of types 1 to 4 from the group of technical faults that are considered as missing values and imputed using the CPI method.



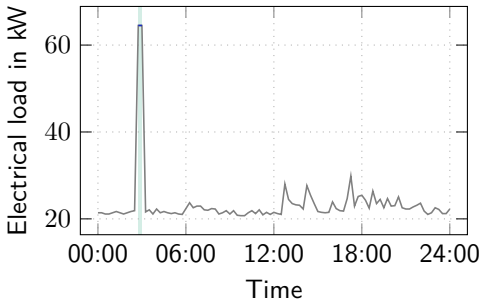
(a) Imputed anomaly of type 1 on 27.09.2016.



(b) Imputed anomaly of type 2 on 15.03.2016.



(c) Imputed anomaly of type 3 on 20.10.2016.



(d) Imputed anomaly of type 4 on 30.10.2016.

Figure 5.18 Examples of the labeled anomalies of types 1 to 4 from the technical faults that are considered as missing values and imputed using the CPI method. Note that the anomalies of types 3 and 4 actually have a length of one but are marked together with the previous value to be recognizable.

In general, for all four exemplary anomalies, the CPI method removes the labeled anomalies with their peaks and zero values and replaces them with patterns in the usual value range of the time series. However, despite the selection of generally matched patterns, the level of the patterns used for imputation is lower or higher than that of the adjacent values for the exemplary anomalies of types 3 and 4. Moreover, if we look more closely at the imputed anomalies of all types and compare them with the labeled anomalies, we find that the imputation by the CPI method always also affects the last data point before each labeled anomaly.

5.4 Discussion

In this section, we discuss the reported results and the benefits of the proposed CPI method for imputing energy time series.

In the evaluation presented in Section 5.3, the CPI method obtains a lower MAPE for most shares of inserted missing values and a lower WAPE for all shares of missing values with respect to the benchmarks. From these results and the exemplary imputations in the data with inserted missing values, we infer that the CPI method selects matching blocks of data for the imputation and ensures that the overall energy per gap remains unchanged while imputing the missing values. We assume that the comparatively low WAPE is closely related to the use of energy as a distance measure, which is possible thanks to the use of energy time series as input. The scaling performed by the CPI method even reduces the WAPE to nearly zero for all shares of missing values, so the CPI method performs even better. In view of the already good results without scaling, the contribution of scaling is, however, relatively small.

Furthermore, we consider the computational cost of the CPI method to be comparatively moderate because it requires more run-time than the Linear Interpolation, more or less run-time than the OWA method depending on the share of missing values, and less run-time than the Prophet-based method. The run-time of the CPI method additionally scales well with increasing shares of missing values and the length of the input time series. One reason for the observed run-times of the CPI method could be the required model estimation observed in the run-time decomposition. Nevertheless, when relating the computational cost to the use of matching patterns, the CPI method, like the OWA method, provides accurate results with a reasonable run-time. Interestingly, we also observe that the Prophet-based method has a longer run-time than the CPI method, which also uses the Prophet method in one step. We assume that this observation is caused by a more time-consuming training of the Prophet-based method due to its larger input. When applied as benchmark method, the Prophet-based method receives 96 values per day as input as opposed to only one value per day when used in the CPI method.

When applying the CPI method to an exemplary real-world energy time series E with labeled anomalies that are considered as missing values, the resulting imputation leads to a monotonously increasing energy time series E as desired. Considering the related imputed power time series P, the CPI method generally imputes the labeled peaks and zero values with matching patterns in the typical value range of the considered time series. The shown imputations of the exemplary anomalies of the types 1 to 4 from the group of technical faults generally confirm this observation. However, the imputations of the anomalies of types 3 and 4 are on a lower or higher level than the values adjacent to the imputed values. The reason for the inappropriate levels could be that anomalies of types 3 and 4 have a length of one and that the CPI method thus linearly interpolates the corresponding missing value in the energy time series E. As a result, the sum of the consumed or produced energy during the gap and its previous value is just split between these two values in the linear interpolation. While this procedure is generally correct, anomalies of

types 3 and 4 violate the underlying assumption that the amount of energy consumed or produced during a gap is the same as that of the imputed values. For this reason, future work could adapt the CPI method such that it linearly interpolates anomalies of types 3 and 4 in the related power time series P. However, this adaptation would also require the CPI method to update the related energy time series E because of the changed amount of energy. Furthermore, we observe in the imputed exemplary anomalies that the imputations of the anomalies of types 1 and 2 by the CPI method start also one data point before the actually labeled anomalies. We suppose that this is caused by the CPI method's derivation of the power time series P from the energy time series E that always affects two values. Therefore, future work could investigate whether this derivation causes the longer imputations and could adapt the respective imputations in the power time series P.

Overall, the proposed CPI method provides several benefits. The main benefit is that the CPI method generally provides a realistic imputation for energy time series, even for large gaps with several weeks of consecutively missing values. In contrast to almost all other methods in the literature, the CPI method uses the often provided energy time series E_i, i. e., the actual meter readings, instead of power time series P_i, i. e., the average power per interval. Using an energy time series E allows including the information on the total energy consumed or produced during gaps. Using this information enables a robust selection of matching blocks of data and ensures that the overall energy per gap remains unchanged while imputing the missing values with realistic patterns. Through imputing missing values, the CPI methods increases the completeness of collected energy time series E_i and of the derivable power time series P_i, which are then available to smart grid applications relying on complete input data. For the imputation itself, the CPI method does not need any additional information such as weather data or consumption data from spatially close smart meters. It only requires three parameters in the dissimilarity measure, however their choice usually does not strongly influence the performance of the CPI method.

5.5 Contribution and Future Work

In the present chapter, we investigate how anomalies detected in energy time series can be compensated, thus answering research question [RQ3]. For this, we compare the performance of the proposed CPI method with benchmark methods on two real-world data sets, one with inserted missing values and one with detected anomalies considered as missing values. The comparison includes the use of matching patterns, the conservation of energy, the computational cost and its decomposition, and the relation of the use of matching patterns and the computational cost. Lastly, we present exemplary imputations to visually illustrate the evaluation result and show an example of how the CPI method imputes data with detected anomalies considered as missing values.

Following this approach, the present chapter makes the following contributions:

- We propose the CPI method for univariate energy time series. Using an energy time series E as input, it copies blocks of data with similar characteristics and pastes them into gaps of the time series.
- We demonstrate that the proposed CPI method outperforms the selected benchmark methods for six different shares of inserted missing values regarding the use of matching patterns and the conservation of the overall energy of every imputed gap.
- We show that the CPI method also has only moderate computational cost compared to the benchmark methods and that this cost scales well with increasing shares of missing values and the length of the input time series. Additionally, we present that the CPI method offers a good trade-off between the use of matching patterns and the computational cost.
- We confirm in the presented exemplary imputations that the CPI method copies matching patterns into gaps but has difficulties with anomalies comprising unrealistic low or high amounts of energy.

Given the proposed CPI method, future work could follow different directions. The robustness of the CPI method could be analyzed and improved regarding aperiodic events or time series with other temporal resolutions and periodicities such as residential solar power generation or fast charging of electrical vehicles. Similarly, time series containing both power consumption and generation from renewable energy sources could be of interest for further investigation. A robustness analysis could further include the selection of the weights in the dissimilarity criterion – for example in dynamic environments – and the compilation of the calibration data set used for determining the weights. Furthermore, imputing anomalies comprising unrealistic low or high amounts of energy and the transitions between existing and imputed values on both ends of gaps could be further investigated.

Moreover, a trend analysis could enhance the CPI method's selection of matching days especially for longer gaps. Similarly, additional information such as voltage magnitude and spatial temporal correlations could be used to improve the matching days selection.

Furthermore, the CPI method could be integrated in applications that rely on complete input data such as grid simulation, load forecasting, and load management. Anomaly or error detection functions could also be included in the CPI method itself to repair implausible values. Moreover, a reporting and analysis tool could use the CPI method to estimate the imputation quality based on inserted missing values.

6 Managing Anomalies in Energy Time Series Forecasting

As motivated in the introduction, anomaly management can aim to account for anomalies in energy time series forecasting. In general, considering anomalies and their potential impact has a long history in statistics (Box and Tiao 1975; Chang et al. 1988; Denby and Martin 1979) and is still a challenge in various domains, including finance (Grané and Veiga 2014; Nyitrai and Virág 2019), the process industry (Xie et al. 2016; Yin and Wang 2013; Yin et al. 2014), and organizational science (Aguinis et al. 2013).

In the energy system, the potential influence of anomalies on applications such as billing and forecasting is also generally known (Wang et al. 2019). When developing methods for anomaly detection and compensation, some works use the resulting forecasting performance as an evaluation metric (e.g., Akouemo and Povinelli 2016; Akouemo and Povinelli 2017; Quintana et al. 2022), thus implicitly expressing the importance of appropriately managing anomalies. In energy time series forecasting, anomalies are, however, typically only taken into account as a necessary but not elaborated step of the pre-processing, after the used input data has been identified as limiting the forecast performance (e.g., Ben Taieb and Hyndman 2014; Charlton and Singleton 2014; Ranjan et al. 2021; Xie and Hong 2016).

At the same time, the influence of the input data on important system functions and applications is increasingly recognized especially under the development of the energy system to a cyber-physical system. As a result, investigating the robustness of forecasting methods against anomalous data (e.g., Luo et al. 2018a; Zhang et al. 2020), strengthening existing forecasting methods (e.g., Zhou et al. 2022), and developing forecasting methods resilient to cyber attacks (e.g., Jiao et al. 2022; Luo et al. 2018b; Luo et al. 2023; Yue et al. 2019; Zheng et al. 2020) gains growing attention in the literature and proves the importance of anomaly management in energy time series forecasting. Also with a focus on the existing limited data quality or decision optimization, energy forecasting methods are proposed that explicitly correct the used data input by detecting anomalies and replacing the detected

Parts of this chapter are reproduced from

M. Turowski, O. Neumann, L. Mannsperger, K. Kraus, K. Layer, R. Mikut, and V. Hagenmeyer (2023). *Managing Anomalies in Energy Time Series for Automated Forecasting*. Submitted.

anomalies with appropriate values (e.g., Chakhchoukh et al. 2011; Chen et al. 2014; Luo et al. 2018c) or use the information on predicted anomalies to adapt energy production (Teng et al. 2022). Despite these advances regarding specific forecasting methods and the general consideration of anomalies, it is not known how best to deal with anomalies. However, a proper anomaly management is essential, thus a rigorous comparison of available strategies for managing anomalies in energy time series forecasting is lacking.

Therefore, the present chapter proposes and evaluates different general strategies for managing anomalies in energy time series forecasting. For this purpose, we build on the typically used strategies mentioned above and describe four different general strategies, namely the raw, the robust, the detection, and the compensation strategy. While the raw strategy uses the data input without any changes for the forecast, the robust strategy applies robust forecasting methods to the data input. The detection strategy provides information on anomalies detected in the input data to the forecasting method and the compensation strategy detects and compensates anomalies in the input data before applying a forecasting method.

To examine these strategies, we perform a four-step evaluation using a representative selection of forecasting methods and two real-world data sets, namely data with inserted synthetic anomalies derived from real-world data and data containing labeled anomalies. In a first step, we compare the raw and robust strategies that use the same input data but different forecasting methods to identify which of them performs better. Afterward, we determine in a second step for the detection strategy and in a third step for the compensation strategy whether the selected supervised or unsupervised anomaly detection method provides the best basis for the respective strategy. Lastly, we compare all proposed strategies considering the previous findings.

With the evaluated strategies, we answer research question [RQ4] presented in Section 1.1 that addresses how an anomaly management can account for anomalies in energy time series forecasting. By answering research question [RQ4], the proposed strategies provide means to manage anomalies in energy time series forecasting using the pipeline for managing anomalies and considering the prior anomaly detection and compensation (see Figure 6.1).

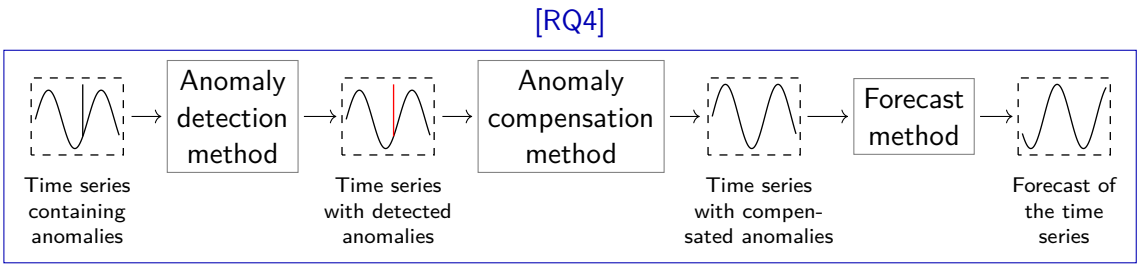


Figure 6.1 By answering research question [RQ4], the proposed strategies provide means to manage anomalies in energy time series forecasting using the pipeline for managing anomalies and considering the prior anomaly detection and compensation.

The remainder of the present chapter is structured as follows. Section 6.1 presents the strategies for managing anomalies in energy time series forecasting. In Section 6.2, we describe the experimental setting of the performed evaluation. In Section 6.3, we present the results of the evaluation. Finally, we discuss the results and the strategies in Section 6.4 and conclude the chapter in Section 6.5.

6.1 Strategies for Managing Anomalies in Energy Time Series Forecasting

In this section, we present four general strategies for managing anomalies in energy time series forecasting. As shown in Figure 6.2, the strategies differ in the used steps and input to the applied forecasting method. We thus describe the included steps, the used input, and the underlying assumptions for each strategy in the following.

Raw Strategy The first strategy is the so-called raw strategy. It directly uses a power time series containing anomalies P as input to a forecasting method. Based on this input, the applied forecasting method provides a forecast of the power time series P. The raw strategy assumes that the anomalies contained in the input time series do not strongly influence the forecast from the applied forecasting method and that thus the applied forecasting method still achieves an accurate forecast.

Robust Strategy The second strategy is the so-called robust strategy. In this strategy, the power time series containing anomalies P also directly serves as input to a forecasting method that provides the forecast of the power time series P. However, the applied forecasting method is considered as robust against anomalies. This strategy assumes that forecasting methods differ in their robustness against anomalies in the input data and thus robust methods can provide accurate forecasts regardless of anomalies present in the input data.

Detection Strategy The third strategy is the so-called detection strategy. This strategy first applies a supervised or unsupervised anomaly detection method to the power time series containing anomalies P. The resulting power time series with detected anomalies P serves as input to the forecasting method that then provides the forecast of the power time series P. The assumption of the detection strategy is that the applied forecasting method can incorporate information about detected anomalies in its model so that the consideration of detected anomalies leads to an accurate forecast.

Compensation Strategy The fourth strategy is the so-called compensation strategy. It also first applies a supervised or unsupervised anomaly detection method to the power time series containing anomalies P. However, this strategy then uses the power time series with detected anomalies P as input to an anomaly compensation method that replaces the detected anomalies with realistic values. The resulting power time series with compensated detected anomalies P serves as input for the forecasting method that provides the forecast of the power time series P. The compensation strategy assumes the anomalies have to be compensated in order to enable the applied forecasting method to provide an accurate forecast.

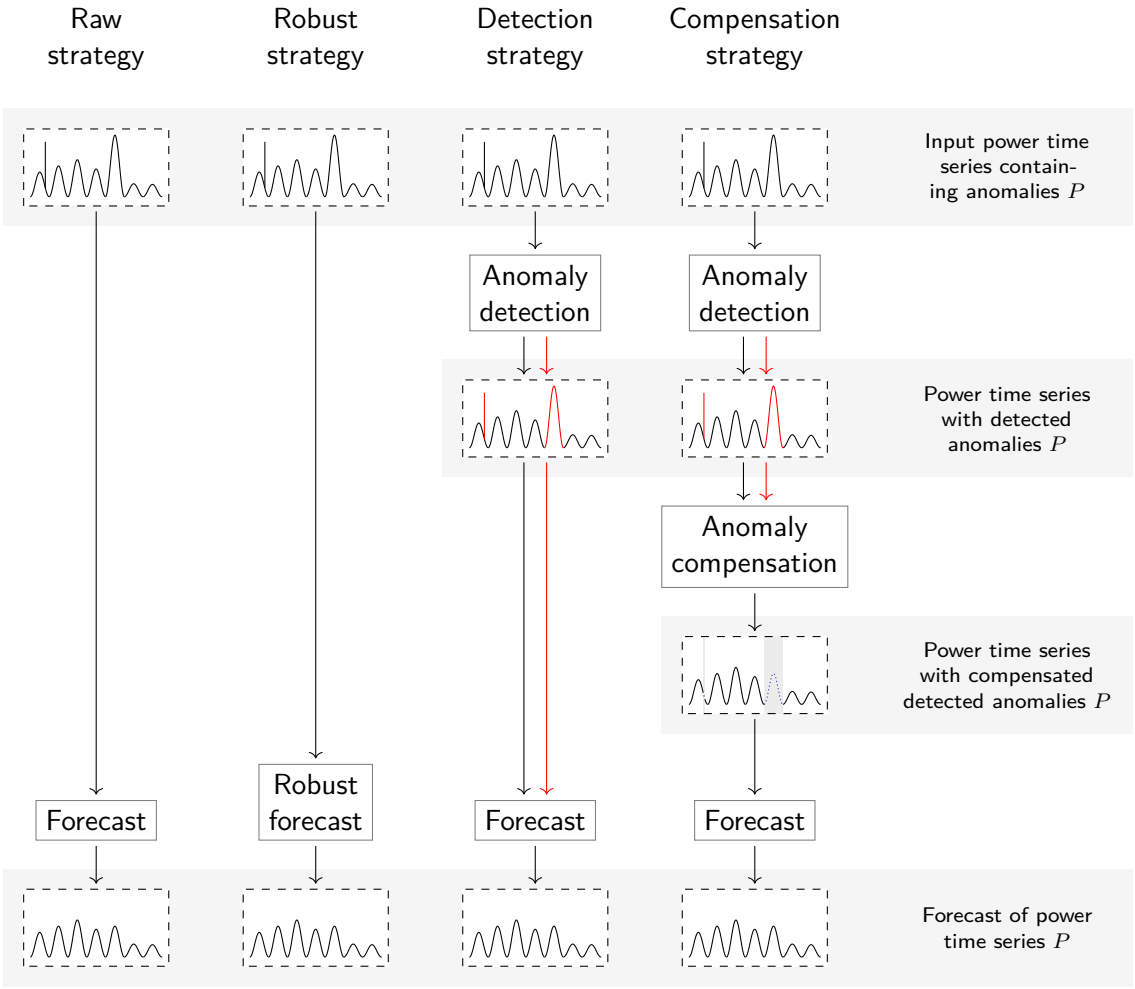


Figure 6.2 The four strategies for managing anomalies in energy time series forecasting. The raw strategy directly uses the input power time series P to provide a forecast. The robust strategy applies a robust forecasting method directly to the input power time series P. The detection strategy first detects anomalies in the input power time series P, before providing a forecast using the information on the detected anomalies from the power time series with detected anomalies P. The compensation strategy detects anomalies and additionally compensates the detected anomalies before performing a forecast based on the power time series with compensated detected anomalies P.

6.2 Experimental Setting

In this section, we present how we evaluate the proposed strategies for managing anomalies in energy time series forecasting. After describing the used data and the inserted synthetic anomalies, we introduce the applied anomaly detection approach, the used anomaly detection methods, the applied anomaly compensation method, and the used forecasting methods. Finally, we describe the evaluation criterion and the used hard- and software.

6.2.1 Data Sets with Anomalies

For the evaluation, we use two real-world data sets that differ in the observed anomalies. While we insert synthetic anomalies into the first, the second one contains labeled anomalies.

Data with Synthetic Anomalies The first data set is the "ElectricityLoadDiagrams20112014 Data Set"[1] from the UCI Machine Learning Repository (Dua and Graff 2019), which we also use in Chapters 4 and 5 and in Turowski et al. (2022b) and Turowski et al. (2022a). It includes electrical power time series from 370 clients with different consumption patterns (Rodrigues and Trindade 2018). The 370 time series are available in a quarter-hourly resolution for a period of up to four years, namely from the beginning of 2011 until the end of 2014. As in Section 4.2.1, we choose the power time series `MT_200` for the evaluation to cover the entire four-year period, to account for the electrical load of a typical client, and to consider a comparatively anomaly-free time series. For a plot of the entire power time series P, we refer to Figure 4.4a in Section 4.2.1.

Since the chosen time series does not include labeled anomalies, we insert synthetic anomalies into it in the same way as in Section 4.2.1 and in Turowski et al. (2022a): We consider the two anomaly groups technical faults in the metering infrastructure and unusual consumption. For the insertion, we use the defined anomaly types 1 to 4 from the technical faults and the defined anomaly types 5 to 8 from the unusual consumption. Anomalies of types 1 to 4 are based on anomalies identified in real-world power time series in Turowski et al. (2022b). These anomalies violate the underlying distribution corresponding to normal behavior. Anomalies of types 5 to 8 represent unusual behavior as described in Turowski et al. (2022a). These anomalies are characterized by an unusually low or high power consumption. Figures 4.5 and 4.6 in Section 4.2.1 show exemplary anomalies of types 1 to 4 and types 5 to 8. For the evaluation, we insert once 20 anomalies of types 1 to 4 each from the group of technical faults and once 20 anomalies of types 5 to 8 each from the group of unusual consumption into the selected time series. The inserted anomalies correspond to 5 % of the data for the technical faults and 11 % of the data for the unusual consumption.

1 https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

Data with Labeled Anomalies The second data set is the electrical data collected on the Campus North of the Karlsruhe Institute of Technology (KIT), which we also use in Chapters 3 to 5 and in Turowski et al. (2022b). From this data, we choose the smart meter of the previously considered typical mid-campus office building: Its power time series P from 2016 is used for the evaluation in Chapter 4 and its energy time series E from 2016 for the evaluation in Chapter 5. For the present evaluation, we also choose its power time series P from 2016 as it contains 19 labeled anomalies. The labeled anomalies belong to the four identified anomaly types of the group of technical faults described in Chapter 3 and correspond to 6 % of the data. For more details on the contained anomalies, a plot of the selected power time series P, and a plot of exemplary contained anomalies, we refer to Table 4.1 and Figures 4.7a and 4.8 in Section 4.2.1.

