Modeling and Communicating Flexibility in Smart Grids Using Artificial Neural Networks as Surrogate Models

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften

Dr.-Ing.

von der KIT-Fakultät für Wirtschaftswissenschaften
des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Kevin Michael Förderer, M.Sc.

Tag der mündlichen Prüfung: 13.07.2021

Referent: Prof. Dr. Hartmut Schmeck
Korreferent: Prof. Dr. Andreas Oberweis

Abstract

Increasing shares of renewable energies and the transition towards electric vehicles pose major challenges to the energy system. In order to tackle these in an economically sensible way, the flexibility of distributed energy resources (DERs), such as battery energy storage systems, combined heat and power plants, and heat pumps, needs to be exploited. Modeling and communicating this flexibility is a fundamental step when trying to achieve control over DERs. The literature proposes and makes use of many different approaches, not only for the exploitation itself, but also in terms of models.

In the first step, this thesis presents an extensive literature review and a general framework for classifying exploitation approaches and the communicated models. Often, the employed models only apply to specific types of DERs, or the models are so abstract that they neglect constraints and only roughly outline the true flexibility. Surrogate models, which are learned from data, can pose as generic DER models and may potentially be trained in a fully automated process.

In this thesis, the idea of encoding the flexibility of DERs into ANNs is systematically investigated. Based on the presented framework, a set of ANN-based surrogate modeling approaches is derived and outlined, of which some are only applicable for specific use cases. In order to establish a baseline for the approximation quality, one of the most versatile identified approaches is evaluated in order to assess how well a set of reference models is approximated. If this versatile model is able to capture the flexibility well, a more specific model can be expected to do so even better.

The results show that simple DERs are very closely approximated, and for more complex DERs and combinations of multiple DERs, a high approximation quality can be achieved by introducing buffers. Additionally, the investigated approach has been tested in scheduling tasks for multiple different DERs, showing that it is indeed possible to use ANN-based surrogates for the flexibility of DERs to derive load schedules. Finally, the computational complexity of utilizing the different approaches for controlling DERs is compared.

Acknowledgments

First and foremost, my sincere thanks and gratitude to my supervisor Prof. Hartmut Schmeck for giving me the opportunity to pursue a doctoral degree and providing guidance, valuable advice, as well as the freedom to direct my research activities according to my personal interests. I would also like to express my appreciation to Prof. Andreas Oberweis, Prof. Wolf Fichtner, and Prof. Andreas Geyer-Schulz for their interest in my work.

Thanks so much to my colleagues at the FZI Research Center for Information Technology and at KIT for the welcoming and fantastic working atmosphere, all the fascinating discussions, and the valuable feedback. I am very happy to have made new friends and rekindled old friendships during my time at the FZI.

My deepest gratitude goes to my friends and family. I cannot thank you enough for all the awesome experiences and I will fondly recall our culinary, travel, sports, and many other activities. Finally, a very special thanks to my parents, who encouraged and supported me in pursuing a higher education.

Table of contents

Abstract . . . iii

Acknowledgments . . . v

Abbreviations and Notation . . . xi

1 Introduction . . . 1
- 1.1 Motivation . . . 1
- 1.2 Research Questions . . . 3
- 1.3 Contributions . . . 4
- 1.4 Previous Publications . . . 5
- 1.5 Structure . . . 8

2 Fundamentals . . . 9
- 2.1 Smart Grids . . . 9
  - 2.1.1 Ancillary Services . . . 9
  - 2.1.2 Demand Side Management . . . 10
  - 2.1.3 A Definition of Flexibility . . . 12
- 2.2 Communication Relationships . . . 13
  - 2.2.1 Control Structures . . . 13
  - 2.2.2 Stigmergy . . . 14
- 2.3 Modeling . . . 15
  - 2.3.1 Abstractness . . . 15
  - 2.3.2 White-Box, Black-Box, and Grey-Box Models . . . 15
  - 2.3.3 Surrogate Modeling . . . 16
- 2.4 Machine Learning . . . 16
  - 2.4.1 Experience . . . 17
  - 2.4.2 Tasks . . . 17
  - 2.4.3 Performance measures . . . 18
  - 2.4.4 Artificial Neural Networks . . . 19
  - 2.4.5 Support Vector Data Description . . . 23
- 2.5 Optimization . . . 26
  - 2.5.1 Computational Complexity . . . 26
  - 2.5.2 Lagrange Multipliers . . . 27
  - 2.5.3 Gradient-Based Optimization . . . 28

2.5.4 Meta-Heuristics . . . 28
2.5.5 Black-Box Optimization . . . 29
2.5.6 Online Optimization . . . 30
2.5.7 Model Predictive Control . . . 30
2.5.8 Sequential Decision-Making . . . 30

3 State of the Technology . . . 35
3.1 Exploitation of Flexibility . . . 35
3.2 Modeling Flexibility . . . 35
3.3 Surrogate Modeling . . . 36
3.4 Learning About Flexibility . . . 37
3.5 Flexibility of Chemical Processes . . . 37