6.2.2 Applied Anomaly Detection Approach

Given the general performance improvement of the approach introduced in Chapter 4 and in Turowski et al. (2022a) for enhancing the detection performance of supervised and unsupervised anomaly detection methods using the latent space data representation, we also apply this approach in the evaluation of the proposed strategies. Following this approach, we apply selected anomaly detection methods to the latent space representation of the used data created by a trained generative method (see Figure 6.3). We select a conditional Invertible Neural Network (cINN) (Ardizzone et al. 2019) and a conditional Variational Autoencoder (cVAE) (Sohn et al. 2015) as generative methods for creating the latent space representations. For both, we use the implementations detailed in Section 4.2.2.

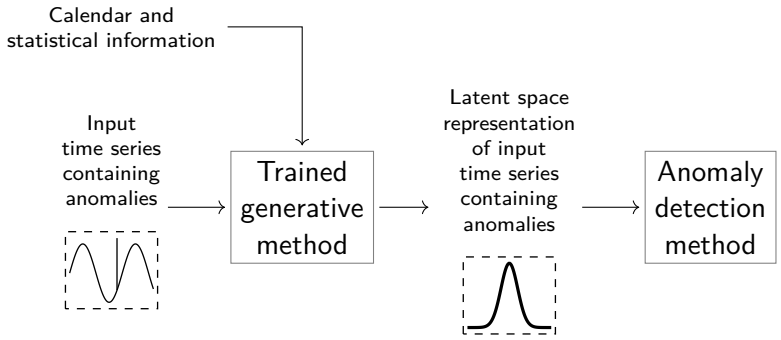


Figure 6.3 According to the selected anomaly detection approach, a trained generative method creates the latent space data representation of an input time series containing anomalies. The latent space data representation then serves as input to an anomaly detection method.

The training of the used conditional Invertible Neural Network (cINN) and conditional Variational Autoencoder (cVAE) follows the training described in Section 4.2.2: We train the supervised cINN and cVAE using a certain number of data points from the selected data, namely the first 15,000 data points from the data with synthetic anomalies and the first 8,735 data points from the data with labeled anomalies. The selected numbers of data points correspond to about five months and three months of data respectively.

We apply the unsupervised cINN and cVAE to the data with inserted synthetic anomalies assuming 10 % of anomalous data and to the data with labeled anomalies assuming 5 % of anomalous data by setting the contamination parameter of the unsupervised cINN and cVAE to 0.1 and 0.05 respectively. Regardless of supervised or unsupervised anomaly detection, both generative methods obtain standardized data points of the selected time series of a data set as samples with a size of 96. Both generative methods also use the mean of the considered time series sample as statistical information as well as the hour of the day, the month of the year, and the weekday as calendar information.

6.2.3 Applied Anomaly Detection Methods

Since the evaluation of the selected anomaly detection approach in Section 4.3 and in Turowski et al. (2022a) already assesses a variety of anomaly detection methods on the selected data sets, we use these evaluation results to select anomaly detection methods for the present evaluation. For each selected data set and group of anomalies, we choose the best-performing latent space representation, supervised anomaly detection method with best-performing hyperparameters, and unsupervised anomaly detection method for the present evaluation of the proposed strategies (see Table 6.1). We briefly describe each supervised and unsupervised anomaly detection method selected based on the evaluation result and their application in the following.

Table 6.1 Overview of the supervised and unsupervised anomaly detection methods and latent space representations applied to the selected data sets and group of anomalies in the evaluation of the proposed strategies.

Data set	Group of anomalies	Type of detection method	Selected latent space representation	Selected anomaly detection method
Data with synthetic anomalies	Technical faults	Supervised	cINN	NB
		Unsupervised	cINN	VAE
	Unusual consumption	Supervised	cINN	XGBoost
		Unsupervised	cVAE	LOF
Data with labeled anomalies	Technical faults	Supervised	cINN	SVC
		Unsupervised	cVAE	iForest

Supervised Methods The first considered supervised anomaly detection method is the Gaussian Naïve Bayes (NB). Based on the assumption that each pair of input features is independent from each other, it estimates a conditional probability using the prior probability of the output variable (Tan et al. 2019). The second considered supervised method is the XGBoost. As a gradient boosting machine, it uses decision trees with gradient decent to minimize a regularized objective function (Chen and Guestrin 2016). The third considered supervised method is the Support Vector Machine for Classification (SVC). It determines the hyperplane with the highest distance to the nearest data points of the binary classes and uses this hyperplane to classify test samples (Vapnik 2000).

Unsupervised Methods The first considered unsupervised anomaly detection method is the Variational Autoencoder (VAE). It learns to map its input to its output using the probability distribution of ideally anomaly-free data in the latent space, so it is trained to only reconstruct non-anomalous data (Kingma and Welling 2014). The second considered unsupervised method is the Local Outlier Factor (LOF). It estimates the local density of a sample by the distance to its k-nearest neighbors and uses low local densities compared to its neighbors to determine anomalies (Breunig et al. 2000). The third considered unsupervised detection method is the Isolation Forest (iForest). It creates an ensemble of isolation trees and uses short average path lengths in these trees to determine anomalies (Liu et al. 2008).

Application We apply the selected supervised and unsupervised anomaly detection methods in the same way as described in Section 4.2.4: The unsupervised detection methods make use of the complete selected time series of each data set. The supervised detection methods, however, use the first 5,000 data points for training and are then applied to all data points except the first 15,000 or 8,735 data points respectively used for the training of the generative methods.

6.2.4 Applied Anomaly Compensation Method

For the anomaly compensation in the evaluation of the proposed strategies, we consider the results of the evaluation of imputation methods applied for compensation in Chapter 5 and in Weber et al. (2021). In that evaluation, the Copy-Paste Imputation (CPI) method (Weber et al. 2021), that requires an energy time series E as input, outperforms all other evaluated methods. However, given that the present evaluation uses power time series P as data, we select the Prophet-based imputation method as the second best method for the evaluation of the proposed strategies as it works with power time series P. The Prophet-based imputation method is built on the forecasting method Prophet which is capable to estimate a time series model on irregularly spaced data (Taylor and Letham 2018). Prophet uses a modular regression model that considers trend, seasonality, and holidays as key components. It can be described as

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{6.1}$$

where g models the trend, s the seasonality, h the holidays, and ϵ_t all other changes not represented in the model. The Prophet-based imputation method trains the regression model using all values available in the power time series P. Given the trained regression model, the Prophet-based imputation method considers all anomalies in the power time series P as missing values and imputes them with the corresponding values from the trained regression model.

6.2.5 Anomaly-Free Baseline Strategy

In the evaluation, we examine the proposed raw, robust, detection, and compensation strategies all based on data containing inserted synthetic or labeled anomalies. For the data in which we insert synthetic anomalies, we additionally provide an anomaly-free baseline for the evaluation of these strategies. This baseline strategy comprises forecasts that are calculated on that selected data but without any inserted anomalies (see Figure 6.4).

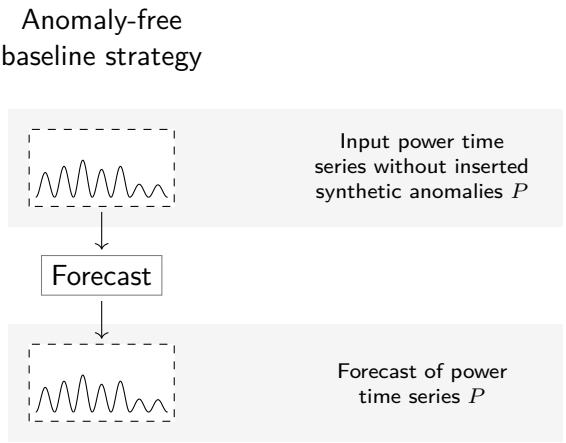


Figure 6.4 For the evaluation of the proposed strategies on the data with inserted synthetic anomalies, we use the forecast calculated on the input power time series without inserted anomalies P as an anomaly-free baseline strategy.

6.2.6 Applied Forecasting Methods

For the evaluation of the proposed strategies, we consider an one-step 15 minutes-ahead forecast for which we apply a representative selection of forecasting methods to the selected data sets. We first present the applied forecasting methods and their input data for the baseline, raw, and compensation strategies, before we describe them for the detection strategy, the robust strategy, and the anomaly-free baseline strategy. We lastly present the used train-test split.

Methods Applied in Raw and Compensation Strategies To examine the raw and compensation strategies comprehensively, we consider methods with different learning assumptions. We apply six forecasting methods, namely two naive and four advanced methods, where the advanced methods comprise a simple statistical method, a simple and a more complex machine learning method, and a statistical learning method.

The first naive method is the Last Value Forecast. It uses the previous value for the value to be predicted, i. e.,

$$\hat{y}_t = y_{t-1}, \tag{6.2}$$

where $\hat{y}_t$ is the forecast value of the electrical load at time t and y_{t-1} is the electrical load at time $t-1$.

The second naive method is the Last Week Forecast. It takes the corresponding value of the last week as the forecast value, i. e.,

$$\hat{y}_t = y_{t-672}, \tag{6.3}$$

where $\hat{y}_t$ is the forecast value of the electrical load at time t and y_{t-672} is the electrical load one week ago at time $t-672$.

The first advanced method is the Linear Regression (LinR). As a statistical method, it models the forecast value as linear relationship between the historic load and calendar information and determines the corresponding parameters using ordinary least squares. It is defined as

$$\hat{y} = c + \sum_j \beta_j \cdot L_j + \sum_k \gamma_k \cdot C_k + \epsilon, \tag{6.4}$$

where c is a constant, L_j are the lagged load features, C_k are the calendar information, and ϵ is the error. Note that we omit the time indices in the equation for the sake of simplicity.

The second advanced method is a commonly applied simple machine learning method, namely a Neural Network (NN). It organizes a network of interconnected nodes in input, hidden, and output layers to apply different functions to activate the corresponding nodes to learn the relationship between input and output (e.g., Werbos 1974; Mitchell 1997). The implementation of the used NN is detailed in Table 6.2. For its training, we use a batch size of 64, the Adam optimizer (Kingma and Ba 2015), and a maximum of 50 epochs.

Table 6.2 Implementation details of the applied NN.

Layer	Description
Input	[Load data, encoded calendar information]
1	Dense 32 neurons; activation: relu
2	Dense 16 neurons; activation: relu
Output	Dense 1 neuron; activation: linear

The third advanced method is the Profile Neural Network (PNN) (Heidrich et al. 2020) as a state-of-the-art and more complex machine learning method. It combines statistical information in form of standard load profiles with convolutional neural networks (CNNs) to improve the forecasting accuracy. For this, it decomposes a power time series P into a standard load profile module, a trend module, and a colorful noise module, before aggregating their outputs to obtain the forecast (Heidrich et al. 2020). For the training, the PNN uses a batch size of 512, the Adam optimizer (Kingma and Ba 2015), and a maximum of 50 epochs.

The fourth advanced method is the Support Vector Regression (SVR). It represents a statistical learning method that determines a regression plane with the smallest distance

to all data points used for the training. The data points closest to the regression plane on both sides are the so-called support vectors (Drucker et al. 1996). We apply the SVR with a RBF kernel, $C = 1.0$, and $\varepsilon = 0.0$.

All introduced forecasting methods use the historical values of the selected power time series P that contains inserted synthetic or labeled anomalies. The advanced methods also consider calendar information as input. The calendar information comprises each weekday encoded as a Boolean and the workdays (Monday to Friday) encoded as a Boolean. It also includes the hour of the day encoded as the sine function $\sin(2 \cdot \pi \cdot \text{hour}/23)$ and the cosine function $\cos(2 \cdot \pi \cdot \text{hour}/23)$, the day of the month encoded as the sine function $\sin(2 \cdot \pi \cdot (\text{day} - 1)/\text{days of the month})$ and the cosine function $\cos(2 \cdot \pi \cdot (\text{day} - 1)/\text{days of the month})$, and the month of the year encoded as the sine function $\sin(2 \cdot \pi \cdot \text{month}/11)$ and the cosine function $\cos(2 \cdot \pi \cdot \text{month}/11)$. While the naive methods directly use the power values, all other methods obtain the considered calendar information and the lags 4, 96, 192, and 672 of the standardized power values.

Methods Applied in Detection Strategy For the detection strategy that can use information on the detected anomalies for the forecast, we apply the forecasting methods introduced for the raw and compensation strategies with exception of the Last Value Forecast. This way, we also evaluate the detection method using forecasting methods with different learning assumptions. However, we adapt the methods as follows: To further consider a naive method, we modify the Last Week Forecast so that it uses the corresponding value of the second to last week as the forecast value if the value to be predicted is a detected anomaly. In accordance with the detection strategy, all methods obtain the information on the detected anomalies with lags 4, 96, 192, and 672 as additional features.

Methods Applied in Robust Strategy To comprehensively evaluate the robust strategy, we apply three forecasting methods with different learning assumptions that are known to be robust against anomalies. For this, we choose one naive and two advanced methods. The advanced methods comprise a statistical learning method and a more complex machine learning method.

The naive method is the Median Weekday Forecast. It calculates the median values of each weekday and uses the corresponding value as forecast value, i. e.,

$$\hat{y}_t = \tilde{y}_t, \tag{6.5}$$

where $\hat{y}_t$ is the forecast value of the electrical load at time t and $\tilde{y}_t$ is the median value of the corresponding weekday at time t.

The first advanced method is the Random Forest (RF) Regressor representing a statistical learning method. It creates several randomly drawn regression trees and takes the mean of each individual tree's forecast as forecast (Breiman 2001), i. e.,

$$\hat{y}_t = \frac{1}{B} \sum_{b=1}^{B} t_b(x), \tag{6.6}$$

where B is the number of bootstrap samples of the training set, t_b is an individual fitted tree, and x are the values from the test set. For the evaluation, we use $B = 100$.

The second advanced method is the XGBoost Regressor, which represents a more complex machine learning method. It iteratively creates regression trees and uses gradient descent to minimize a regularized objective function (Chen and Guestrin 2016).

These forecasting methods also all use the historical values of the selected power time series P that contains inserted synthetic or labeled anomalies. The advanced methods additionally obtain the above mentioned calendar information as input. While the Median Weekday Forecast uses the historical load values directly, the RF Regressor and the XGBoost Regressor use the considered calendar information and the lags 4, 96, 192, and 672 of the standardized power values.

Methods Applied in Anomaly-Free Baseline Strategy To calculate the anomaly-free baseline strategy for the data containing synthetic anomalies, we apply all forecasting methods described for the raw and compensation strategies as well as the robust strategy to the same data but without inserted synthetic anomalies. These forecasting methods obtain the inputs in the way as described for the raw, compensation, and robust strategies.

Train-Test Split Regardless of the considered strategy, we use the same train-test split for all evaluated forecasting methods. However, the train-test split differs for the considered data sets and applied anomaly detection:

- For the data set with inserted synthetic anomalies, each forecasting method is trained on 80 % of the available data and tested on the remaining 20 %. For all strategies applied to this data, the available data is the selected time series without the first 15,096 data points in the case of supervised anomaly detection and without the first 96 data points in the case of unsupervised anomaly detection. When calculating the anomaly-free baseline strategy for this data set, we use the same period of time as in the case of the unsupervised anomaly detection, i. e., all values except the first 96 data points.
- For the data set with labeled anomalies, each forecasting method is trained on about 66 % of the available data and tested on the remaining 34 % to account for the shorter available data length. For all strategies applied to this data, the available data is the selected time series without the first 8,831 data points in the

case of supervised anomaly detection and without the first 96 data points in the case of unsupervised anomaly detection.

6.2.7 Evaluation Criterion

To evaluate the proposed strategies for managing anomalies in energy time series forecasting, we use the accuracy of the obtained forecasts. We measure the accuracy with the commonly used Root Mean Squared Error (RMSE). It is defined as

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}, \tag{6.7}$$

where N is the number of data points to be predicted. We thus calculate the square root of the squared difference between the value to be forecast and the forecast value divided by the number of data points in the test set.

6.2.8 Hard- and Software

To obtain comparable results, we use the same hardware throughout the evaluation. The used hardware is a 48 core system with 256 GB RAM, where each core has 2.1 GHz. Moreover, we implement all evaluated strategies and used anomaly detection methods, anomaly compensation method, and forecasting methods in Python.

For the anomaly detection using the latent space data representation created by the selected cINN or cVAE, we apply the implementation and methods described in Chapter 4 and in Turowski et al. (2022a). It uses FrEIA[2] and PyTorch[3] (Paszke et al. 2019) for the cINN, PyTorch (Paszke et al. 2019) for the cVAE, the available XGBoost implementation[4] (Chen and Guestrin 2016), Keras[5] (Chollet et al. 2015) for the Variational Autoencoder (VAE), and scikit-learn[6] for all other anomaly detection methods.

Regarding the anomaly compensation, we use the implementation of the Prophet-based method described in Section 5.2 and in Weber et al. (2021) that is based on the available Prophet implementation[7] (Taylor and Letham 2018).

2 https://github.com/VLL-HD/FrEIA
3 https://pytorch.org/
4 https://xgboost.ai/
5 https://keras.io/
6 https://scikit-learn.org/
7 https://facebook.github.io/prophet/

Regarding the forecasting methods, we use Keras[8] (Chollet et al. 2015) for the NN and scikit-learn for the LinR, SVR, and RF Regressor. Additionally, we apply the available implementation (Chen and Guestrin 2016) for the XGBoost Regressor, and adapt the available implementation of the PNN[9] (Heidrich et al. 2020) to work without weather data. We finally use pyWATTS[10] (Heidrich et al. 2021) to implement the proposed strategies and to automate their evaluation.

6.3 Results

To examine the presented general strategies, we perform a four-step evaluation. In a first step, we compare the raw and robust strategies that use the same input data but different forecasting methods to identify which of them performs better. Afterward, we determine in a second step for the detection strategy and in a third step for the compensation strategy whether the selected supervised or unsupervised anomaly detection method provides the best basis for the respective strategy. Finally, we compare all the proposed strategies, taking into account the findings from the previous steps. In all steps, we present the results for the data with inserted synthetic anomalies and the data with labeled anomalies.

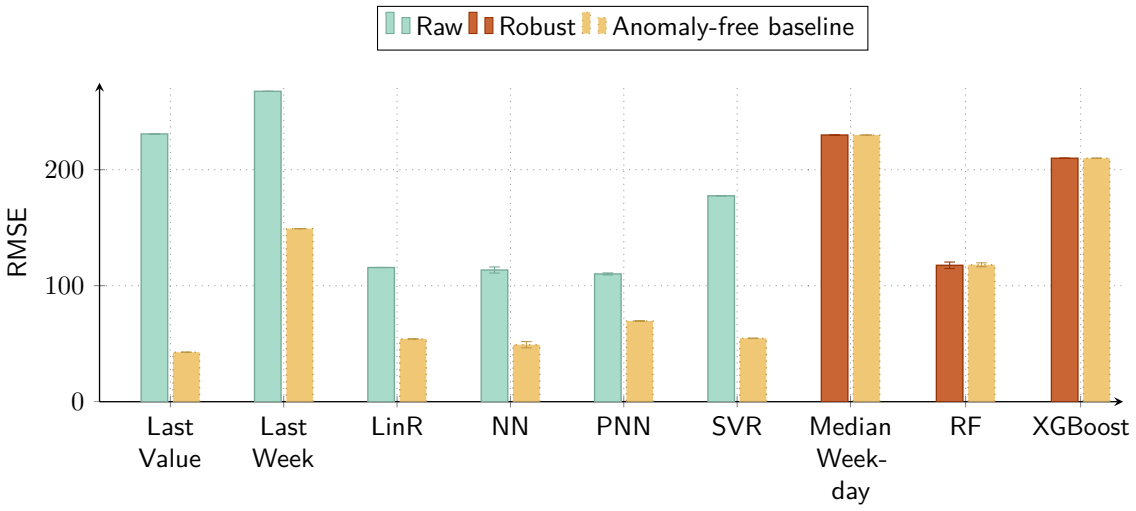
6.3.1 Comparison of Raw and Robust Strategies

Since they use the same input data and only differ in the applied forecasting methods, we compare the raw strategy and the robust strategy. For this comparison, we apply the associated forecasting methods to the data available in the case of the unsupervised anomaly detection. As a reference, we additionally report the anomaly-free baseline strategy for the data with inserted synthetic anomalies.

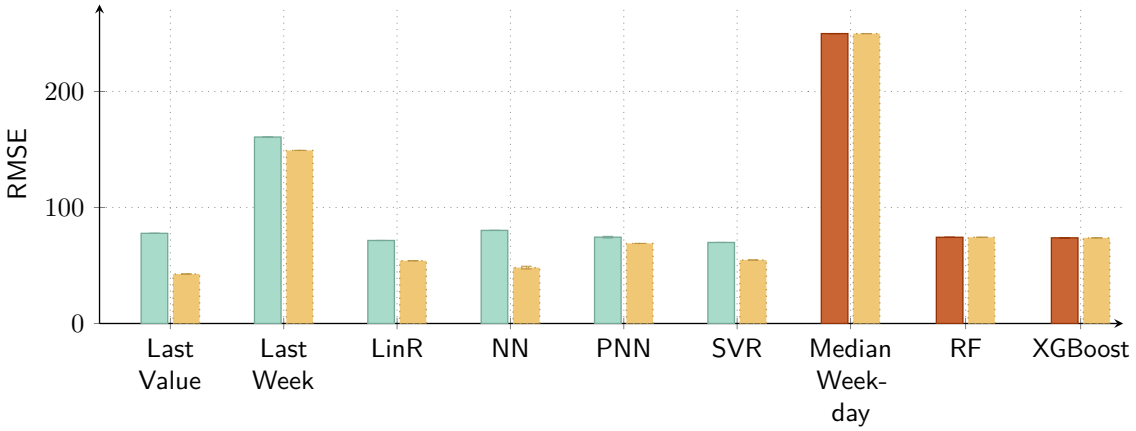
Data With Synthetic Anomalies We evaluate the forecasting methods selected for the raw and robust strategies on the data with inserted synthetic anomalies from both anomaly groups, namely technical faults and unusual consumption. Figures 6.5a and 6.5b present the resulting RMSE for the technical faults and unusual consumption. The bars show the average RMSE for the raw or robust strategy and the anomaly-free baseline strategy. The error bars indicate the best and the worst observed RMSE.

With regard to the technical faults, we observe that forecasting methods using the raw strategy have a noticeably higher RMSE compared to the anomaly-free baseline strategy as the forecasting methods using the robust strategy. The forecasting methods using the robust strategy achieve RMSEs that are as low as the RMSE when using the anomaly-free baseline strategy. When comparing the forecasting methods using the

8 https://keras.io/
9 https://github.com/benHeid/Profile-Neural-Network
10 https://github.com/KIT-IAI/pyWATTS



(a) Technical faults.



(b) Unusual consumption.

Figure 6.5 The RMSE of the six forecasting methods using the raw strategy and three forecasting methods using the robust strategy that are applied to the data with 20 synthetic anomalies of each type from the technical faults and unusual consumption. For each method, the bars indicate the average RMSE for the raw or robust strategy and the anomaly-free baseline strategy. The error bars show the best and the worst observed RMSE. Note that the anomaly-free baseline strategy generally performs best because it uses data that does not contain inserted synthetic anomalies.

raw and robust strategies regarding their actual accuracy, the LinR, the NN, and the PNN using the raw strategy and the RF Regressor using the robust strategy obtain the lowest RMSE. The SVR achieves the next best RMSE and is followed by the XGBoost Regressor, the Last Value Forecast, the Median Weekday Forecast, and the Last Value Forecast. The Last Week Forecast has the worst RMSE.

For the unusual consumption, the forecasting methods using the raw strategy show a reduced higher RMSE compared to the anomaly-free baseline strategy. The forecast-

ing methods using the robust strategy again have an RMSE that is very similar to that of the anomaly-free baseline strategy. Regarding the actual accuracy of the forecasting methods using the raw and robust strategies, all forecasting methods except the Last Week Forecast using the raw strategy and the Median Weekday Forecast using the robust strategy achieve similar RMSEs for both strategies. In comparison, the RMSEs of the Last Week Forecast using the raw strategy and the Median Weekday Forecast using the robust strategy are noticeably higher.

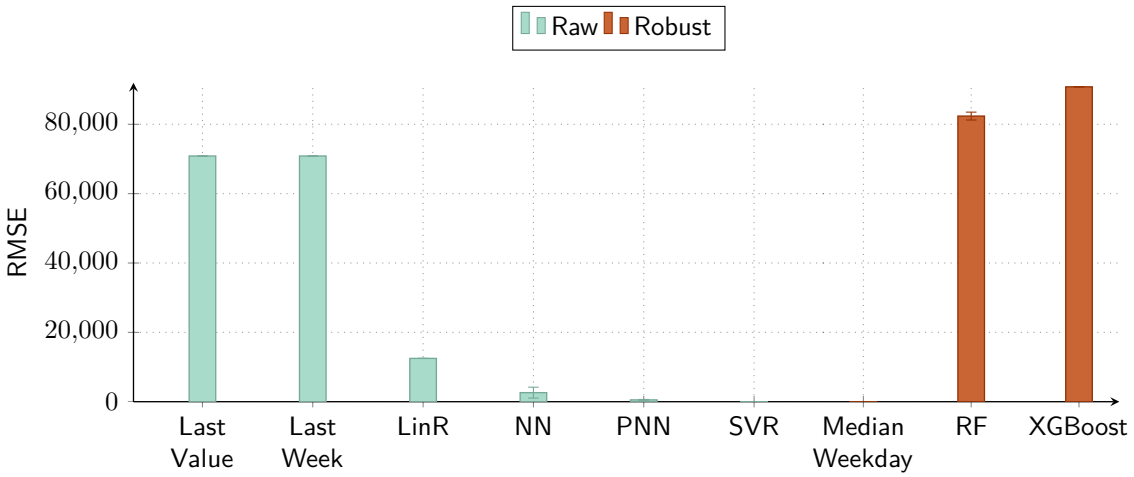
Data With Labeled Anomalies We also evaluate all forecasting methods selected for the raw and robust strategies on the data with labeled anomalies. Figure 6.6a shows the resulting RMSE of all forecasting methods and Figure 6.6b the resulting RMSE of three best-performing forecasting method for the labeled technical faults. For each forecasting method, the bars show the average RMSE for the raw or robust strategy. The error bars present the best and the worst observed RMSE.

Applied to the data with labeled anomalies, the forecasting methods using the raw or robust strategy perform differently. While the XGBoost Regressor, the RF Regressor, the Last Value Forecast, and the Last Week Forecast attain very high RMSEs, the LinR and the NN show high RMSEs and the PNN, the SVR, and the Median Weekday Forecast comparatively low RMSEs. Considering the latter three forecasting methods, we observe in Figure 6.6b that the SVR and the Median Weekday Forecast achieve a considerably lower RMSE than the PNN.

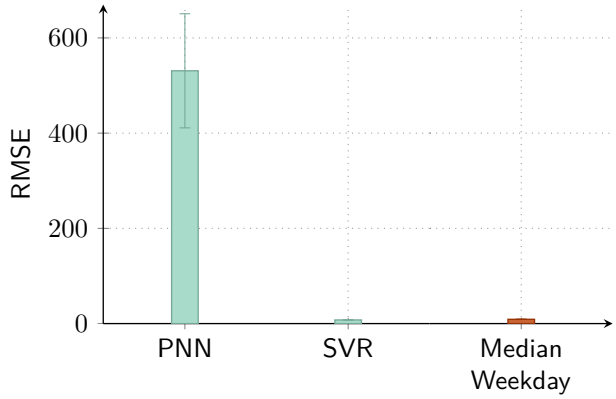
6.3.2 Best Anomaly Detection for Detection Strategy

Since the detection strategy assumes that forecasting methods can incorporate information about detected anomalies in their model and that this incorporation contributes to accurate forecasts, we also examine the anomaly detection for this strategy. To determine the best anomaly detection for the detection strategy, we apply this strategy to data analyzed by the selected supervised anomaly detection method and data analyzed by the selected unsupervised anomaly detection method. In the following, we refer to the detection strategy using the best-performing supervised anomaly detection method as the detection supervised strategy and to the detection strategy using the best-performing unsupervised anomaly detection method as the detection unsupervised strategy. For comparison, we also present the anomaly-free baseline strategy for the data with inserted synthetic anomalies.

Data With Synthetic Anomalies We evaluate the forecasting methods selected for the detection strategy on the data with inserted synthetic anomalies from both technical faults and unusual consumption. For both groups of anomalies, we insert 20 anomalies of each type belonging to this group. Figure 6.7a shows the resulting RMSE for the technical faults and Figure 6.7b for the unusual consumption. For each forecasting method, the bars indicate the average RMSE for the detection supervised



(a) All nine applied forecasting methods.

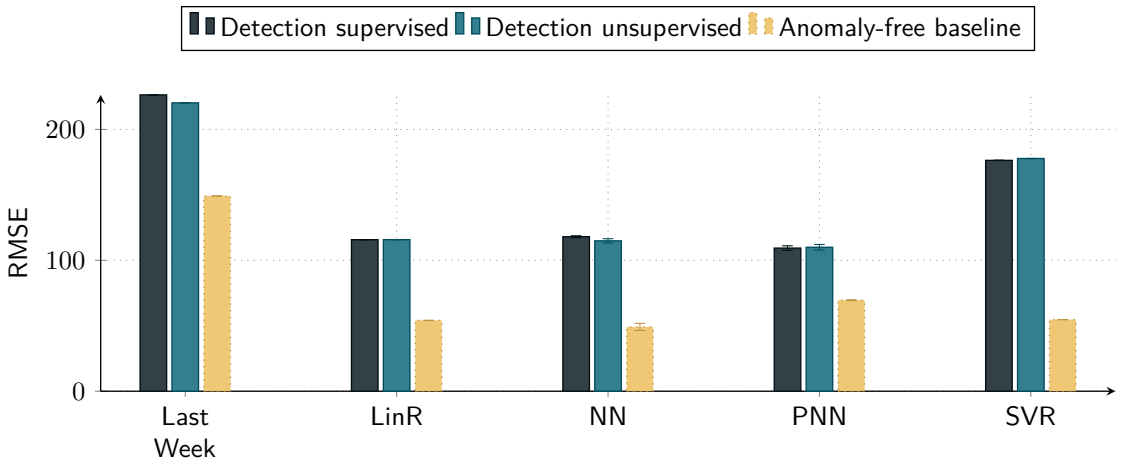


(b) The three best performing forecasting methods.

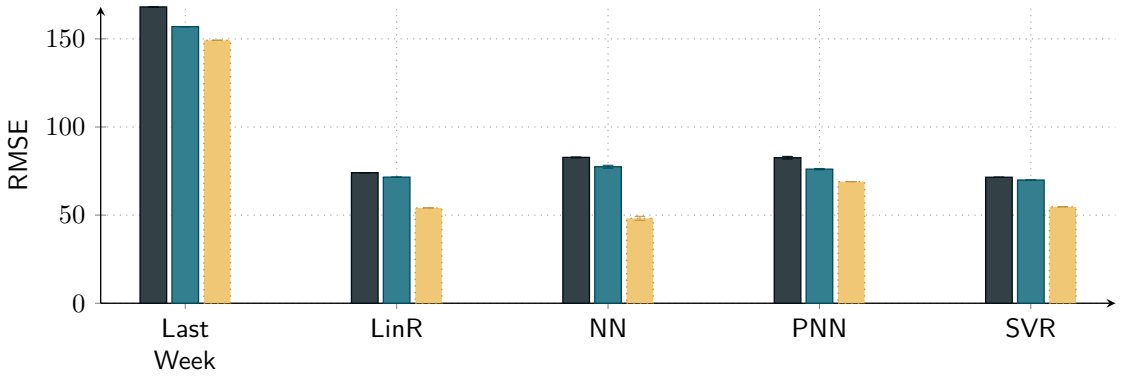
Figure 6.6 The RMSE of the six forecasting methods using the raw strategy and three forecasting methods using the robust strategy that are applied to the data with labeled technical faults. For each method, the bars indicate the average RMSE for the raw or robust strategy. The error bars show the best and the worst observed RMSE.

strategy, the detection unsupervised strategy, and the anomaly-free baseline strategy. The error bars show the best and the worst observed RMSE.

For the technical faults, the RMSE of the detection unsupervised strategy is similar to that of the detection supervised strategy for all six considered forecasting methods. The detection unsupervised strategy has only a slightly higher RMSE for the SVR but a slightly lower RMSE for the Last Week Forecast and the NN. Compared to the anomaly-free baseline strategy, the RMSE of all forecasting methods is also clearly greater for both detection strategies. Moreover, considering the actual accuracy of the forecasting methods using the detection strategies, the LinR, the NN, and the PNN achieve the lowest RMSE, the Last Week Forecast and the SVR the highest RMSE.



(a) Technical faults.



(b) Unusual consumption.

Figure 6.7 The RMSE of the five forecasting methods applied to the data with 20 synthetic anomalies of each type from the technical faults and unusual consumption. For each method, the bars indicate the average RMSE for the detection strategy using the best-performing supervised anomaly detection method, the detection strategy using the best-performing unsupervised anomaly detection method, and the anomaly-free baseline strategy. The error bars show the best and the worst observed RMSE. Note that the anomaly-free baseline strategy generally performs best because it uses data that does not contain inserted synthetic anomalies.

For the unusual consumption, we observe slightly stronger differences between the detection supervised strategy and the detection unsupervised strategy. For all considered forecasting methods, the RMSE of the detection unsupervised strategy is lower than that of the detection supervised strategy. Moreover, the RMSEs of both detection strategies are closer to the RMSE of the anomaly-free baseline strategy for all forecasting methods. Furthermore, when comparing the actual accuracy of the forecasting methods using the detection strategies, we observe that the LinR, the NN, the PNN, and the SVR obtain a similarly low RMSE and the Last Week Forecast clearly has the highest RMSE.

Data With Labeled Anomalies We also evaluate all forecasting methods selected for the detection strategy on the data with labeled anomalies. Figure 6.8 shows the resulting RMSE for the labeled technical faults. For each forecasting method, the bars show the average RMSE for the detection supervised strategy and the detection unsupervised strategy. The error bars present the best and the worst observed RMSE.

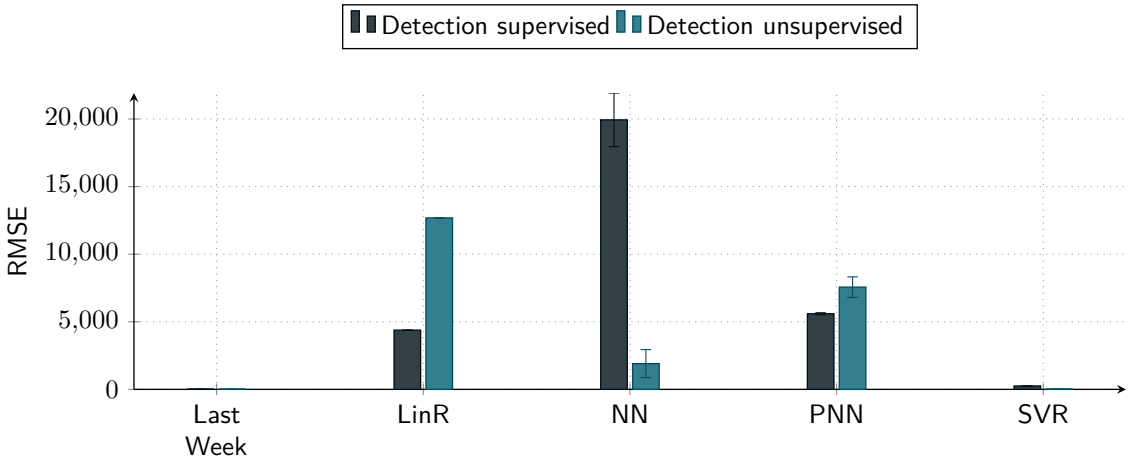


Figure 6.8 The RMSE of the five forecasting methods applied to the data with labeled technical faults. For each method, the bars indicate the average RMSE for the detection strategy using the best-performing supervised anomaly detection method, and the detection strategy using the best-performing unsupervised anomaly detection method. The error bars show the best and the worst observed RMSE.

We observe that the detection supervised strategy has a considerably higher RMSE than the detection unsupervised strategy for the NN and vice versa for the LinR. For the PNN, the detection unsupervised strategy also has a larger but more similar RMSE. For the SVR, the RMSE of the detection unsupervised strategy is smaller than the RMSE of the detection supervised strategy. For the Last Week Forecast, the RMSE of both detection strategies is similarly small. Nevertheless, the obtained RMSE strongly differs between the considered forecasting methods when using the detection strategies: While the Last Week Forecast and the SVR have a comparatively very small RMSE for both detection strategies, all other methods achieve a comparatively very high RMSE with at least one of the two detection strategies.

6.3.3 Best Anomaly Detection for Compensation Strategy

To determine the best anomaly detection for the compensation strategy, we also apply this strategy to data analyzed by the selected supervised anomaly detection method and data analyzed by the selected unsupervised anomaly detection method. In the following, we refer to the compensation strategy using the best-performing supervised anomaly detection method as the compensation supervised strategy and to the compensation

strategy using the best-performing unsupervised anomaly detection method as the compensation unsupervised strategy. For comparison, we also present the anomaly-free baseline strategy for the data with inserted synthetic anomalies.

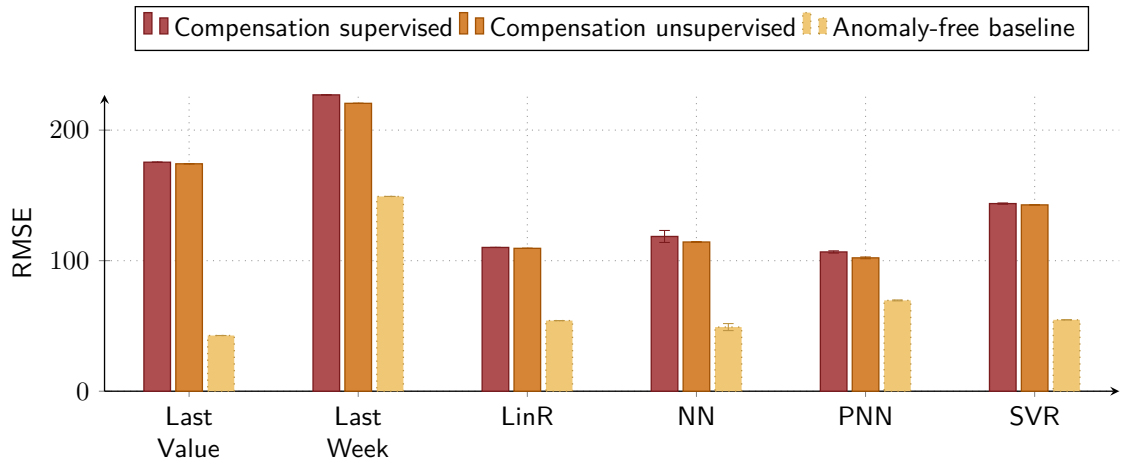
Data With Synthetic Anomalies To examine the forecasting methods selected for the compensation strategy, we consider the data with inserted synthetic anomalies from technical faults and unusual consumption. For both groups of anomalies, we again insert 20 anomalies of each type belonging to this group. Figures 6.9a and 6.9b show the resulting RMSE for the technical faults and for the unusual consumption. For each forecasting method, the bars indicate the average RMSE for the compensation supervised strategy, the compensation unsupervised strategy, and the anomaly-free baseline strategy. The error bars show the best and the worst observed RMSE.

For the technical faults, all six considered forecasting methods achieve similar RMSEs with the compensation supervised strategy and the compensation unsupervised strategy. For the Last Week Forecast, the NN, and the PNN, the compensation unsupervised strategy results in a slightly lower RMSE. In comparison to the anomaly-free baseline strategy, all forecasting methods have, however, a recognizably higher RMSE for both compensation strategies. Regarding the actual accuracy of the forecasting methods using the compensation strategies, the LinR, the NN, and the PNN achieve the lowest RMSE, followed by the SVR, the Last Value Forecast, and the Last Week Forecast.

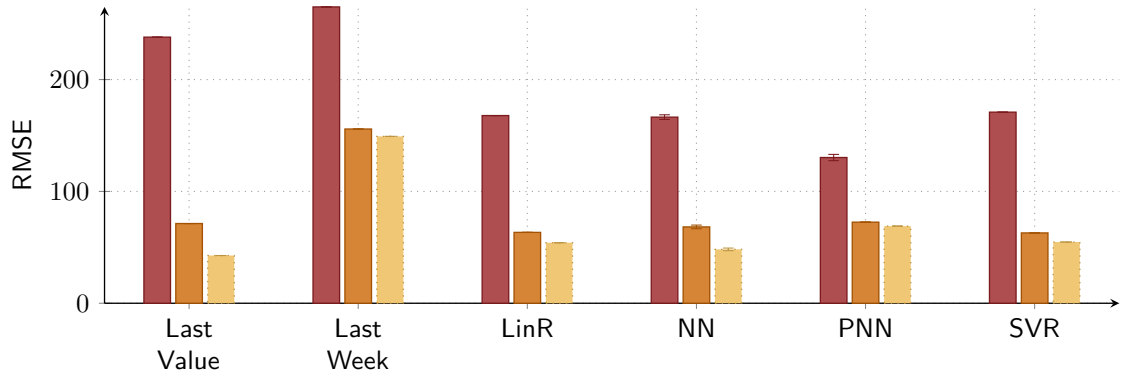
Concerning the unusual consumption, the six forecasting methods obtain different RMSEs with the compensation supervised strategy and the compensation unsupervised strategy. The compensation supervised strategy is associated with a considerably higher RMSE compared to the compensation unsupervised strategy for all evaluated forecasting methods. While the compensation supervised strategy achieves a clearly higher RMSE with all forecasting methods compared to the anomaly-free baseline strategy, the compensation unsupervised strategy yields an RMSE that is close to the RMSE of the anomaly-free baseline strategy. With regard to the actual accuracy of the forecasting methods using the compensation strategies, we observe that all forecasting methods except the Last Week Forecast have a similarly low RMSE with the compensation unsupervised strategy. For the compensation supervised strategy, however, the PNN achieves the lowest RMSE, followed by the similarly well performing LinR, NN, and SVR. The Last Value Forecast and the Last Week Forecast have a higher RMSE.

Data With Labeled Anomalies We also examine the forecasting method selected for the compensation strategy on the data with labeled anomalies. Figure 6.10 presents the resulting RMSE for the labeled technical faults. For each evaluated forecasting method, the bars show the average RMSE for the compensation supervised strategy and the compensation unsupervised strategy. The error bars present the best and the worst observed RMSE.

We observe that all forecasting methods except the PNN achieve a lower RMSE using the compensation supervised strategy than using the compensation unsupervised strategy. For the Last Value Forecast, the LinR, the NN, and the SVR, this difference is clearly



(a) Technical faults.



(b) Unusual consumption.

Figure 6.9 The RMSE of the six forecasting methods applied to the data with 20 synthetic anomalies of each type from the technical faults and unusual consumption. For each method, the bars indicate the average RMSE for the compensation strategy using the best-performing supervised anomaly detection method, the compensation strategy using the best-performing unsupervised anomaly detection method, and the anomaly-free baseline strategy. The error bars show the best and the worst observed RMSE. Note that the anomaly-free baseline strategy generally performs best because it uses data that does not contain inserted synthetic anomalies.

noticeable. The opposite difference is also recognizable for the PNN. The forecasting methods also obtain different RMSEs using the compensation strategies. The Last Value Forecast, the LinR, the NN, and the SVR have the lowest RMSE using the compensation supervised strategy, followed by the PNN and the Last Week Forecast. Using the compensation unsupervised strategy, the NN achieves the lowest RMSE, followed by the similarly well performing Last Value Forecast, the LinR, the PNN, and the SVR. The Last Week Forecast has a higher RMSE.

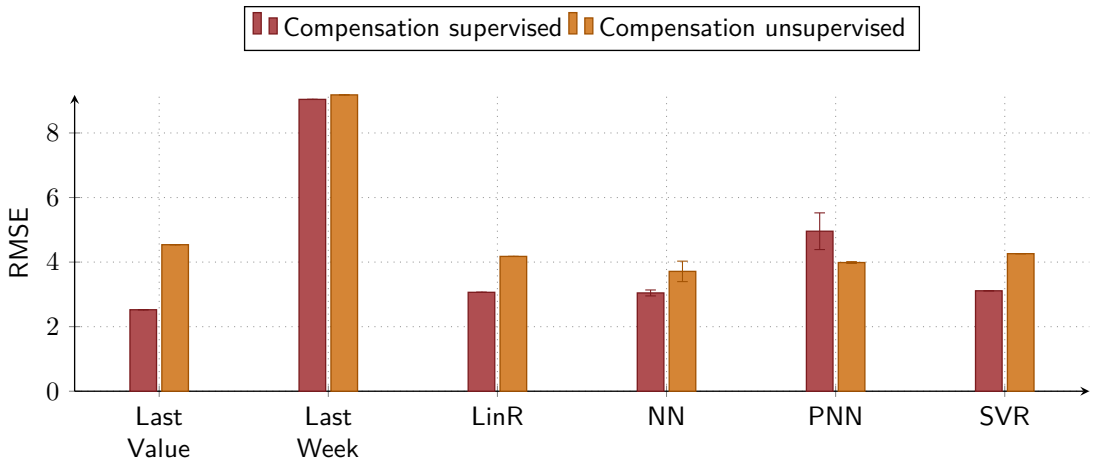


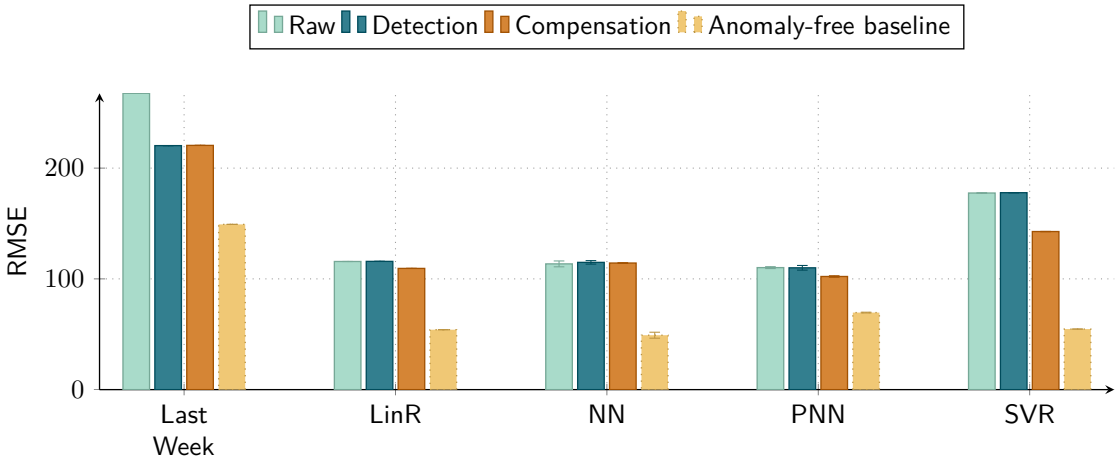
Figure 6.10 The RMSE of the six forecasting methods applied to the data with labeled technical faults. For each method, the bars indicate the average RMSE for the compensation strategy using the best-performing supervised anomaly detection method and the compensation strategy using the best-performing unsupervised anomaly detection method. The error bars show the best and the worst observed RMSE.

6.3.4 Comparison of all Strategies

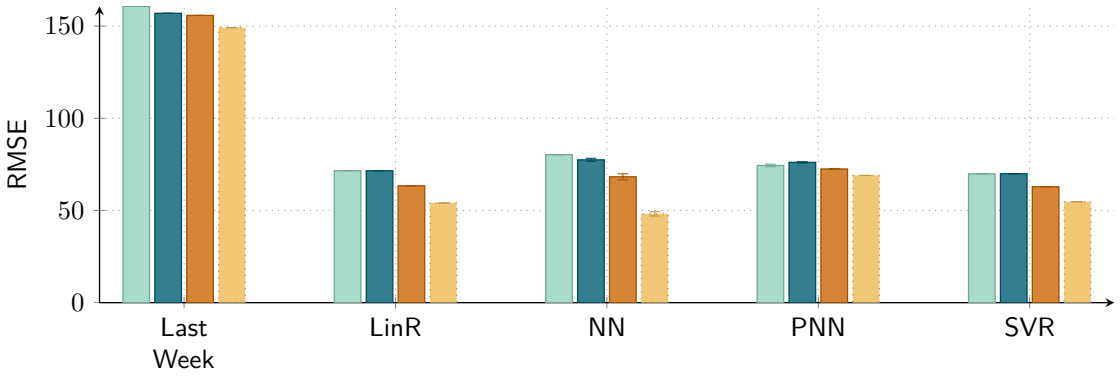
We finally compare all proposed strategies considering the previous findings. We first briefly describe the previous findings and their implication for the comparison of all proposed strategies, before we present the results of this comparison.

When comparing the raw and the robust strategies, we find that most forecasting methods of the raw strategy provide a lower or at least similar RMSE as the forecasting methods of the robust strategy. We thus consider the raw strategy in the final comparison and do not consider the robust strategy any further. In the comparison of the detection supervised and the detection unsupervised strategies, we observe that the detection unsupervised strategy results in a similar or even slightly lower RMSE for all forecasting methods applied to the data with synthetic anomalies and in a low RMSE for the methods that perform best on the data with labeled anomalies. We, therefore, select the detection unsupervised strategy for both data sets in the final comparison. In the comparison between compensation supervised and compensation unsupervised strategies, the compensation unsupervised strategy yields a lower RMSE for all forecasting methods when applied to the data with synthetic anomalies. For the data with labeled anomalies, the compensation supervised strategy provides a lower RMSE for almost all forecasting methods. Therefore, we choose the compensation unsupervised strategy for the data with synthetic anomalies and the compensation supervised strategy for the data with labeled anomalies in the final comparison. For the final comparison of the selected strategies, we generally consider all previously used forecasting methods. However, since the Last Value Forecast is not available for the detection strategy, we omit this forecasting method in the following comparison.

Data With Synthetic Anomalies To compare all selected strategies except the robust strategy for the reasons given above, we apply them to the data with synthetic anomalies from technical faults and unusual consumption. For both groups of anomalies, we insert 20 anomalies of each type belonging to this group. Figure 6.11a shows the resulting RMSE for the technical faults and Figure 6.11b for the unusual consumption. For each considered forecasting method, the bars indicate the average RMSE for the raw strategy, the detection strategy, the compensation strategy, and the anomaly-free baseline strategy. The error bars depict the best and the worst observed RMSE.



(a) Technical faults.



(b) Unusual consumption.

Figure 6.11 The RMSE of the five forecasting methods applied to the data with 20 synthetic anomalies of each type from the technical faults and unusual consumption. For each method, the bars indicate the average RMSE for the raw strategy, detection strategy, compensation strategy, and anomaly-free baseline strategy. The error bars show the best and the worst observed RMSE. Note that the anomaly-free baseline strategy generally performs best because it uses data that does not contain inserted synthetic anomalies.

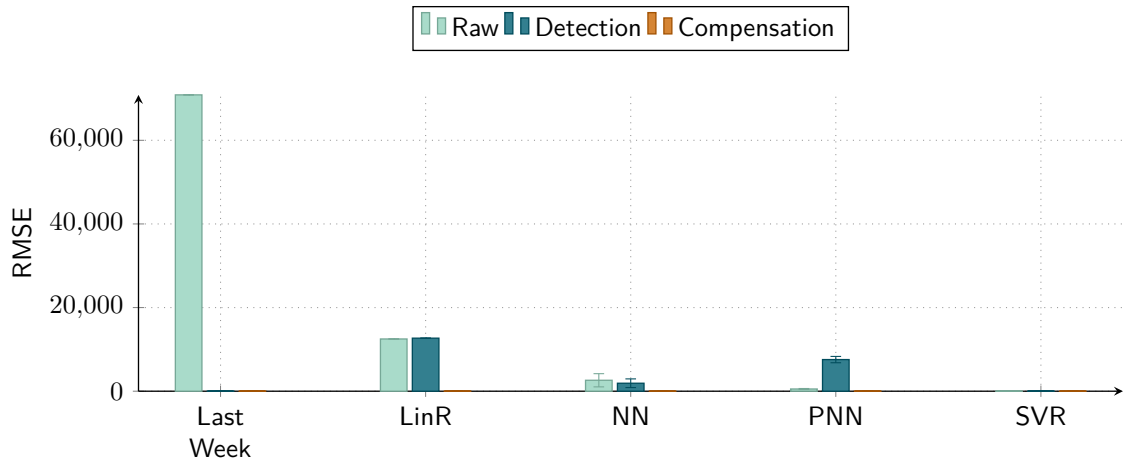
For the technical faults, all considered forecasting methods except the Last Week Forecast and the NN have the lowest RMSE when using the compensation strategy. The Last Week Forecast achieves its lowest RMSE using the detection strategy and the NN with the

raw strategy. Moreover, the difference between the RMSE when using the compensation strategy and the RMSE using the second best strategy is largest for the SVR and rather small for the LinR and PNN. Furthermore, we see the largest difference between the RMSEs in the use of the raw, detection, and compensation strategies for the Last Week Forecast, followed by the SVR. Compared to the anomaly-free baseline strategy, the RMSE of all forecasting methods, especially the SVR and the Last Week Forecast, is also noticeably greater for all three strategies. Considering the actual accuracy, the LinR, the NN, and the PNN form the group of forecasting methods that achieves the lowest RMSE, followed by the SVR and the Last Week Forecast.

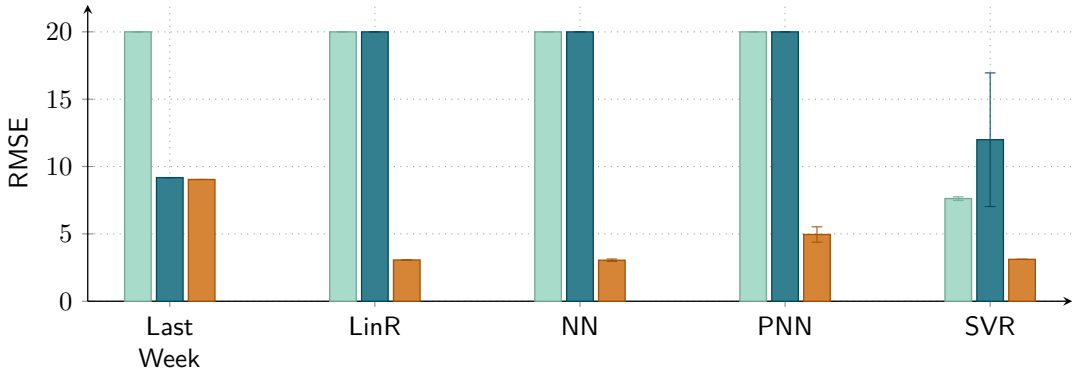
For the unusual consumption, all considered forecasting methods consistently achieve the lowest RMSE using the compensation strategy. The difference in the RMSE between using the compensation strategy and using the second best strategy is big for the SVR, LinR, and NN and small for the PNN and Last Week Forecast. Moreover, we observe the largest differences between the RMSEs for the use of the raw, detection, and compensation strategies for the NN and the LinR whereas the difference is small for the PNN and the Last Week Forecast. In comparison to the anomaly-free baseline strategy, the RMSE of all forecasting methods except the LinR and the NN is only slightly larger but more clearly larger for the LinR and the NN. With regard to their actual accuracy, all forecasting methods achieve a similarly low RMSE except for the Last Week Forecast that obtains the considerably highest RMSE.

Data With Labeled Anomalies For the comparison of all strategies, we also apply them to the data with labeled anomalies. Figure 6.12 shows the resulting RMSE for the labeled technical faults. For each evaluated forecasting method, the bars present the average RMSE for the raw strategy, the detection strategy, and the compensation strategy. The error bars indicate the best and the worst observed RMSE.

We observe that all considered forecasting methods consistently achieve the lowest RMSE using the compensation strategy. However, all forecasting methods except the SVR achieve highly different results using the raw, detection, and compensation strategies. Using the raw strategy leads to an exceptionally high RMSE for the Last Week Forecast, using the raw and detection strategies to very high RMSEs for the LinR, the NN, and PNN. For this reason, the difference in the RMSE between using the compensation strategy and using the second best strategy is very large for the LinR, the NN, and PNN. Moreover, comparing the forecasting method regarding their actual accuracy, the LinR, the NN, and the SVR achieve the lowest RMSEs using the compensation strategy and the SVR low RMSEs across all three strategies.



(a) The five forecasting methods with untruncated y-axis.



(b) The five forecasting methods with y-axis truncated at a RMSE of 20.

Figure 6.12 The RMSE of the five forecasting methods applied to the data with labeled technical faults. For each method, the bars indicate the average RMSE for the raw strategy, detection strategy, and compensation strategy. The error bars show the best and the worst observed RMSE.

6.4 Discussion

In this section, we first discuss the results from the evaluation of the proposed strategies for managing anomalies in energy time series forecasting, before reviewing the evaluation regarding its limitations and insights.

When first comparing the raw and the robust strategies, we observe for the technical faults in the data with synthetic anomalies that the RMSEss of the forecasting methods from the raw strategy deviate more from the RMSEs of the anomaly-free baseline strategy than the RMSEs of the forecasting methods from the robust strategy. We assume that the considered technical faults tend to have a large impact on the forecast accuracy, which the forecasting methods from the robust strategy are better able to handle, confirming their perception as robust. Nevertheless, only one forecasting method from the robust strategy, namely the RF Regressor, achieves an RMSE that

is comparable to the three best-performing forecasting methods from the raw strategy. We make a similar observation for the unusual consumption although the RMSEs of the forecasting methods from the raw strategy deviate less from the RMSEs of the anomaly-free baseline strategy and two forecasting methods from the robust strategy perform similarly well as the five best-performing methods from the raw strategy. In contrast, the results are mixed for the data with labeled anomalies: We observe an extraordinary high RMSE for several forecasting methods but a low RMSE for the SVR and the Median Weekday Forecast. We again suppose that the small amount of data and the contained technical faults may cause this observation. Moreover, we see that selecting different or evaluating more robust forecasting methods could influence the perception of this strategy. Given these observations, one could further investigate both strategies with additional methods or data to verify our decision not to consider the robust strategy further and focus on the raw strategy in the final comparison.

Considering the best anomaly detection for the detection strategy, the results show that the detection strategy with supervised anomaly detection and the detection strategy with unsupervised anomaly detection perform very similarly for the technical faults in the data with synthetic anomalies. Moreover, for the unusual consumption, the detection strategy using unsupervised anomaly detection results in a slightly better forecast accuracy. From this observation, one could infer that the type of anomaly detection does not really influence the detection strategy. However, the results for the data with labeled anomalies show that the selected anomaly detection can strongly influence the forecast accuracy because we observe noticeably high RMSEs for several applied forecasting methods. One reason for this observation could be that the used data with labeled anomalies have a small size that might interact with the contained labeled technical faults and their distribution in the data. Although the forecasting methods that perform best on this data tend to use the detection strategy with unsupervised anomaly detection, there is no clear recommendation which anomaly detection is advantageous for this data. Therefore, one should examine these data with additional detection methods or with the applied methods other similar data to come to a clear recommendation. Given the resulting recommendation, it could be reasonable to consider the detection strategy with supervised anomaly detection for the data with labeled anomalies in the final comparison.

With regard to the best anomaly detection for the compensation strategy, the compensation strategy with unsupervised anomaly detection performs slightly better for the technical faults and noticeably better for the unusual consumption in the data with synthetic anomalies compared to the compensation strategy with supervised anomaly detection. However, for the data with labeled anomalies, it is the opposite because the compensation strategy with supervised anomaly detection achieves a higher accuracy for almost all considered forecasting methods. We suppose that the length and the contained technical faults of the data with labeled anomalies might favor supervised anomaly detection since it uses training data to learn the anomalies to be detected. In any case, when comparing supervised and unsupervised anomaly detection, one has to keep in mind that the related methods are applied to different sets of the considered data because of a training set required by supervised anomaly detection methods.

In the final comparison of the considered strategies, we observe that using the compensation strategy yields the lowest RMSE or an RMSE similar to that of the best strategy for most forecasting methods in the case of the technical faults or even for all methods in the case of the unusual consumption in the data with synthetic anomalies. For the data with labeled anomalies, using the compensation strategy yields also the best RMSE for all forecasting methods. Nonetheless, we also observe that forecasts using the compensation strategy deviate less from the forecasts using the anomaly-free baseline strategy for the unusual consumption compared to the technical faults, probably due to their finer characteristics. Therefore, the advantage of selecting the best strategy depends on the anomalies contained in the used data.

Nevertheless, we note that these results are associated with certain limitations. For the evaluation, we apply forecasting methods with certain parameters and a mostly well-performing but basic set of features. It may be interesting to investigate comprehensively the influence of other parameters and additional features on the performance of the different strategies. Similarly, the used forecasting methods could additionally get input features based on exogenous data, such as weather data, that is known to improve forecasts. Moreover, all reported results are based on the applied methods, the selected evaluation criteria, and the used data. Therefore, future work could evaluate the proposed strategies with further forecasting methods, evaluation criteria, and data. For the detection strategy, the application and evaluation of other detection methods could additionally be of interest. The compensation strategy in particular could also be evaluated with different detection, compensation, and forecasting methods, which additionally implies a potential investigation of the interaction of these methods. Regarding the evaluation criteria, forecasting-independent criteria such as the carbon dioxide emissions associated with the computations required for the different strategies could add a forecasting-independent and environmentally sensitive criterion to the evaluation.

Overall, we conclude from the performed evaluation that the compensation strategy is generally beneficial as it allows for better or at least similar forecasting results as the other evaluated strategies when the input data contains anomalies. This accuracy improvement is higher for unusual consumption than for technical faults. By favoring accurate forecasts, the compensation strategy provides a means for appropriately managing anomalies in energy time series forecasting.

6.5 Contribution and Future Work

In the present chapter, we investigate how an anomaly management can account for anomalies in energy time series forecasting, thus answering research question [RQ4]. For this, we evaluate the proposed general strategies for managing anomalies in energy time series forecasting using a representative selection of forecasting methods, a real-world data set containing inserted synthetic anomalies, and a real-world data set containing labeled anomalies. The evaluation comprises four steps and starts by comparing the proposed raw and robust strategies as they both use the same input data but different

forecasting methods. It continues with determining the best anomaly detection for the proposed detection strategy and then for the proposed compensation strategy. The evaluation ends with all proposed strategies being compared.

With this approach, the present chapter provides the following contributions:

- We propose four general strategies for managing anomalies in energy time series forecasting, namely the raw, robust, detection, and compensation strategy.
- We find that most forecasting methods of the raw strategy provide a lower or at least similar accuracy as the forecasting methods of the robust strategy.
- We determine that the detection strategy achieves the highest accuracy using unsupervised anomaly detection for both used data sets. Moreover, we find that the compensation strategy obtains the highest accuracy using the unsupervised anomaly detection for the data with synthetic anomalies and the supervised anomaly detection for the data with labeled anomalies.
- We show that the proposed compensation strategy is generally beneficial as it allows for better or at least similar prediction results as the other evaluated strategies when the input data contains anomalies.

Given the proposed strategies for managing anomalies in energy time series, future work could address several follow-up questions. For example, future work could verify the results by applying other data, anomaly detection methods, and anomaly compensation methods. Especially a further investigation of the exclusion of the robust strategy and the best anomaly detection for the detection strategy and the labeled data could be of interest. Similarly, future work could evaluate the proposed strategies with further forecasting methods, which also includes the opportunity to examine their suitability for the different proposed strategies and to compare them with methods applied in other strategies. Furthermore, future work could integrate the proposed strategies into existing approaches for automated machine learning to include them in the optimization problem of finding the best forecast for a given data set.

7 Discussion

In addition to the aspects covered in the discussion of the proposed methods for modeling, detecting, compensating, and managing anomalies in Sections 3.5, 4.4, 5.4, and 6.4, the present chapter discusses two common underlying assumptions and the associated limitations, the proposed anomaly management, and the general evaluation approach taken in the present thesis.

The present thesis bases the previously presented anomaly modeling, anomaly detection, anomaly compensation, and anomaly management on a shared set of assumptions that have implicit limitations and offer potential extensions. The first assumption relates to the considered data. In the present thesis, we use two data sets for the performed evaluations, namely a data set where we insert synthetic anomalies and a data set where we label the contained anomalies. Except for the difference in the anomalies and their length, both data sets include energy time series that exhibit the typical characteristics of multi-seasonality, aggregation-level-dependent predictability, and exogenous influence. However, these energy time series only cover the electrical consumption of single clients or buildings and have a quarter-hourly resolution. For this reason, the presented results are limited to consumption data at a low to medium aggregation level of energy time series with respect to time and space. Therefore, one could validate the results with energy time series representing production and energy time series of other temporal or spatial aggregation levels such as electrical devices. Moreover, it could be insightful to consider further energy time series from the used data sets, other data sets comprising longer energy time series with labeled anomalies, or energy time series containing other forms of energy such as natural gas or heat.

The second assumption refers to the examined anomalies. In the present thesis, we define the associated basic perception of normality with the daily patterns as the smallest element of multi-seasonality. We thereby assume that the anomalies contained therein are very likely to manifest themselves in the larger weekly and seasonal patterns of multi-seasonality. However, this assumption might not hold for all anomalies observed in real-world energy time series so one should investigate observed anomalies in this respect. Alternatively, or additionally, the perception of normality could be extended to weekly and seasonal patterns to cover all patterns of multi-seasonality. Similarly, an adapted perception of normality could also consider the so far disregarded characteristics of energy time series, namely aggregation-level dependent predictability and exogenous

influence. Since the perception of normality used in this thesis leads to a focus on the two anomaly groups technical faults and unusual consumption, a modified perception of normality would likely change the actually examined anomalies. Nevertheless, when keeping the initial perception of normality, it could also be possible to identify further anomalies or groups of anomalies. Furthermore, physical hardware installed in real-world laboratories of the energy system could be used to flexibly change the applied perception of normality, to characterize new anomalies, to verify previously identified anomalies, or to determine the likely cause of observed anomalies.

Regarding the anomaly management proposed in this thesis, we choose forecasting as an exemplary subsequent application. In making this decision, we assume that forecasting is a relevant and representative application of an anomaly management. While we definitely find evidence for the relevance of forecasting, we do not thoroughly examine whether it is representative for generally possible subsequent applications such as load profiling or load management. These applications obviously could require other evaluation criteria than the accuracy used for forecasting and additionally and implicitly have different requirements for an anomaly management. For example, forecasting itself already imposes other requirements if it is considered as online forecasting. In addition to taking into account the relationship between the considered application and anomaly management, it could be interesting to examine the interaction between the elements of the proposed anomaly management and to possibly develop an integrated optimization of all elements. In similar way, one could also investigate whether the proposed anomaly management benefits from using generated synthetic energy time series or integrating anomaly generation into synthetic time series generation.

Overall, we perform a comprehensive evaluation of the proposed data-driven methods for managing anomalies. By using one data set for the empirical identification and modeling of anomalies and another one for the evaluation of the proposed data-driven methods, we separate the modeling of the anomalies from the experiments with those anomalies. Since we also evaluate the proposed data-driven methods with the data used for the identification and modeling of anomalies, we additionally implicitly evaluate the modeled anomalies in the performed evaluations. As the results from both data sets are largely consistent in these evaluations, we conclude that our observation and modeling of the anomalies contained in the one data set are sufficiently accurate. On a more general level, the evaluation results also show that the proposed data-driven methods perform well in the example of energy time series forecasting, emphasizing the importance of anomaly management. Moreover, the evaluation shows the practical suitability of the proposed anomaly management by using real-world data and real-world anomalies. With the presented evaluations and the proposed methods, the present dissertation additionally prepares a first step toward managing anomalies in fully automated smart grid settings, where, for example, the best general strategy for a particular situation should be selected in an automated manner.

8 Summary and Outlook

With the progressing implementation of the smart grid, more and more smart meters record power or energy consumption and generation as time series. The increasing availability of these recorded energy time series enables the goal of the automated operation of smart grid applications such as load analysis, load forecasting, and load management. However, to perform well, these applications usually require clean data that describes the typical behavior of the underlying system well. Unfortunately, recorded energy time series are usually not clean but contain anomalies, i. e., patterns that deviate from what is considered normal. Since anomalies thus potentially contain data points or patterns that represent false or misleading information, they can be problematic for any analysis of this data performed by smart grid applications. Therefore, the present thesis proposes data-driven methods for managing anomalies in energy time series. It introduces an anomaly management whose characteristics correspond to steps in a sequential pipeline, namely anomaly detection, anomaly compensation, and a subsequent application. Using forecasting as an exemplary subsequent application and real-world data with inserted synthetic anomalies and labeled anomalies, this thesis answers four research questions along that pipeline for managing anomalies in energy time series.

The first research question *How can anomalies in energy time series be modeled and generated to improve anomaly detection?* is answered in Chapter 3. We introduce a method that is capable of generating four types of synthetic anomalies derived from real-world anomalies that can be inserted in arbitrary quantity and at random points of time into an arbitrary energy and power time series. To develop this generation method, we identify commonly occurring anomaly types in real-world energy and power time series and use the resulting formal model of each type as basis for the generation method. By being capable of generating realistic synthetic anomalies on request, the introduced generation method assures the quality of to-be-developed anomaly detection methods.

The second research question *How can anomaly detection methods for energy time series be enhanced?* is addressed in Chapter 4. We introduce a novel approach that generally enhances anomaly detection methods for energy time series by taking advantage of their latent space representation. According to this approach, a previously trained conditional Invertible Neural Network or conditional Variational Autoencoder creates the latent space representation of an input time series containing anomalies. The resulting latent space data representation then serves as an input for an arbitrary existing supervised

or unsupervised anomaly detection method, which generally have a higher detection performance using this latent space representation.

The third research question *How can anomalies detected in energy time series be compensated?* is investigated in Chapter 5. We propose the Copy-Paste Imputation (CPI) method for time series containing energy measurements that copies blocks of data with similar characteristics into existing gaps. By copying blocks of matching patterns, the CPI method realistically imputes detected anomalies that have been labeled as missing values. Moreover, the CPI method guarantees that the total recorded energy remains unchanged during the imputation.

The fourth research question *How can an anomaly management account for anomalies in energy time series forecasting?* is answered in Chapter 6. We propose four general strategies for managing anomalies in energy time series forecasting that build on typically used strategies. After describing these strategies, we evaluate them with a representative selection of forecasting methods. For the strategies using anomaly detection, we find that using an unsupervised anomaly detection tends to be advantageous. Based on a comparison of all considered strategies, we determine that the compensation strategy, which detects and compensates anomalies in the input data before applying a forecasting method, is the most beneficial strategy when the input data contains anomalies.

Based on the answers to these four research questions, the anomaly management presented in this thesis exhibits four characteristics. First, the presented anomaly management is guided by well-defined anomalies derived from real-world energy time series. These anomalies serve as a basis for generating synthetic anomalies in energy time series to promote the development of powerful anomaly detection methods. Second, the presented anomaly management applies an anomaly detection approach to energy time series that is capable of providing a high anomaly detection performance. Third, the presented anomaly management also compensates detected anomalies in energy time series realistically by considering the characteristics of the respective data. Fourth, the proposed anomaly management applies and evaluates general anomaly management strategies in view of the subsequent forecasting that uses this data.

Given these characteristics of the presented anomaly management for energy time series, there are several possible further research directions. In the following, we focus on three research directions that complement the specific remarks on future work regarding the proposed methods for modeling, detecting, compensating, and managing anomalies in Sections 3.6, 4.5, 5.5, and 6.5.

The first research direction concerns the anomaly management itself. To detail the proposed anomaly management with regard to the considered application, one could comprehensively gather the requirements of possible applications and, if necessary, adapt the anomaly management and the general strategies accordingly. Moreover, it could be interesting to examine the interaction between anomaly modeling, anomaly detection, anomaly compensation, and the different anomaly management strategies since it could be useful to find a common optimum. Furthermore, synthetic energy time series could

support the anomaly management by, for example, providing additional energy time series or synthetic energy time series with already inserted synthetic anomalies.

The second research direction is about the influence of the used data on the anomaly management. One could investigate how using further energy time series from the selected data sets or longer labeled energy time series influence the performed evaluations of the anomaly management. Similarly, production energy time series, energy time series with low or high temporal and spatial aggregation levels, and energy time series of other forms of energy such as natural gas or heat could be interesting to analyze regarding the performed evaluations.

The third research direction considers the role of the considered anomalies. To verify the results and one underlying assumption, one could investigate whether the used anomalies in daily patterns affect the larger weekly and seasonal patterns. Moreover, it could be of interest to extend the perception of normality and to include all characteristics of energy time series, i. e., aggregation-level dependent predictability, exogenous influence, and all patterns of multi-seasonality. Furthermore, physical hardware in real-world laboratories of the energy system could be used to investigate anomalies and their role even more comprehensively. The laboratory environment could, for example, allow to flexibly adapt the applied perception of normality, to identify new anomalies, to verify the characteristics of previously identified anomalies, and to find the probable cause of observed anomalies.

Appendix

A Modeling Anomalies in Energy Time Series

A.1 Statistics of Identified Anomalies

Table A.1 Overview of the 50 one-year time series from the selected smart meters that are used to label the four identified types of anomalies. For each time series, the overall average power and energy consumption as well as the number, minimum length, maximum length, and average power and energy consumption of the labeled anomalies of the types 1 and 2 are reported.

Time	Overall		Type 1					Type 2				
series	kW	kWh	#	Min	Max	kW	kWh	#	Min	Max	kW	kWh
1	11.8	36136.4	8	2	208	16.9	9839.4	3	2	3	15.5	36334.9
2	28.6	6308.5	7	2	208	22.9	1590.8	3	2	3	36.8	5310.9
3	121.3	16508.4	8	2	208	144.3	3905.3	14	2	27	144.0	9231.7
4	35.7	51037.7	7	2	208	34.6	15653.9	3	2	3	36.9	50612.8
5	91.6	67761.1	9	2	208	-13389.9	18031.2	15	2	27	98.5	34858.4
6	1.7	58368.5	8	2	8931	0.9	12781.4	0	-	-	-	-
7	301.7	28417.3	7	2	208	250.8	8649.4	3	2	3	345.5	27697.7
8	58.1	1247.5	9	2	208	11662.8	375.7	13	2	27	124.6	688.8
9	15.5	15227.5	7	2	208	12.3	4678.7	3	2	332	16.8	15105.7
10	4.4	65001.2	9	2	208	-2523.3	12794.6	7	2	27	0.8	22416.6
11	69.2	29795.1	7	2	208	64.4	9191.6	3	2	332	74.7	29882.1
12	11.9	49030.3	7	2	208	14.8	15179.6	3	2	332	9.0	49333.9
13	27.8	7366.5	8	2	208	63.3	2381.6	9	2	270	36.5	7133.5
14	12.0	25829.4	9	2	208	20.2	8235.4	5	3	270	11.3	25985.9
15	13.9	62656.2	7	2	208	16.5	19379.0	4	2	5	17.8	63026.7
16	0.7	6757.4	7	2	208	0.6	2091.7	4	2	6	0.7	6798.9
17	56.6	72354.2	7	2	208	66.3	22360.6	4	2	5	72.3	72769.3
18	19.6	39898.6	7	2	208	22.1	12328.3	2	3	16	23.1	40021.8
19	0.3	8018.0	7	2	208	0.4	2480.6	2	3	16	0.2	8058.0
20	2.3	34096.7	7	2	208	3.2	10546.5	2	3	16	2.2	34301.2
21	1.7	173336.8	7	2	208	1.1	53519.8	3	3	16	0.4	171582.1
22	34.0	24674.7	7	2	208	36.8	7574.9	2	3	16	44.0	24409.0
23	1.0	8222.8	7	2	208	1.3	2135.3	2	3	16	1.1	8763.0
24	1.1	79733.7	7	2	208	0.4	24468.9	0	-	-	-	-
25	1.7	25918.2	7	2	208	1.7	7405.4	2	3	16	2.0	26573.7
26	25.3	47407.2	7	2	208	28.6	14622.6	2	3	16	30.3	47470.3
27	28.2	151172.1	12	2	6	28.8	85601.1	0	-	-	-	-
28	7.1	58558.8	5	2	9	8.3	13350.9	2	10	17	6.1	58462.0
29	13.1	38328.4	5	2	9	18.6	8748.0	1	17	17	6.5	38388.1
30	0.7	570.7	18	2	9	0.5	219.4	1	17	17	0.7	575.1
31	0.2	183.4	16	2	9	0.3	67.0	1	17	17	0.2	183.6
32	83.3	91543.3	5	2	9	99.4	20869.8	1	17	17	113.4	91660.3
33	1.0	56481.2	5	2	9	0.9	12980.3	1	17	17	1.2	56629.4
34	1.1	163877.9	6	2	1363	1.9	32573.8	2	1731	6452	0.6	171591.2
35	2.0	170468.5	5	2	9	3.4	39103.9	2	17	3535	2.0	170194.1
36	54.6	455610.1	7	2	9	64.9	126438.8	2	17	4425	13.4	461428.7
37	0.4	224231.4	5	2	9	0.3	51062.6	3	17	2290	0.3	224305.7
38	1.1	52122.9	5	2	9	1.6	11861.1	2	10	17	1.9	52123.0
39	0.6	7139.3	5	2	9	0.6	1627.7	2	7	17	0.9	7086.7
40	3.1	11695.8	5	2	9	4.6	2685.1	2	10	17	1.9	11667.7
41	19.3	31659.7	6	2	386	36.1	6086.3	5	17	3516	12.4	32017.5
42	0.7	103539.5	6	2	452	0.7	19983.6	1	17	17	0.8	105146.9
43	4.6	19312.0	5	2	9	6.8	4425.2	2	2	17	5.6	19260.8
44	96.3	93851.6	5	2	9	90.1	21437.6	1	17	17	166.7	94007.3
45	85.4	45205.9	5	2	9	67.0	10358.4	1	17	17	193.9	45306.2
46	18.8	17973.7	6	2	488	33.1	3497.5	2	727	1957	33.6	18307.7
47	0.2	199217.2	5	2	9	0.1	45346.7	0	-	-	-	-
48	1.6	43087.4	6	2	9	3.3	9424.6	4	8	17	2.4	40338.4
49	1.2	8411.2	7	2	9	1.2	2724.2	3	8	17	1.1	8273.2
50	56.4	28194.2	6	2	9	0.0	6050.3	5	8	14869	52.8	28081.2

Table A.2 Overview of the 50 one-year time series from the selected smart meters that are used to label the four identified types of anomalies. For each time series, the overall average power and energy consumption as well as the number, minimum length, maximum length, and average power and energy consumption of the labeled anomalies of the types 3 and 4 are reported. Note that anomalies of types 3 and 4 always have a length of one and that these types comprise two cases.

Time	Overall		Type 3			Type 4		
series	kW	kWh	#	kW	kWh	#	kW	kWh
1	11.8	36136.4	0	-	-	1	36779.9	27.1
2	28.6	6308.5	1	6740.1	14.0	1	6776.9	62.7
3	121.3	16508.4	2	6255.1	-10008059.8	4	10786.9	5006955.5
4	35.7	51037.7	1	53132.4	15.5	1	53290.0	80.7
5	91.6	67761.1	2	12797.3	-25481.8	3	8718.7	2740483.8
6	1.7	58368.5	0	-	-	0	-	-
7	301.7	28417.3	1	30783.5	178.1	1	30923.4	480.5
8	58.1	1247.5	2	490.4	-23582.7	2	331.9	456993.9
9	15.5	15227.5	1	16234.2	8.9	1	16311.2	37.1
10	4.4	65001.2	0	-	-	0	-	-
11	69.2	29795.1	1	30572.6	37.0	1	30622.0	144.9
12	11.9	49030.3	1	49416.9	4.6	2	49375.3	29.8
13	27.8	7366.5	1	8021.2	21.2	1	8077.8	50.2
14	12.0	25829.4	1	26075.4	5.2	1	26081.9	26.3
15	13.9	62656.2	1	63571.0	8.0	2	63322.9	25.0
16	0.7	6757.4	1	6822.1	0.3	2	6811.0	1.4
17	56.6	72354.2	1	73590.1	33.1	1	73651.3	102.7
18	19.6	39898.6	1	40593.7	8.7	2	40667.8	63.4
19	0.3	8018.0	1	8089.7	0.1	1	8091.9	0.5
20	2.3	34096.7	1	34398.8	1.2	1	34412.7	8.1
21	1.7	173336.8	1	178587.3	0.7	2	174707.7	3.9
22	34.0	24674.7	1	26082.4	21.5	1	26165.2	53.7
23	1.0	8222.8	1	9449.6	0.1	1	9474.5	0.3
24	1.1	79733.7	0	-	-	1	83031.0	4.9
25	1.7	25918.2	1	28404.9	0.3	1	28593.6	2.4
26	25.3	47407.2	1	48514.9	13.6	1	48585.8	53.9
27	28.2	151172.1	0	-	-	1	235908.3	41.3
28	7.1	58558.8	0	-	-	3	58827.1	279.6
29	13.1	38328.4	0	-	-	1	38717.3	22.4
30	0.7	570.7	0	-	-	0	-	-
31	0.2	183.4	0	-	-	1	187.4	0.5
32	83.3	91543.3	0	-	-	1	92873.7	127.0
33	1.0	56481.2	0	-	-	1	59574.2	2.3
34	1.1	163877.9	0	-	-	0	-	-
35	2.0	170468.5	0	-	-	2	176801.9	9.2
36	54.6	455610.1	0	-	-	0	-	-
37	0.4	224231.4	0	-	-	1	225476.6	1.5
38	1.1	52122.9	0	-	-	2	52182.2	3.2
39	0.6	7139.3	0	-	-	2	7176.6	1.0
40	3.1	11695.8	0	-	-	2	11754.8	165.1
41	19.3	31659.7	0	-	-	0	-	-
42	0.7	103539.5	0	-	-	2	105784.1	195.0
43	4.6	19312.0	0	-	-	1	19401.5	25.7
44	96.3	93851.6	0	-	-	1	96294.1	188.9
45	85.4	45205.9	0	-	-	1	47441.0	203.2
46	18.8	17973.7	0	-	-	1	18924.6	19.9
47	0.2	199217.2	0	-	-	1	199777.8	0.9
48	1.6	43087.4	0	-	-	1	44925.4	5.2
49	1.2	8411.2	0	-	-	1	8594.0	2.6
50	56.4	28194.2	0	-	-	0	-	-

A.2 Used Parameters

Table A.3 Overview of the number, minimum length, maximum length, r_{min}, r_{max}, and k used as parameters to generate synthetic anomalies for the evaluated 50 one-year power time series from the selected smart meters using the t-SNE and the discriminative method. Note that anomalies of types 3 and 4 always have a length of one and comprise two cases.

Time		Type 1			Type 2			Type 3			Type 4		
series	k	#	Min	Max	#	Min	Max	#	r_{min}	r_{max}	#	r_{min}	r_{max}
1	35787	8	3	96	3	2	3	0	-	-	1	1.15	8.1
2	5730	7	3	96	3	2	3	1	0.61	1.62	1	1.15	8.1
3	17649	8	3	96	14	2	27	2	-	-	4	11.01	13
4	48127	7	3	96	3	2	3	1	0.61	1.62	1	1.15	8.1
5	68477	9	3	96	15	2	27	2	-	-	3	11.01	13
6	80207	8	3	96	0	-	-	0	-	-	0	1.15	8.1
7	25239	7	3	96	3	2	3	1	0.61	1.62	1	1.15	8.1
8	731	9	3	96	13	2	27	2	-	-	2	11.01	13
9	14104	7	3	96	3	2	48	1	0.61	1.62	1	1.15	8.1
10	49387	9	3	96	7	2	27	0	-	-	0	-	-
11	29056	7	3	96	3	2	48	1	0.61	1.62	1	1.15	8.1
12	49172	7	3	96	3	2	48	1	0.61	1.62	2	1.15	8.1
13	6393	8	3	96	9	2	48	1	0.61	1.62	1	1.15	8.1
14	25862	9	3	96	5	3	48	1	0.61	1.62	1	1.15	8.1
15	62272	7	3	96	4	2	5	1	0.61	1.62	2	1.15	8.1
16	6764	7	3	96	4	2	6	1	0.61	1.62	2	1.15	8.1
17	71565	7	3	96	4	2	5	1	0.61	1.62	1	1.15	8.1
18	39421	7	3	96	2	3	16	1	0.61	1.62	2	11.01	13
19	8020	7	3	96	2	3	16	1	-	-	1	1.15	8.1
20	34042	7	3	96	2	3	16	1	-	-	1	1.15	8.1
21	168614	7	3	96	3	3	16	1	-	-	2	11.01	13
22	23011	7	3	96	2	3	16	1	0.61	1.62	1	1.15	8.1
23	2653	7	3	96	2	3	16	1	-	-	1	11.01	13
24	75229	7	3	96	0	-	-	0	-	-	1	11.01	13
25	17937	7	3	96	2	3	16	1	-	-	1	1.15	8.1
26	46301	7	3	96	2	3	16	1	0.61	1.62	1	1.15	8.1
27	28114	12	3	6	0	-	-	0	-	-	1	1.15	8.1
28	58016	5	3	9	2	10	17	0	-	-	3	11.01	13
29	37605	5	3	9	1	17	17	0	-	-	1	1.15	8.1
30	509	18	3	9	1	17	17	0	-	-	0	-	-
31	177	16	3	9	1	17	17	0	-	-	1	1.15	8.1
32	89750	5	3	9	1	17	17	0	-	-	1	1.15	8.1
33	51978	5	3	9	1	17	17	0	-	-	1	1.15	8.1
34	165403	6	3	96	2	44	48	0	-	-	0	-	-
35	161244	5	3	9	2	17	48	0	-	-	2	11.01	13
36	431796	7	3	9	2	17	48	0	-	-	0	-	-
37	222477	5	3	9	3	17	48	0	-	-	1	1.15	8.1
38	52017	5	3	9	2	10	17	0	-	-	2	11.01	13
39	7001	5	3	9	2	7	17	0	-	-	2	1.15	8.1
40	11188	5	3	9	2	10	17	0	-	-	2	11.01	13
41	31823	6	3	96	5	17	48	0	-	-	0	-	-
42	102079	6	3	96	1	17	17	0	-	-	2	11.01	13
43	18806	5	3	9	2	2	17	0	-	-	1	11.01	13
44	90338	5	3	9	1	17	17	0	-	-	1	1.15	8.1
45	42124	5	3	9	1	17	17	0	-	-	1	1.15	8.1
46	17234	6	3	96	2	44	48	0	-	-	1	1.15	8.1
47	198393	5	3	9	0	-	-	0	-	-	1	11.01	13
48	32838	6	3	9	4	8	17	0	-	-	1	11.01	13
49	8159	7	3	9	3	8	17	0	-	-	1	1.15	8.1
50	26747	6	3	9	5	8	48	0	-	-	0	-	-

Table A.4 Overview of the number, minimum length, maximum length, r_{min}, r_{max}, and k used as parameters to generate synthetic anomalies for the evaluated 50 one-year power time series from the selected smart meters regarding the training of the evaluated supervised anomaly detection methods. Note that anomalies of types 3 and 4 always have a length of one and comprise two cases.

Time		Type 1			Type 2			Type 3			Type 4		
series	k	#	Min	Max	#	Min	Max	#	r_{min}	r_{max}	#	r_{min}	r_{max}
1	0	16	3	96	6	2	3	0	-	-	2	1.15	8.1
2	0	14	3	96	6	2	3	2	0.61	1.62	2	1.15	8.1
3	0	16	3	96	28	2	27	4	-	-	8	11.01	13
4	0	14	3	96	6	2	3	2	0.61	1.62	2	1.15	8.1
5	0	18	3	96	30	2	27	4	-	-	6	11.01	13
6	0	16	3	96	0	-	-	0	-	-	0	-	-
7	0	14	3	96	6	2	3	2	0.61	1.62	2	1.15	8.1
8	0	18	3	96	26	2	27	4	-	-	4	11.01	13
9	0	14	3	96	6	2	48	2	0.61	1.62	2	1.15	8.1
10	0	18	3	96	14	2	27	0	-	-	0	-	-
11	0	14	3	96	6	2	48	2	0.61	1.62	2	1.15	8.1
12	0	14	3	96	6	2	48	2	0.61	1.62	4	1.15	8.1
13	0	16	3	96	18	2	48	2	0.61	1.62	2	1.15	8.1
14	0	18	3	96	10	3	48	2	0.61	1.62	2	1.15	8.1
15	0	14	3	96	8	2	5	2	0.61	1.62	4	1.15	8.1
16	0	14	3	96	8	2	6	2	0.61	1.62	4	1.15	8.1
17	0	14	3	96	8	2	5	2	0.61	1.62	2	1.15	8.1
18	0	14	3	96	4	3	16	2	0.61	1.62	4	11.01	13
19	0	14	3	96	4	3	16	2	-	-	2	1.15	8.1
20	0	14	3	96	4	3	16	2	-	-	2	1.15	8.1
21	0	14	3	96	6	3	16	2	-	-	4	11.01	13
22	0	14	3	96	4	3	16	2	0.61	1.62	2	1.15	8.1
23	0	14	3	96	4	3	16	2	-	-	2	11.01	13
24	0	14	3	96	0	-	-	0	-	-	2	11.01	13
25	0	14	3	96	4	3	16	2	-	-	2	1.15	8.1
26	0	14	3	96	4	3	16	2	0.61	1.62	2	1.15	8.1
27	0	24	3	6	0	-	-	0	-	-	2	1.15	8.1
28	0	10	3	9	4	10	17	0	-	-	6	11.01	13
29	0	10	3	9	2	17	17	0	-	-	2	1.15	8.1
30	0	36	3	9	2	17	17	0	-	-	0	-	-
31	0	32	3	9	2	17	17	0	-	-	2	1.15	8.1
32	0	10	3	9	2	17	17	0	-	-	2	1.15	8.1
33	0	10	3	9	2	17	17	0	-	-	2	1.15	8.1
34	0	12	3	96	4	44	48	0	-	-	0	1.15	8.1
35	0	10	3	9	4	17	48	0	-	-	4	11.01	13
36	0	14	3	9	4	17	48	0	-	-	0	-	-
37	0	10	3	9	6	17	48	0	-	-	2	1.15	8.1
38	0	10	3	9	4	10	17	0	-	-	4	11.01	13
39	0	10	3	9	4	7	17	0	-	-	4	1.15	8.1
40	0	10	3	9	4	10	17	0	-	-	4	11.01	13
41	0	12	3	96	10	17	48	0	-	-	0	-	-
42	0	12	3	96	2	17	17	0	-	-	4	11.01	13
43	0	10	3	9	4	2	17	0	-	-	2	11.01	13
44	0	10	3	9	2	17	17	0	-	-	2	1.15	8.1
45	0	10	3	9	2	17	17	0	-	-	2	1.15	8.1
46	0	12	3	96	4	44	48	0	-	-	2	1.15	8.1
47	0	10	3	9	0	-	-	0	-	-	2	11.01	13
48	0	12	3	9	8	8	17	0	-	-	2	11.01	13
49	0	14	3	9	6	8	17	0	-	-	2	1.15	8.1
50	0	12	3	9	10	8	48	0	-	-	0	-	-

B Detecting Anomalies in Energy Time Series

B.1 Default Hyperparameters

Table B.1 Overview of the hyperparameters, their default values, and the evaluated values of all seven selected supervised anomaly detection methods.

Detection method	Hyperparameter	Default value	Evaluated values
kNN	n_neighbors	5	1, 3, 5, 7, 10
	p	2	1, 2, 3
	weights	uniform	uniform, distance
LR	C	1	0.01, 0.1, 1, 10, 100
	penalty	l2	l1, l2, elasticnet, none
	solver	lbfgs	newton-cg, lbfgs, liblinear, sag, saga
MLP	activation	relu	logistic, tanh, relu
	alpha	0.0001	0.00001, 0.0001, 0.001
	batch_size	auto	10, 11, 12, 13, 14, 15, 16, 32, 64, 128, 200
	hidden_layer_size	(100,)	(25,), (50,), (75,), (100,), (125,), (150,), (25, 25), (50, 50), (75, 75), (100, 100), (125, 125), (150, 150), (25, 25, 25), (50, 50, 50), (75, 75, 75), (100, 100, 100), (125, 125, 125), (150, 150, 150)
NB	*no hyperparameters*		
RF	criterion	gini	gini, entropy
	max_features	auto	sqrt, log2
SVC	C	1	0.01, 0.1, 1, 10, 100
	gamma	scale	scale, auto
	kernel	rbf	linear, sigmoid, rbf
XGBoost	booster	gbtree	gbtree, gblinear, dart
	importance_type	gain	gain, weight, cover, total_gain, total_cover
	reg_lambda	1	0, 0.1, 0.5, 1, 2, 4

B.2 Best-Performing Hyperparameters for Data With Synthetic Anomalies

Table B.2 The best-performing hyperparameters of the k-Nearest Neighbor (kNN) for all data representations for the data with synthetic technical faults.

Data representation	n_neighbors	p	weights
Latent cINN	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance
Latent cVAE	1	2	uniform
	1	2	distance
Scaled	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance
	3	3	distance
Unscaled	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance
	3	3	distance

Table B.3 The best-performing hyperparameters of the kNN for all data representations for the data with synthetic unusual consumption.

Data representation	n_neighbors	p	weights
Latent cINN	1	1	uniform
	1	1	distance
Latent cVAE	10	2	distance
Scaled	5	3	uniform
	5	3	distance
Unscaled	5	3	uniform
	5	3	distance

Table B.4 The best-performing hyperparameters of the Logistic Regression (LogR) for all data representations for the data with synthetic technical faults.

Data representation	C	penalty	solver
Latent cINN	10	none	sag
Latent cVAE	0.01	none	newton-cg
	0.1	none	newton-cg
	1	none	newton-cg
	10	none	newton-cg
	100	none	newton-cg
Scaled	0.1	l2	liblinear
Unscaled	0.01	none	saga
	1	l2	sag
	10	none	sag
	100	l2	saga

Table B.5 The best-performing hyperparameters of the LogR for all data representations for the data with synthetic unusual consumption.

Data representation	C	penalty	solver
Latent cINN	0.01	l2	liblinear
Latent cVAE	100	l1	liblinear
Scaled	0.01	l1	liblinear
	0.01	l1	saga
Unscaled	0.01	l1	liblinear
	0.01	l1	saga
	0.01	l2	lbfgs
	0.01	l2	liblinear
	0.01	l2	sag
	0.01	l2	saga
	0.01	none	lbfgs
	0.01	none	sag
	0.01	none	saga
	0.1	l1	saga
	0.1	l2	lbfgs
	0.1	l2	liblinear
	0.1	l2	sag
	0.1	l2	saga
	0.1	none	lbfgs
	0.1	none	sag
	0.1	none	saga
	1	l1	saga
	1	l2	lbfgs
	1	l2	liblinear
	1	l2	sag
	1	l2	saga
	1	none	lbfgs
	1	none	sag
	1	none	saga
	10	l1	saga
	10	l2	lbfgs
	10	l2	liblinear
	10	l2	sag
	10	l2	saga
	10	none	lbfgs
	10	none	sag
	10	none	saga
	100	l1	saga
	100	l2	lbfgs
	100	l2	liblinear
	100	l2	sag
	100	l2	saga
	100	none	lbfgs
	100	none	sag
	100	none	saga

Table B.6 The best-performing hyperparameters of the Multi-Layer Perceptron (MLP) for all data representations for the data with synthetic technical faults.

Data representation	activation	alpha	batch_size	hidden_layer_size
Latent cINN	relu	0.001	15	(100,)
Latent cVAE	relu	0.0001	14	(75,75)
Scaled	relu	0.0001	14	(125,125)
	relu	0.001	12	(125,125,125)
	relu	0.001	15	(125)
Unscaled	relu	0.001	32	(125)

Table B.7 The best-performing hyperparameters of the MLP for all data representations for the data with synthetic unusual consumption.

Data representation	activation	alpha	batch_size	hidden_layer_size
Latent cINN	logistic	0.001	11	(150,)
Latent cVAE	relu	0.00001	11	(75. 75)
Scaled	relu	0.0001	12	(50,50)
Unscaled	relu	0.00001	64	(100,100,100)

Table B.8 The best-performing hyperparameters of the Random Forest (RF) for all data representations for the data with synthetic technical faults.

Data representation	criterion	max_features
Latent cINN	gini	sqrt
Latent cVAE	entropy	sqrt
Scaled	gini	sqrt
Unscaled	gini	sqrt

Table B.9 The best-performing hyperparameters of the RF for all data representations for the data with synthetic unusual consumption.

Data representation	criterion	max_features
Latent cINN	gini	sqrt
Latent cVAE	gini	sqrt
Scaled	entropy	log2
Unscaled	entropy	log2

Table B.10 The best-performing hyperparameters of the Support Vector Machine for Classification (SVC) for all data representations for the data with synthetic technical faults.

Data representation	C	gamma	kernel
Latent cINN	100	scale	rbf
Latent cVAE	100	scale	rbf
Scaled	0.1	scale	rbf
Unscaled	0.1	scale	rbf

Table B.11 The best-performing hyperparameters of the SVC for all data representations for the data with synthetic unusual consumption.

Data representation	C	gamma	kernel
Latent cINN	10	auto	rbf
	100	auto	rbf
Latent cVAE	1	scale	rbf
Scaled	10	scale	rbf
Unscaled	10	scale	rbf

Table B.12 The best-performing hyperparameters of the XGBoost for all data representations for the data with synthetic technical faults.

Data representation	booster	importance_type	reg_lambda
Latent cINN	gbtree	gain	0.1
	gbtree	weight	0.1
	gbtree	cover	0.1
	gbtree	total_gain	0.1
	gbtree	total_cover	0.1
	dart	gain	0.1
	dart	weight	0.1
	dart	cover	0.1
	dart	total_gain	0.1
	dart	total_cover	0.1
Latent cVAE	gbtree	gain	1
	gbtree	weight	1
	gbtree	cover	1
	gbtree	total_gain	1
	gbtree	total_cover	1
	dart	gain	1
	dart	weight	1
	dart	cover	1
	dart	total_gain	1
	dart	total_cover	1
Scaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0
Unscaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0

Table B.13 The best-performing hyperparameters of the XGBoost for all data representations for the data with synthetic unusual consumption.

Data representation	booster	importance_type	reg_lambda
Latent cINN	gbtree	gain	1
	gbtree	weight	1
	gbtree	cover	1
	gbtree	total_gain	1
	gbtree	total_cover	1
	dart	gain	1
	dart	weight	1
	dart	cover	1
	dart	total_gain	1
	dart	total_cover	1
Latent cVAE	gblinear	gain	0
	gblinear	weight	0
	gblinear	cover	0
	gblinear	total_gain	0
	gblinear	total_cover	0
Scaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0
Unscaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0

B.3 Best-Performing Hyperparameters for Data With Labeled Anomalies

Table B.14 The best-performing hyperparameters of the kNN for all data representations for the data with labeled technical faults.

Data representation	n_neighbors	p	weights
Latent cINN	1	3	uniform
	1	3	distance
Latent cVAE	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance
Scaled	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance
Unscaled	1	2	uniform
	1	2	distance
	1	3	uniform
	1	3	distance

Table B.15 The best-performing hyperparameters of the LogR for all data representations for the data with labeled technical faults.

Data representation	C	penalty	solver
Latent cINN	10	l2	liblinear
Latent cVAE	0.01	none	lbfgs
	0.1	none	lbfgs
	1	none	lbfgs
	10	none	lbfgs
	100	none	lbfgs
Scaled	1	l2	liblinear
Unscaled	10	l2	liblinear

Table B.16 The best-performing hyperparameters of the MLP for all data representations for the data with labeled technical faults. Note that there are 167 hyperparameter combinations for the latent conditional Invertible Neural Network (cINN), 513 for the scaled, and 486 for the unscaled data representations with rank one. Of these, only the 10 combinations that require the least average fitting time are listed for graphical clarity.

Data representation	activation	alpha	batch_size	hidden_layer_size
Latent cINN	relu	0.0001	200	(50,50,50)
	relu	0.001	128	(50,50,50)
	relu	0.001	200	(50,50,50)
	relu	0.0001	128	(75,75)
	relu	0.0001	200	(25,25,25)
	relu	0.00001	128	(50,50,50)
	relu	0.0001	128	(25,25,25)
	relu	0.00001	128	(50,50)
	relu	0.001	128	(75,75)
	relu	0.0001	64	(50,50,50)
Latent cVAE	relu	0.0001	13	(125,125,125)
Scaled	relu	0.00001	128	(75,75,75)
	relu	0.001	200	(50,50,50)
	relu	0.001	200	(75,75,75)
	relu	0.00001	200	(50,50,50)
	relu	0.0001	128	(50,50,50)
	relu	0.001	200	(100,100,100)
	relu	0.00001	200	(50,50)
	relu	0.0001	128	(75,75,75)
	relu	0.001	200	(25,25,25)
	relu	0.0001	128	(25,25,25)
Unscaled	relu	0.00001	128	(75,75,75)
	relu	0.0001	200	(50,50)
	relu	0.001	200	(75,75,75)
	relu	0.001	200	(100,100,100)
	relu	0.001	128	(50,50)
	relu	0.00001	128	(50,50,50)
	relu	0.001	128	(75,75,75)
	relu	0.001	128	(100,100)
	relu	0.001	200	(75,75)
	relu	0.0001	200	(75,75)

Table B.17 The best-performing hyperparameters of the RF for all data representations for the data with labeled technical faults.

Data representation	criterion	max_features
Latent cINN	gini	log2
Latent cVAE	gini	sqrt
	gini	log2
	entropy	sqrt
	entropy	log2
Scaled	gini	log2
Unscaled	gini	sqrt

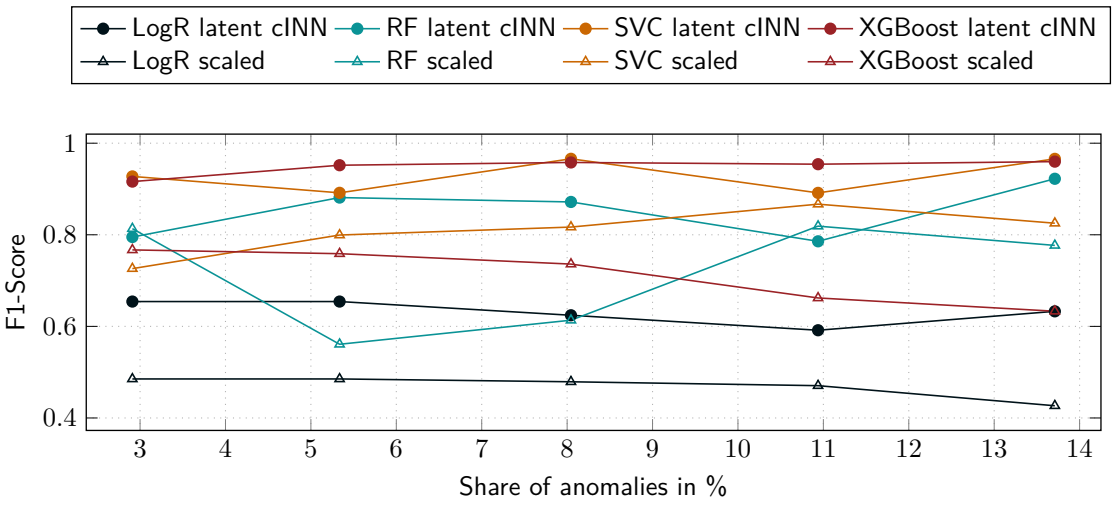
Table B.18 The best-performing hyperparameters of the SVC for all data representations for the data with labeled technical faults.

Data representation	C	gamma	kernel
Latent cINN	1	auto	rbf
	10	auto	rbf
	100	auto	rbf
Latent cVAE	0.1	scale	rbf
	1	scale	rbf
	10	scale	rbf
	100	scale	rbf
Scaled	0.1	auto	rbf
	1	auto	rbf
	10	auto	rbf
	100	auto	rbf
Unscaled	1	auto	sigmoid

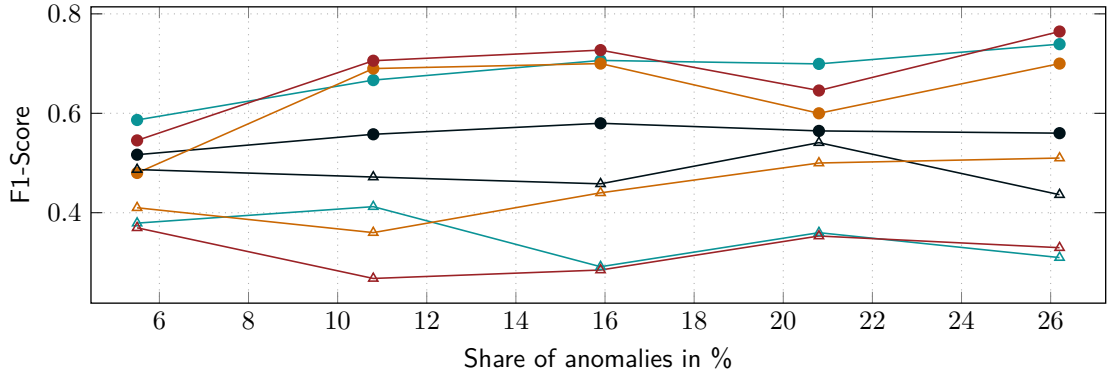
Table B.19 The best-performing hyperparameters of the XGBoost for all data representations for the data with labeled technical faults.

Data representation	booster	importance_type	reg_lambda
Latent cINN	gbtree	gain	0.1
	gbtree	weight	0.1
	gbtree	cover	0.1
	gbtree	total_gain	0.1
	gbtree	total_cover	0.1
	dart	gain	0.1
	dart	weight	0.1
	dart	cover	0.1
	dart	total_gain	0.1
	dart	total_cover	0.1
Latent cVAE	gbtree	gain	2
	gbtree	weight	2
	gbtree	cover	2
	gbtree	total_gain	2
	gbtree	total_cover	2
	dart	gain	2
	dart	weight	2
	dart	cover	2
	dart	total_gain	2
	dart	total_cover	2
Scaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0
Unscaled	gbtree	gain	0
	gbtree	weight	0
	gbtree	cover	0
	gbtree	total_gain	0
	gbtree	total_cover	0
	dart	gain	0
	dart	weight	0
	dart	cover	0
	dart	total_gain	0
	dart	total_cover	0

B.4 Additional Results



(a) Technical faults



(b) Unusual consumption

Figure B.1 The F1-Scores of the four remaining supervised detection methods applied to the data with different shares of synthetic anomalies from technical faults and unusual consumption using the best-performing hyperparameters. For each method, one line each indicates the resulting F1-Score for the cINN latent space and scaled data representations.

C Compensating Anomalies in Energy Time Series

Table C.1 Time series from the data set with inserted missing values used for the evaluation.

Number	Time series
1	MT_007
2	MT_008
3	MT_009
4	MT_010
5	MT_011
6	MT_013
7	MT_014
8	MT_016
9	MT_017
10	MT_018
11	MT_020
12	MT_021
13	MT_022
14	MT_023
15	MT_026
16	MT_029
17	MT_034
18	MT_035
19	MT_036
20	MT_037
21	MT_038
22	MT_040
23	MT_042
24	MT_045
25	MT_046
26	MT_050
27	MT_055
28	MT_057
29	MT_064
30	MT_067
31	MT_068
32	MT_077
33	MT_079
34	MT_084
35	MT_088
36	MT_090
37	MT_093
38	MT_094
39	MT_095
40	MT_096
41	MT_097
42	MT_098
43	MT_099
44	MT_118
45	MT_119
46	MT_123
47	MT_128
48	MT_140
49	MT_141
50	MT_249

Table C.2 Time series from the data set with inserted missing values used for the calibration.

Number	Time series
1	MT_001
2	MT_002
3	MT_004
4	MT_005
5	MT_006

Bibliography

Aggarwal, C. C. (2017). *Outlier Analysis*. 2nd Ed. Springer International Publishing. DOI: 10.1007/978-3-319-47578-3 (cit. on pp. 11–13).

Aguinis, H., R. K. Gottfredson, and H. Joo (2013). *Best-Practice Recommendations for Defining, Identifying, and Handling Outliers*. In: *Organizational Research Methods*, Vol. 16, No. 2, pp. 270–301. DOI: 10.1177/1094428112470848 (cit. on p. 97).

Akouemo, H. N. and R. J. Povinelli (2014). *Time series outlier detection and imputation*. In: *2014 IEEE PES General Meeting | Conference & Exposition*. IEEE. DOI: 10.1109/PESGM.2014.6939802 (cit. on pp. 69, 80).

Akouemo, H. N. and R. J. Povinelli (2016). *Probabilistic anomaly detection in natural gas time series data*. In: *International Journal of Forecasting*, Vol. 32, No. 3, pp. 948–956. DOI: 10.1016/j.ijforecast.2015.06.001 (cit. on p. 97).

Akouemo, H. N. and R. J. Povinelli (2017). *Data Improving in Time Series Using ARX and ANN Models*. In: *IEEE Transactions on Power Systems*, Vol. 32, No. 5, pp. 3352–3359. DOI: 10.1109/TPWRS.2017.2656939 (cit. on pp. 2, 4, 69, 80, 97).

Alahakoon, D. and X. Yu (2016). *Smart Electricity Meter Data Intelligence for Future Energy Systems: A Survey*. In: *IEEE Transactions on Industrial Informatics*, Vol. 12, No. 1, pp. 425–436. DOI: 10.1109/TII.2015.2414355 (cit. on p. 1).

Alquthami, T., A. AlAmoudi, A. M. Alsubaie, A. B. Jaber, N. Alshlwan, M. Anwar, and S. Al Husaien (2020). *Analytics framework for optimal smart meters data processing*. In: *Electrical Engineering*, Vol. 102, No. 3, pp. 1241–1251. DOI: 10.1007/s00202-020-00949-0 (cit. on pp. 1, 69).

Ang, Y., Y. Qian, and S. Gao (2020). *Factory Energy Data Imputation by Nearest Neighbor Search with Clustering*. In: *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA 2020)*. IEEE, pp. 302–307. DOI: 10.1109/AEECA49918.2020.9213497 (cit. on p. 70).

Ardizzone, L., C. Lüth, J. Kruse, C. Rother, and U. Köthe (2019). *Guided Image Generation with Conditional Invertible Neural Networks*. arXiv: 1907.02392 (cit. on pp. 37, 39, 48, 102).

Arning, A., A. Rakesh, and P. Raghavan (1996). *A Linear Method for Deviation Detection in Large Databases*. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD '96)*. Ed. by E. Simoudis, J. Han, and U. Fayyad. Palo Alto, USA: AAAI Press, pp. 164–169. DOI: 10.5555/3001460.3001495 (cit. on p. 11).

Barnett, V. and T. Lewis (1978). *Outliers in Statistical Data*. Chichester New York Brisbane Toronto: John Wiley & Sons (cit. on pp. 11, 12).

Ben Taieb, S. and R. J. Hyndman (2014). *A gradient boosting approach to the Kaggle load forecasting competition*. In: *International Journal of Forecasting*, Vol. 30, No. 2, pp. 382–394. DOI: 10.1016/j.ijforecast.2013.07.005 (cit. on p. 97).

Bokde, N., M. W. Beck, F. Martínez Álvarez, and K. Kulat (2018). *A novel imputation methodology for time series based on pattern sequence forecasting*. In: *Pattern Recognition Letters*, Vol. 116, pp. 88–96. DOI: 10.1016/j.patrec.2018.09.020 (cit. on pp. 69, 81).

Borges, C. E., O. Kamara-Esteban, T. Castillo-Calzadilla, C. M. Andonegui, and A. Alonso-Vicario (2020). *Enhancing the missing data imputation of primary substation load demand records*. In: *Sustainable Energy, Grids and Networks*, Vol. 23, p. 100369. DOI: 10.1016/j.segan.2020.100369 (cit. on pp. 70, 81).

Box, G. E. P. and G. C. Tiao (1975). *Intervention analysis with applications to economic and environmental problems*. In: *Journal of the American Statistical Association*, Vol. 70, No. 349, pp. 70–79. DOI: 10.1080/01621459.1975.10480264 (cit. on p. 97).

Breiman, L. (2001). *Random Forests*. In: *Machine Learning*, Vol. 45, pp. 5–32. DOI: 10.1023/A:1010933404324 (cit. on pp. 51, 108).

Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone (1984). *Classification And Regression Trees*. 1st Ed. New York: Routledge. DOI: 10.1201/9781315139470 (cit. on p. 28).

Breunig, M. M., H.-P. Kriegel, R. T. Ng, and J. Sander (2000). *LOF: Identifying Density-Based Local Outliers*. In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*. ACM, pp. 93–104. DOI: 10.1145/342009.335388 (cit. on pp. 52, 104).

Cao, W., D. Wang, J. Li, H. Zhou, L. Li, and Y. Li (2018). *BRITS: Bidirectional Recurrent Imputation for Time Series*. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., pp. 6775–6785 (cit. on pp. 69, 81).

Capozzoli, A., M. S. Piscitelli, S. Brandi, D. Grassi, and G. Chicco (2018). *Automated load pattern learning and anomaly detection for enhancing energy management in smart buildings*. In: *Energy*, Vol. 157, pp. 336–352. DOI: 10.1016/j.energy.2018.05.127 (cit. on p. 2).

Chakhchoukh, Y., P. Panciatici, and L. Mili (2011). *Electric load forecasting based on statistical robust methods*. In: *IEEE Transactions on Power Systems*, Vol. 26, No. 3, pp. 982–991. DOI: 10.1109/TPWRS.2010.2080325 (cit. on p. 98).

Chandola, V., A. Banerjee, and V. Kumar (2009). *Anomaly Detection: A Survey*. In: *ACM Computing Surveys*, Vol. 41, No. 3, 15:1–15:58. DOI: 10.1145/1541880.1541882 (cit. on pp. 2, 11).

Chang, I., G. C. Tiao, and C. Chen (1988). *Estimation of Time Series Parameters in the Presence of Outliers*. In: *Technometrics*, Vol. 30, No. 2, pp. 193–204. DOI: 10.1080/00401706.1988.10488367 (cit. on p. 97).

Charlton, N. and C. Singleton (2014). *A refined parametric model for short term load forecasting*. In: *International Journal of Forecasting*, Vol. 30, No. 2, pp. 364–368. DOI: 10.1016/j.ijforecast.2013.07.003 (cit. on p. 97).

Chen, C. and L.-M. Liu (1993). *Forecasting time series with outliers*. In: *Journal of Forecasting*, Vol. 12, No. 1, pp. 13–35. DOI: 10.1002/for.3980120103 (cit. on p. 4).

Chen, T. and C. Guestrin (2016). *XGBoost: A Scalable Tree Boosting System*. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 785–794. DOI: 10.1145/2939672.2939785 (cit. on pp. 51, 53, 103, 108–110).

Chen, W., K. Zhou, S. Yang, and C. Wu (2017). *Data quality of electricity consumption data in a smart grid environment*. In: *Renewable and Sustainable Energy Reviews*, Vol. 75, pp. 98–105. DOI: 10.1016/j.rser.2016.10.054 (cit. on p. 2).

Chen, X., C. Kang, X. Tong, Q. Xia, and J. Yang (2014). *Improving the accuracy of bus load forecasting by a two-stage bad data identification method*. In: *IEEE Transactions on Power Systems*, Vol. 29, No. 4, pp. 1634–1641. DOI: 10.1109/TPWRS.2014.2298463 (cit. on p. 98).

Chollet, F. et al. (2015). *Keras*. https://keras.io (cit. on pp. 29, 53, 109, 110).

Cover, T. M. and P. E. Hart (1967). *Nearest Neighbor Pattern Classification*. In: *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp. 21–27. DOI: 10.1109/TIT.1967.1053964 (cit. on pp. 28, 51).

Dannecker, L. (2015). *Energy Time Series Forecasting*. Wiesbaden: Springer Fachmedien. DOI: 10.1007/978-3-658-11039-0 (cit. on p. 11).

Dasgupta, D. and S. Forrest (1996). *Novelty Detection in Time Series Data using Ideas from Immunology*. In: *Proceedings of the Fifth International Conference on Intelligent Systems*, pp. 82–87 (cit. on p. 11).

De Nadai, M. and M. van Someren (2015). *Short-term anomaly detection in gas consumption through ARIMA and Artificial Neural Network forecast*. In: *2015 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS) Proceedings*. IEEE. DOI: 10.1109/EESMS.2015.7175886 (cit. on p. 15).

Denby, L. and R. D. Martin (1979). *Robust Estimation of the First-Order Autoregressive Parameter*. In: *Journal of the American Statistical Association*, Vol. 74, No. 365, pp. 140–146. DOI: 10.1080/01621459.1979.10481630 (cit. on p. 97).

Donevski, M. and T. Zia (2019). *A Survey of Anomaly and Automation from a Cybersecurity Perspective*. In: *2018 IEEE Globecom Workshops (GC Wkshps)*. IEEE. DOI: 10.1109/GLOCOMW.2018.8644456 (cit. on p. 2).

Drucker, H., C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik (1996). *Support Vector Regression Machines*. In: *Advances in Neural Information Processing Systems*. Ed. by M. C. Mozer, M. Jordan, and T. Petsche. Vol. 9. MIT Press (cit. on p. 107).

Dua, D. and C. Graff (2019). *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml (cit. on pp. 10, 12, 41, 77, 101).

Duong, T. V., H. H. Bui, D. Q. Phung, and S. Venkatesh (2005). *Activity recognition and abnormality detection with the switching hidden semi-Markov model*. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, pp. 838–845. DOI: 10.1109/CVPR.2005.61 (cit. on p. 11).

Ester, M., H.-P. Kriegel, J. Sander, and X. Xu (1996). *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 (KDD '96)*. Ed. by E. Simoudis, J. Han, and U. Fayyad. Palo Alto, CA, USA: AAAI Press, pp. 226–231. DOI: 10.5555/3001460.3001507 (cit. on p. 27).

Fahim, M., K. Fraz, and A. Sillitti (2020). *TSI: Time series to imaging based model for detecting anomalous energy consumption in smart buildings*. In: *Information Sciences*, Vol. 523, pp. 1–13. DOI: 10.1016/j.ins.2020.02.069 (cit. on p. 16).

Fan, C., F. Xiao, Y. Zhao, and J. Wang (2018). *Analytical investigation of autoencoder-based methods for unsupervised anomaly detection in building energy data*. In: *Applied Energy*, Vol. 211, pp. 1123–1135. DOI: 10.1016/j.apenergy.2017.12.005 (cit. on p. 15).

Fan, W., M. Miller, S. J. Stolfo, W. Lee, and P. K. Chan (2001). *Using artificial anomalies to detect unknown and known network intrusions*. In: *Knowledge and Information Systems*, Vol. 6, pp. 507–527. DOI: 10.1007/s10115-003-0132-7 (cit. on p. 16).

Fang, X., S. Misra, G. Xue, and D. Yang (2012). *Smart Grid - The New and Improved Power Grid: A Survey*. In: *IEEE Communications Surveys & Tutorials*, Vol. 14, No. 4, pp. 944–980. DOI: 10.1109/SURV.2011.101911.00087 (cit. on p. 1).

Foorthuis, R. (2021). *On the nature and types of anomalies: a review of deviations in data*. In: *International Journal of Data Science and Analytics*, Vol. 12, No. 4, pp. 297–331. DOI: 10.1007/s41060-021-00265-1 (cit. on pp. 13, 20).

Friese, M., J. Stork, R. Ramos Guerra, T. Bartz-Beielstein, S. Thaker, O. Flasch, and M. Zaefferer (2013). *UniFIeD Univariate Frequency-based Imputation for Time Series Data*. Tech. rep. Cologne University of Applied Sciences (cit. on pp. 69, 81).

Gaur, M., S. Makonin, I. V. Bajić, and A. Majumdar (2019). *Performance Evaluation of Techniques for Identifying Abnormal Energy Consumption in Buildings*. In: *IEEE Access*, Vol. 7, pp. 62721–62733. DOI: 10.1109/ACCESS.2019.2915641 (cit. on p. 16).

Giljum, S. and F. Hinterberger (2014). *The Limits of Resource Use and Their Economic and Policy Implications*. In: *Factor X: Policy, Strategies and Instruments for a Sustainable Resource Use*. Ed. by M. Angrick, A. Burger, and H. Lehmann. Dordrecht: Springer Netherlands, pp. 3–17. DOI: 10.1007/978-94-007-5706-6_1 (cit. on p. 1).

González Ordiano, J. Á., S. Waczowicz, V. Hagenmeyer, and R. Mikut (2018). *Energy forecasting tools and services*. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 8, No. 2, e1235. DOI: 10.1002/widm.1235 (cit. on pp. 4, 69).

Goodfellow, I., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2014). *Generative Adversarial Nets*. In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger. Vol. 27. Curran Associates, Inc., pp. 4089–4099 (cit. on p. 37).

Grané, A. and H. Veiga (2014). *Outliers, GARCH-type models and risk measures: A comparison of several approaches*. In: *Journal of Empirical Finance*, Vol. 26, pp. 26–40. DOI: 10.1016/j.jempfin.2014.01.005 (cit. on p. 97).

Gulati, M. and P. Arjunan (2022). *LEAD1.0: A Large-scale Annotated Dataset for Energy Anomaly Detection in Commercial Buildings*. In: *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22)*. ACM, pp. 485–488. DOI: 10.1145/3538637.3539761 (cit. on p. 15).

Gupta, M., J. Gao, C. C. Aggarwal, and J. Han (2014). *Outlier Detection for Temporal Data: A Survey*. In: *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 9, pp. 2250–2267. DOI: 10.1109/TKDE.2013.184 (cit. on p. 13).

Guralnik, V. and J. Srivastava (1999). *Event Detection from Time Series Data*. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '99)*. Ed. by S. Chaudhur and D. Madigan. ACM, pp. 33–42. DOI: 10.1145/312129.312190 (cit. on p. 11).

Hagenmeyer, V., H. K. Çakmak, C. Düpmeier, T. Faulwasser, J. Isele, H. B. Keller, P. Kohlhepp, U. Kühnapfel, U. Stucky, S. Waczowicz, and R. Mikut (2016). *Information and Communication Technology in Energy Lab 2.0: Smart Energies System Simulation and Control Center with an Open-Street-Map-Based Power Flow Simulation Example*. In: *Energy Technology*, Vol. 4, No. 1, pp. 145–162. DOI: 10.1002/ente.201500304 (cit. on p. 2).

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning*. 2nd Ed. New York: Springer. DOI: 10.1007/978-0-387-84858-7 (cit. on p. 51).

Hawkins, D. M. (1980). *Identification of Outliers*. Springer Science & Business Media. DOI: 10.1007/978-94-015-3994-4 (cit. on pp. 11, 13).

Heidrich, B., A. Bartschat, M. Turowski, O. Neumann, K. Phipps, S. Meisenbacher, K. Schmieder, N. Ludwig, R. Mikut, and V. Hagenmeyer (2021). *pyWATTS: Python Workflow Automation Tool for Time Series*. Under Review. arXiv: 2106.10157 (cit. on pp. 26, 29, 53, 110).

Heidrich, B., N. Ludwig, M. Turowski, and R. Mikut (2022). *Adaptively coping with concept drifts in energy time series forecasting using profiles*. In: *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22)*. ACM, pp. 459–470. DOI: 10.1145/3538637.3539759 (cit. on p. 11).

Heidrich, B., M. Turowski, N. Ludwig, R. Mikut, and V. Hagenmeyer (2020). *Forecasting energy time series with profile neural networks*. In: *The Eleventh ACM International Conference on Future Energy Systems (e-Energy '20)*, pp. 220–230. DOI: 10.1145/3396851.3397683 (cit. on pp. 106, 110).

Heidrich, B., M. Turowski, K. Phipps, K. Schmieder, W. Süß, R. Mikut, and V. Hagenmeyer (2023). *Controlling Non-Stationarity and Periodicities in Time Series Generation Using Conditional Invertible Neural Networks*. In: *Applied Intelligence*, Vol. 53, pp. 8826–8843. DOI: 10.1007/s10489-022-03742-7 (cit. on pp. 16, 48).

Himeur, Y., A. Alsalemi, F. Bensaali, and A. Amira (2020a). *A Novel Approach for Detecting Anomalous Energy Consumption Based on Micro-Moments and Deep Neural Networks*. In: *Cognitive Computation*, Vol. 12, No. 6, pp. 1381–1401. DOI: 10.1007/s12559-020-09764-y (cit. on pp. 15, 16).

Himeur, Y., A. Alsalemi, F. Bensaali, and A. Amira (2020b). *Building power consumption datasets: Survey, taxonomy and future directions*. In: *Energy and Buildings*, Vol. 227, p. 110404. DOI: 10.1016/j.enbuild.2020.110404 (cit. on p. 15).

Himeur, Y., K. Ghanem, A. Alsalemi, F. Bensaali, and A. Amira (2021). *Artificial intelligence based anomaly detection of energy consumption in buildings: A review, current trends and new perspectives*. In: *Applied Energy*, Vol. 287, p. 116601. DOI: 10.1016/j.apenergy.2021.116601 (cit. on pp. 2, 15, 35).

Hinton, G., N. Srivastava, and K. Swersky (2012). *Neural Networks for Machine Learning Lecture: Lecture 6a Overview of mini-batch gradient descent*. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf (cit. on p. 28).

Hung, C.-M., S. Zhong, W. Goodwin, O. P. Jones, M. Engelcke, I. Havoutis, and I. Posner (2022). *Reaching Through Latent Space: From Joint Statistics to Path Planning in Manipulation*. In: *IEEE Robotics and Automation Letters*, Vol. 7, No. 2, pp. 5334–5341. DOI: 10.1109/LRA.2022.3152697 (cit. on p. 35).

Hyndman, R. J. and G. Athanasopoulos (2021). *Forecasting: Principles and Practice*. 3rd Ed. Melbourne, Australia: OTexts (cit. on pp. 9, 26).

Ipakchi, A. and F. Albuyeh (2009). *Grid of the Future*. In: *IEEE Power and Energy Magazine*, Vol. 7, No. 2, pp. 52–62. DOI: 10.1109/MPE.2008.931384 (cit. on p. 1).

Jagadish, H. V., N. Koudas, and S. Muthukrishnan (1999). *Mining Deviants in a Time Series Database*. In: *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB '99)*. Morgan Kaufmann Publishers, pp. 102–113 (cit. on p. 11).

Jakkula, V. and D. Cook (2010). *Outlier Detection in Smart Environment Structured Power Datasets*. In: *2010 Sixth International Conference on Intelligent Environments*. IEEE, pp. 29–33. DOI: 10.1109/IE.2010.13 (cit. on pp. 15, 16).

Jiao, J., Z. Tang, P. Zhang, M. Yue, and J. Yan (2022). *Cyberattack-resilient load forecasting with adaptive robust regression*. In: *International Journal of Forecasting*, Vol. 38, No. 3, pp. 910–919. DOI: 10.1016/j.ijforecast.2021.06.009 (cit. on p. 97).

Jokar, P., N. Arianpoo, and V. C. M. Leung (2016). *Electricity Theft Detection in AMI Using Customers' Consumption Patterns*. In: *IEEE Transactions on Smart Grid*, Vol. 7, No. 1, pp. 216–226. DOI: 10.1109/TSG.2015.2425222 (cit. on pp. 2, 13, 15, 16).

Katipamula, S. and M. R. Brambley (2005a). *Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems - A Review, Part I*. In: *HVAC&R Research*, Vol. 11, No. 1, pp. 3–25. DOI: 10.1080/10789669.2005.10391123 (cit. on p. 11).

Katipamula, S. and M. R. Brambley (2005b). *Methods for Fault Detection, Diagnostics, and Prognostics for Building Systems - A Review, Part II*. In: *HVAC&R Research*, Vol. 11, No. 2, pp. 169–187. DOI: 10.1080/10789669.2005.10391133 (cit. on p. 11).

Keogh, E., S. Lonardi, and B. I. Y.-c. Chiu (2002). *Finding surprising patterns in a time series database in linear time and space*. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*. ACM, pp. 550–556. DOI: 10.1145/775107.775128 (cit. on p. 11).

Khalilnejad, A., A. M. Karimi, S. Kamath, R. Haddadian, R. H. French, and A. R. Abramson (2020). *Automated pipeline framework for processing of large-scale building energy time series data*. In: *PLoS ONE*, Vol. 15, No. 12, e0240461. DOI: 10.1371/journal.pone.0240461 (cit. on p. 2).

Kim, J.-Y. and S.-B. Cho (2021). *Explainable prediction of electric energy demand using a deep autoencoder with interpretable latent space*. In: *Expert Systems with Applications*, Vol. 186, p. 115842. DOI: 10.1016/j.eswa.2021.115842 (cit. on p. 35).

Kingma, D. P. and J. L. Ba (2015). *Adam: A Method for Stochastic Optimization*. In: *3rd International Conference on Learning Representations (ICLR 2015)* (cit. on pp. 48, 106).

Kingma, D. P. and P. Dhariwal (2018). *Glow: Generative Flow with Invertible 1x1 Convolutions*. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., pp. 10215–10224 (cit. on pp. 37, 48).

Kingma, D. P. and M. Welling (2014). *Auto-Encoding Variational Bayes*. arXiv: 1312.6114v10 (cit. on pp. 37, 52, 104).

Kitts, C. (2006). *Managing space system anomalies using first principles reasoning*. In: *IEEE Robotics and Automation Magazine*, Vol. 13, No. 4, pp. 39–50. DOI: 10.1109/MRA.2006.250571 (cit. on p. 2).

Krawczyk, B. (2016). *Learning from imbalanced data: open challenges and future directions*. In: *Progress in Artificial Intelligence*, Vol. 5, No. 4, pp. 221–232. DOI: 10.1007/s13748-016-0094-0 (cit. on p. 16).

Kullback, S. and R. A. Leibler (1951). *On Information and Sufficiency*. In: *Annals of Mathematical Statistics*, Vol. 22, No. 1, pp. 79–86. DOI: 10.1214/aoms/1177729694 (cit. on p. 39).

Kutsuzawa, K., S. Sakaino, and T. Tsuji (2019). *Trajectory adjustment for nonprehensile manipulation using latent space of trained sequence-to-sequence model*. In: *Advanced Robotics*, Vol. 33, No. 21, pp. 1144–1154. DOI: 10.1080/01691864.2019.1673204 (cit. on p. 35).

Laptev, N. (2018). *AnoGen: Deep Anomaly Generator*. In: *Outlier Detection De-constructed (ODD) Workshop (ODD v5.0)*. DOI: 10.475/123 (cit. on p. 16).

Lee, J., C. Jin, Z. Liu, and H. Davari Ardakani (2017). *Introduction to Data-Driven Methodologies for Prognostics and Health Management*. In: *Probabilistic Prognostics and Health Management of Energy Systems*. Ed. by S. Ekwaro-Osire, A. C. Gonçalves, and F. M. Alemayehu. Cham: Springer International Publishing, pp. 9–32. DOI: 10.1007/978-3-319-55852-3_2 (cit. on p. 2).

Li, F., W. Qiao, H. Sun, H. Wan, J. Wang, Y. Xia, Z. Xu, and P. Zhang (2010a). *Smart Transmission Grid: Vision and Framework*. In: *IEEE Transactions on Smart Grid*, Vol. 1, No. 2, pp. 168–177. DOI: 10.1109/TSG.2010.2053726 (cit. on p. 1).

Li, X., C. P. Bowers, and T. Schnier (2010b). *Classification of Energy Consumption in Buildings With Outlier Detection*. In: *IEEE Transactions on Industrial Electronics*, Vol. 57, No. 11, pp. 3639–3644. DOI: 10.1109/TIE.2009.2027926 (cit. on p. 15).

Liu, F. T., K. M. Ting, and Z.-H. Zhou (2008). *Isolation Forest*. In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE, pp. 413–422. DOI: 10.1109/ICDM.2008.17 (cit. on pp. 51, 104).

Lonardi, S., J. Lin, E. Keogh, and B. l. Y.-c. Chiu (2006). *Efficient Discovery of Unusual Patterns in Time Series*. In: *New Generation Computing*, Vol. 25, No. 1, pp. 61–93. DOI: 10.1007/s00354-006-0004-2 (cit. on p. 11).

Lu, H., M. Du, K. Qian, X. He, and K. Wang (2021). *GAN-based Data Augmentation Strategy for Sensor Anomaly Detection in Industrial Robots*. In: *IEEE Sensors Journal*. DOI: 10.1109/JSEN.2021.3069452 (cit. on p. 16).

Luo, J., T. Hong, and S.-C. Fang (2018a). *Benchmarking robustness of load forecasting models under data integrity attacks*. In: *International Journal of Forecasting*, Vol. 34, No. 1, pp. 89–104. DOI: 10.1016/j.ijforecast.2017.08.004 (cit. on p. 97).

Luo, J., T. Hong, and S.-C. Fang (2018b). *Robust Regression Models for Load Forecasting*. In: *IEEE Transactions on Smart Grid*, Vol. 10, No. 5, pp. 5397–5404. DOI: 10.1109/TSG.2018.2881562 (cit. on p. 97).

Luo, J., T. Hong, Z. Gao, and S.-C. Fang (2023). *A robust support vector regression model for electric load forecasting*. In: *International Journal of Forecasting*, Vol. 39, No. 2, pp. 1005–1020. DOI: 10.1016/j.ijforecast.2022.04.001 (cit. on p. 97).

Luo, J., T. Hong, and M. Yue (2018c). *Real-time anomaly detection for very short-term load forecasting*. In: *Journal of Modern Power Systems and Clean Energy*, Vol. 6, No. 2, pp. 235–243. DOI: 10.1007/s40565-017-0351-7 (cit. on pp. 2, 4, 16, 98).

Markus, A. A., B. W. Hobson, H. B. Gunay, and S. Bucking (2021). *A framework for a multi-source, data-driven building energy management toolkit*. In: *Energy and Buildings*, Vol. 250, p. 111255. DOI: 10.1016/j.enbuild.2021.111255 (cit. on p. 2).

Mateos, G. and G. B. Giannakis (2013). *Load Curve Data Cleansing and Imputation Via Sparsity and Low Rank*. In: *IEEE Transactions on Smart Grid*, Vol. 4, No. 4, pp. 2347–2355. DOI: 10.1109/TSG.2013.2259853 (cit. on pp. 70, 81).

Matheson, D., C. Jing, and F. Monforte (2004). *Meter Data Management for the Electricity Market*. In: *2004 International Conference on Probabilistic Methods Applied to Power Systems*. IEEE, pp. 118–122 (cit. on pp. 70, 81).

Meisenbacher, S., M. Turowski, K. Phipps, M. Rätz, D. Müller, V. Hagenmeyer, and R. Mikut (2022). *Review of automated time series forecasting pipelines*. In: *WIREs Data Mining and Knowledge Discovery*, Vol. 12, No. 6, e1475. DOI: 10.1002/widm.1475 (cit. on pp. 2, 3).

Menon, S., A. Damian, S. Hu, N. Ravi, and C. Rudin (2020). *PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models*. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 2437–2445. DOI: 10.1109/CVPR42600.2020.00251 (cit. on p. 35).

Mitchell, T. M. (1997). *Machine Learning*. New York: McGraw-Hill (cit. on pp. 51, 106).

Moghaddass, R. and J. Wang (2018). *A Hierarchical Framework for Smart Grid Anomaly Detection Using Large-Scale Smart Meter Data*. In: *IEEE Transactions on Smart Grid*, Vol. 9, No. 6, pp. 5820–5830. DOI: 10.1109/TSG.2017.2697440 (cit. on p. 13).

Mohassel, R. R., A. Fung, F. Mohammadi, and K. Raahemifar (2014). *A survey on Advanced Metering Infrastructure*. In: *International Journal of Electrical Power and Energy Systems*, Vol. 63, pp. 473–484. DOI: 10.1016/j.ijepes.2014.06.025 (cit. on p. 9).

Moritz, S. and T. Bartz-Beielstein (2017). *imputeTS: Time Series Missing Value Imputation in R*. In: *R Journal*, Vol. 9, No. 1, pp. 207–218. DOI: 10.32614/rj-2017-009 (cit. on pp. 69, 81).

Nguyen, N. and B. Quanz (2021). *Temporal Latent Auto-Encoder: A Method for Probabilistic Multivariate Time Series Forecasting*. In: *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*. AAAI Press, pp. 9117–9125 (cit. on p. 35).

Ni, K., N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava (2009). *Sensor network data fault types*. In: *ACM Transactions on Sensor Networks*, Vol. 5, No. 3, 25:1–25:29. DOI: 10.1145/1525856.1525863 (cit. on p. 20).

Nizar, A. H., Z. Y. Dong, and Y. Wang (2008). *Power Utility Nontechnical Loss Analysis With Extreme Learning Machine Method*. In: *IEEE Transactions on Power Systems*, Vol. 23, No. 3, pp. 946–955. DOI: 10.1109/TPWRS.2008.926431 (cit. on p. 13).

Nordahl, C., M. Persson, and H. Grahn (2017). *Detection of residents' abnormal behaviour by analysing energy consumption of individual households*. In: *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 729–738. DOI: 10.1109/ICDMW.2017.101 (cit. on p. 2).

Nyitrai, T. and M. Virág (2019). *The effects of handling outliers on the performance of bankruptcy prediction models*. In: *Socio-Economic Planning Sciences*, Vol. 67, pp. 34–42. DOI: 10.1016/j.seps.2018.08.004 (cit. on p. 97).

Pang, G., C. Shen, L. Cao, and A. van den Hengel (2021). *Deep Learning for Anomaly Detection: A Review*. In: *ACM Computing Surveys*, Vol. 54, No. 2. DOI: 10.1145/3439950 (cit. on p. 15).

Paris Agreement (2015). https://unfccc.int/sites/default/files/english_paris_agreement.pdf (cit. on p. 1).

Park, Y., I. M. Molloy, S. N. Chari, Z. Xu, C. Gates, and N. Li (2015). *Learning from Others: User Anomaly Detection Using Anomalous Samples from Other Users*. In: *Computer Security – ESORICS 2015*. Ed. by G. Pernul, P. Y. A. Ryan, and E. Weippl. Cham: Springer, pp. 396–414. DOI: 10.1007/978-3-319-24177-7_20 (cit. on p. 16).

Paszke, A., S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library*. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. D'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. (cit. on pp. 53, 109).

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay (2011). *Scikit-learn: Machine Learning in*

Python. In: *Journal of Machine Learning Research*, Vol. 12, No. 85, pp. 2825–2830 (cit. on pp. 29, 53).

Peppanen, J., X. Zhang, S. Grijalva, and M. J. Reno (2016). *Handling Bad or Missing Smart Meter Data through Advanced Data Imputation*. In: *2016 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE. DOI: 10.1109/ISGT.2016.7781213 (cit. on pp. 69, 70, 81).

Pereira, J. and M. Silveira (2018). *Unsupervised Anomaly Detection in Energy Time Series Data Using Variational Recurrent Autoencoders with Attention*. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, pp. 1275–1282. DOI: 10.1109/ICMLA.2018.00207 (cit. on p. 15).

Pereira, J. and M. Silveira (2019). *Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection*. In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp 2019)*. IEEE. DOI: 10.1109/BIGCOMP.2019.8679157 (cit. on p. 35).

Petropoulos, F. et al. (2022). *Forecasting: theory and practice*. In: *International Journal of Forecasting*, Vol. 38, No. 3, pp. 705–871. DOI: 10.1016/j.ijforecast.2021.11.001 (cit. on p. 4).

Pham, T. S., Q. Uy Nguyen, and X. H. Nguyen (2014). *Generating Artificial Attack Data for Intrusion Detection Using Machine Learning*. In: *Proceedings of the Fifth Symposium on Information and Communication Technology (SoICT '14)*. ACM, pp. 286–291. DOI: 10.1145/2676585.2676618 (cit. on p. 16).

Quintana, M., T. Stoeckmann, J. Y. Park, M. Turowski, V. Hagenmeyer, and C. Miller (2022). *ALDI++: Automatic and parameter-less discord and outlier detection for building energy load profiles*. In: *Energy & Buildings*, Vol. 265, p. 112096. DOI: 10.1016/j.enbuild.2022.112096 (cit. on p. 97).

Rafailov, R., T. Yu, A. Rajeswaran, and C. Finn (2021). *Offline Reinforcement Learning from Images with Latent Space Models*. In: *Proceedings of the 3rd Conference on Learning for Dynamics and Control*. Ed. by A. Jadbabaie, J. Lygeros, G. J. Pappas, P. A. Parrilo, B. Recht, C. J. Tomlin, and M. N. Zeilinger. Vol. 144. PMLR, pp. 1154–1168 (cit. on p. 35).

Ranjan, K. G., B. R. Prusty, and D. Jena (2021). *Review of preprocessing methods for univariate volatile time-series in power system applications*. In: *Electric Power Systems Research*, Vol. 191, p. 106885. DOI: 10.1016/j.epsr.2020.106885 (cit. on p. 97).

Rodrigues, F. and A. Trindade (2018). *Load forecasting through functional clustering and ensemble learning*. In: *Knowledge and Information Systems*, Vol. 57, No. 1, pp. 229–244. DOI: 10.1007/s10115-018-1169-y (cit. on pp. 41, 77, 101).

Rossi, B. and S. Chren (2020). *Smart Grids Data Analysis: A Systematic Mapping Study*. In: *IEEE Transactions on Industrial Informatics*, Vol. 16, No. 6, pp. 3619–3639. DOI: 10.1109/TII.2019.2954098 (cit. on pp. 1, 16).

Rousseeuw, P. J. and A. M. Leroy (1987). *Robust Regression and Outlier Detection*. John Wiley & Sons. DOI: 10.1002/0471725382 (cit. on pp. 11, 13).

Ruff, L., J. R. Kauffmann, R. A. Vandermeulen, G. Montavon, W. Samek, M. Kloft, T. G. Dietterich, and K.-R. Müller (2021). *A unifying review of deep and shallow anomaly detection*. In: *Proceedings of the IEEE*, Vol. 109, No. 5, pp. 756–795. DOI: 10.1109/JPROC.2021.3052449 (cit. on pp. 15, 16).

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). *Learning Representations by Back-Propagating Errors*. In: *Nature*, Vol. 323, pp. 533–536. DOI: 10.1038/323533a0 (cit. on p. 51).

Samitier, C. (2017). *Managing Faults and Anomalies*. In: *Utility Communication Networks and Services*. Cham: Springer, pp. 213–218. DOI: 10.1007/978-3-319-40283-3_31 (cit. on p. 2).

Schmidl, S., P. Wenig, and T. Papenbrock (2022). *Anomaly Detection in Time Series: A Comprehensive Evaluation*. In: *Proceedings of the VLDB Endowment*, Vol. 15, No. 9, pp. 1779–1797. DOI: 10.14778/3538598.3538602 (cit. on p. 35).

Seem, J. E. (2007). *Using intelligent data analysis to detect abnormal energy consumption in buildings*. In: *Energy and Buildings*, Vol. 39, No. 1, pp. 52–58. DOI: 10.1016/j.enbuild.2006.03.033 (cit. on pp. 2, 13).

Shafiee, S. and E. Topal (2009). *When will fossil fuel reserves be diminished?* In: *Energy Policy*, Vol. 37, No. 1, pp. 181–189. DOI: 10.1016/j.enpol.2008.08.016 (cit. on p. 1).

Shahabi, C., X. Tian, and W. Zhao (2000). *TSA-tree: A Wavelet-Based Approach to Improve the Efficiency of Multi-level Surprise and Trend Queries on Time-Series Data*. In: *Proceedings. 12th International Conference on Scientific and Statistica Database Management*. IEEE, pp. 55–68. DOI: 10.1109/SSDM.2000.869778 (cit. on p. 11).

Sharma, A. B., L. Golubchik, and R. Govindan (2010). *Sensor Faults: Detection Methods and Prevalence in Real-World Datasets*. In: *ACM Transactions on Sensor Networks*, Vol. 6, No. 3, 23:1–23:39. DOI: 10.1145/1754414.1754419 (cit. on p. 20).

Skea, J. et al. (2022). *Climate Change 2022: Mitigation of Climate Change. Summary for Policymakers*. Intergovernmental Panel on Climate Change (cit. on p. 1).

Sohn, K., X. Yan, and H. Lee (2015). *Learning Structured Output Representation using Deep Conditional Generative Models*. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., pp. 3483–3491 (cit. on pp. 37, 102).

Steinbuss, G. and K. Böhm (2021a). *Benchmarking Unsupervised Outlier Detection with Realistic Synthetic Data*. In: *ACM Transactions on Knowledge Discovery from Data*, Vol. 15, No. 4. DOI: 10.1145/3441453 (cit. on p. 16).

Steinbuss, G. and K. Böhm (2021b). *Generating Artificial Outliers in the Absence of Genuine Ones – A Survey*. In: *ACM Transactions on Knowledge Discovery from Data*, Vol. 15, No. 2. DOI: 10.1145/3447822 (cit. on p. 16).

Tan, P.-N., M. Steinbach, A. Karpatne, and V. Kumar (2019). *Introduction to Data Mining*. 2nd Ed. New York: Pearson (cit. on pp. 51, 103).

Taylor, S. J. and B. Letham (2018). *Forecasting at Scale*. In: *American Statistician*, Vol. 72, No. 1, pp. 37–45. DOI: 10.1080/00031305.2017.1380080 (cit. on pp. 69, 74, 81, 82, 104, 109).

Teng, S. Y., V. Máša, M. Touš, M. Vondra, H. L. Lam, and P. Stehlík (2022). *Waste-to-energy forecasting and real-time optimization: An anomaly-aware approach*. In: *Renewable Energy*, Vol. 181, pp. 142–155. DOI: 10.1016/j.renene.2021.09.026 (cit. on p. 98).

Turowski, M., B. Heidrich, K. Phipps, K. Schmieder, O. Neumann, R. Mikut, and V. Hagenmeyer (2022a). *Enhancing Anomaly Detection Methods for Energy Time Series Using Latent Space Data Representations*. In: *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22)*. ACM, pp. 208–227. DOI: 10.1145/3538637.3538851 (cit. on pp. 35, 101–103, 109).

Turowski, M., O. Neumann, L. Mannsperger, K. Kraus, K. Layer, R. Mikut, and V. Hagenmeyer (2023). *Managing Anomalies in Energy Time Series for Automated Forecasting*. Submitted (cit. on p. 97).

Turowski, M., M. Weber, O. Neumann, B. Heidrich, K. Phipps, H. K. Çakmak, R. Mikut, and V. Hagenmeyer (2022b). *Modeling and Generating Synthetic Anomalies for Energy and Power Time Series*. In: *The Thirteenth ACM International Conference on Future Energy Systems (e-Energy '22)*. ACM, pp. 471–484. DOI: 10.1145/3538637.3539760 (cit. on pp. 15, 41, 45, 79, 101, 102).

Van der Maaten, L. and G. Hinton (2008). *Visualizing Data using t-SNE*. In: *Journal of Machine Learning Research*, Vol. 9, No. 86, pp. 2579–2605 (cit. on pp. 28, 54).

Vapnik, V. N. (2000). *The Nature of Statistical Learning Theory*. 2nd Ed. New York: Springer. DOI: 10.1007/978-1-4757-3264-1 (cit. on pp. 51, 103).

Villar-Rodriguez, E., J. Del Ser, I. Oregi, M. N. Bilbao, and S. Gil-Lopez (2017). *Detection of non-technical losses in smart meter data based on load curve profiling and time series analysis*. In: *Energy*, Vol. 137, pp. 118–128. DOI: 10.1016/j.energy.2017.07.008 (cit. on p. 16).

Wang, C., K. Wu, T. Zhou, G. Yu, and Z. Cai (2021). *TSAGen: Synthetic Time Series Generation for KPI Anomaly Detection*. In: *IEEE Transactions on Network and Service Management*. DOI: 10.1109/TNSM.2021.3098784 (cit. on p. 16).

Wang, L., Y. Ding, T. Riedel, A. Miclaus, and M. Beigl (2017). *Data Analysis on Building Load Profiles: a Stepping Stone to Future Campus*. In: *2017 International Smart Cities Conference (ISC2)*. IEEE. DOI: 10.1109/ISC2.2017.8090823 (cit. on p. 17).

Wang, L., M. Turowski, M. Zhang, T. Riedel, M. Beigl, R. Mikut, and V. Hagenmeyer (2020). *Point and contextual anomaly detection in building load profiles of a university campus*. In: *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, pp. 11–15. DOI: 10.1109/ISGT-Europe47291.2020.9248792 (cit. on pp. 2, 13, 41).

Wang, Y., Q. Chen, T. Hong, and C. Kang (2019). *Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges*. In: *IEEE Transactions on Smart Grid*, Vol. 10, No. 3, pp. 3125–3148. DOI: 10.1109/TSG.2018.2818167 (cit. on pp. 1, 2, 11, 15, 97).

Weber, M., M. Turowski, H. K. Çakmak, R. Mikut, U. Kühnapfel, and V. Hagenmeyer (2021). *Data-Driven Copy-Paste Imputation for Energy Time Series*. In: *IEEE Transactions on Smart Grid*, Vol. 12, No. 6, pp. 5409–5419. DOI: 10.1109/TSG.2021.3101831 (cit. on pp. 26, 46, 69, 104, 109).

Wen, Q., L. Sun, F. Yang, X. Song, J. Gao, X. Wang, and H. Xu (2021). *Time Series Data Augmentation for Deep Learning: A Survey*. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, pp. 4653–4660. DOI: 10.24963/ijcai.2021/631 (cit. on pp. 15, 16).

Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph. D. thesis. Harvard University (cit. on pp. 51, 106).

Xie, J. and T. Hong (2016). *GEFCom2014 probabilistic electric load forecasting: An integrated solution with forecast combination and residual simulation*. In: *International Journal of Forecasting*, Vol. 32, No. 3, pp. 1012–1016. DOI: 10.1016/j.ijforecast.2015.11.005 (cit. on p. 97).

Xie, X., W. Sun, and K. C. Cheung (2016). *An Advanced PLS approach for key performance indicator-related prediction and diagnosis in case of outliers*. In: *IEEE Transactions on Industrial Electronics*, Vol. 63, No. 4, pp. 2587–2594. DOI: 10.1109/TIE.2015.2512221 (cit. on p. 97).

Yamanishi, K. and J.-i. Takeuchi (2002). *A Unifying Framework for Detecting Outliers and Change Points from Non-Stationary Time Series Data*. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*. ACM, pp. 676–681. DOI: 10.1145/775047.775148 (cit. on p. 11).

Yankov, D., E. Keogh, and U. Rebbapragada (2008). *Disk aware discord discovery: Finding unusual time series in terabyte sized datasets*. In: *Knowledge and Information Systems*, Vol. 17, No. 2, pp. 241–262. DOI: 10.1007/s10115-008-0131-9 (cit. on p. 11).

Yin, S. and G. Wang (2013). *A modified partial robust M-regression to improve prediction performance for data with outliers*. In: *IEEE International Symposium on Industrial Electronics*. IEEE. DOI: 10.1109/ISIE.2013.6563843 (cit. on p. 97).

Yin, S., G. Wang, and X. Yang (2014). *Robust PLS approach for KPI-related prediction and diagnosis against outliers and missing data*. In: *International Journal of Systems Science*, Vol. 45, No. 7, pp. 1375–1382. DOI: 10.1080/00207721.2014.886136 (cit. on p. 97).

Yue, M., T. Hong, and J. Wang (2019). *Descriptive Analytics-Based Anomaly Detection for Cybersecure Load Forecasting*. In: *IEEE Transactions on Smart Grid*, Vol. 10, No. 6, pp. 5964–5974. DOI: 10.1109/TSG.2019.2894334 (cit. on p. 97).

Zhang, J. Q. and Y. Yan (2001). *A Wavelet-Based Approach to Abrupt Fault Detection and Diagnosis of Sensors*. In: *IEEE Transactions on Instrumentation and Measurement*, Vol. 50, No. 5, pp. 1389–1396. DOI: 10.1109/19.963215 (cit. on p. 20).

Zhang, J. E., D. Wu, and B. Boulet (2021). *Time Series Anomaly Detection for Smart Grids: A Survey*. In: *2021 IEEE Electrical Power and Energy Conference (EPEC)*. IEEE, pp. 125–130. DOI: 10.1109/epec52095.2021.9621752 (cit. on pp. 15, 16).

Zhang, Y., F. Lin, and K. Wang (2020). *Robustness of Short-Term Wind Power Forecasting against False Data Injection Attacks*. In: *Energies*, Vol. 13, No. 15. DOI: 10.3390/en13153780 (cit. on p. 97).

Zhao, X., J. Kim, K. Warns, X. Wang, P. Ramuhalli, S. Cetiner, H. G. Kang, and M. Golay (2021). *Prognostics and Health Management in Nuclear Power Plants: An Updated Method-Centric Review With Special Focus on Data-Driven Methods*. In: *Frontiers in Energy Research*, Vol. 9. DOI: 10.3389/fenrg.2021.696785 (cit. on p. 2).

Zhao, Y., S. Wang, and F. Xiao (2013). *Pattern recognition-based chillers fault detection method using Support Vector Data Description (SVDD)*. In: *Applied Energy*, Vol. 112, pp. 1041–1048. DOI: 10.1016/j.apenergy.2012.12.043 (cit. on p. 16).

Zheng, R., J. Gu, Z. Jin, H. Peng, and Y. Zhu (2020). *Load forecasting under data corruption based on anomaly detection and combined robust regression*. In: *International Transactions on Electrical Energy Systems*, Vol. 30, No. 7, e12103. DOI: 10.1002/2050-7038.12103 (cit. on p. 97).

Zhou, B., S. Liu, B. Hooi, X. Cheng, and J. Ye (2019). *BeatGAN: Anomalous Rhythm Detection using Adversarially Generated Time Series*. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, pp. 4433–4439. DOI: 10.24963/ijcai.2019/616 (cit. on p. 16).

Zhou, Y., Z. Ding, Q. Wen, and Y. Wang (2022). *Robust Load Forecasting towards Adversarial Attacks via Bayesian Learning*. In: *IEEE Transactions on Power Systems*, Vol. 8950. DOI: 10.1109/TPWRS.2022.3175252 (cit. on p. 97).

Zhu, J., Y. Shen, Z. Song, D. Zhou, Z. Zhang, and A. Kusiak (2019). *Data-driven building load profiling and energy management*. In: *Sustainable Cities and Society*, Vol. 49, p. 101587. DOI: 10.1016/j.scs.2019.101587 (cit. on p. 2).