4 Communicating the Flexibility of Distributed Energy Resources . . . 39
4.1 Motivations for Communicating Flexibility . . . 40
4.2 General Patterns for the Exploitation of Flexibility . . . 42
4.2.1 Communicating Models for the Flexibility of Distributed Energy Resources . . . 46
4.3 Literature Review . . . 47
4.3.1 Direct Exploitation . . . 50
4.3.2 Exploitation of Abstracted Flexibility . . . 55
4.3.3 Market-based Exploitation . . . 58
4.3.4 Indirect Exploitation . . . 61
4.3.5 State Information Based Exploitation . . . 64
4.4 Modeling Flexibility with Regard to Communication . . . 67
4.4.1 General Modeling Considerations . . . 67
4.4.2 Condensed Results of the Literature Review . . . 67
4.4.3 Modeling Flexibility from a Data-Driven Perspective . . . 72
4.5 Exploitation of Flexibility in the Research Project C/sells . . . 74
4.6 Summary . . . 79

5 Artificial Neural Networks as Surrogate Models for Flexibility . . . 81
5.1 Benefits and Challenges of Surrogate Modeling . . . 82
5.2 Encoding the Flexibility of Distributed Energy Resources in Surrogate Models . . . 84
5.2.1 Direct Exploitation . . . 87
5.2.2 Exploitation of Abstracted Flexibility and Market-Based Exploitation . . . 94
5.2.3 Indirect Exploitation . . . 97
5.2.4 State Information Based Exploitation . . . 98
5.2.5 Summary . . . 98
5.3 Evaluation . . . 99
5.3.1 Goals and Evaluation Criteria . . . 101
5.3.2 Considered Distributed Energy Resources . . . 102
5.3.3 Data Sources . . . 106
5.3.4 Surrogate Generation . . . 124

5.3.5 Evaluation Process 127
5.3.6 Results 128
5.3.7 Discussion 150
5.4 Summary 158

6 Controlling Distributed Energy Resources Using Artificial-Neural-Network-Based Surrogate Models 161
6.1 Criteria for the Assessment 161
6.2 Optimizing with Artificial Neural Network Based Surrogates 163
6.3 Qualitative Assessment 171
6.4 Exemplary Implementation 174
6.4.1 Monte-Carlo-Tree-Search-Based Target Schedule Approximation 174
6.4.2 Results 178
6.5 Summary 180

7 Conclusion and Outlook 181
7.1 Research Questions Briefly Answered 183
7.2 Outlook 184

A Appendix 187
A.1 Ideas for Learning Abstract Representations 187

Bibliography 191

Previous Publications 215

Abbreviations and Notation

Abbreviations

Agg Aggregated

ANN Artificial neural network

BESS Battery energy storage system

BEV Battery electric vehicle

BMWi Bundesministerium für Wirtschaft und Energie (Federal Ministry for Economic Affairs and Energy)

CIM Common information model

CHPP Combined heat and power plant

CPP Critical peak pricing

CPU Central processing unit

CUDA Compute unified device architecture

DER Distributed energy resource

DR Demand response

DSM Demand side management

DSMgr Demand side manager

DSO Distribution system operator

EA Evolutionary algorithm

EMS Energy management system

ESHL Energy smart home lab

EV Electric vehicle

EVSE Electric vehicle supply equipment

FN False negative

FNR False negative rate

FP	False positive
FPR	False positive rate
FZI	Forschungszentrum Informatik
GCB	Gas condensing boiler
GHG	Green house gases
GPU	Graphics processing unit
HoLL	House of Living Labs
HP	Heat pump
HWT	Hot water tank
IEC	International Electrotechnical Commission
KIT	Karlsruher Institut für Technologie
LSE	Load serving entity
MAE	Mean absolute error
MCTS	Monte Carlo tree search
MDP	Markov decision process
MILP	Mixed integer linear program
MSE	Mean squared error
MPC	Model predictive control
PSO	Particle swarm optimization
PV	Photovoltaik
RAM	Random access memory
RES	Renewable energy sources
RMSE	Root mean squared error
RL	Reinforcement learning
SINTEG	Schaufenster für intelligente Energie: Forschung für das Stromnetz der Zukunft (Smart Energy Showcases — Digital Agenda for the Energy Transition
SOC	State of charge
SVDD	Support vector data description
SVM	Support vector machine

TCL	Thermostatically controlled load
TN	True negative
ToU	Time of use
TP	True positive
TSO	Transmission system operator
QoS	Quality of service

Notation

The following list summarizes the notation and symbols found throughout the thesis. Only variables and functions occuring multiple times in the document are listed here. The function $f(x)$ is reused multiple times in the document for the purpose of providing examples.

$\underline{X}, \overline{X}$	Lower and upper boundary of X
$\hat{x}$	Forecast of x
$\mathbb{1}_A(x)$	Indicator function
$\mathbb{R}_{>0}$	Positive real numbers
$x_{a:b}$	Vector $x = (x_a, x_{a+1}, \ldots, x_b)$ of length $b - a + 1$
$f(x)$	Some arbitrary function taking the arbitrary input x
$\mathbf{M}$	A very large number (“Big-M”)
t	Time step
Δt	Length of a time step in seconds
T	Time horizon as number of time steps
s_t	State at the beginning of time step t
a_t	Action taken in time step t
$A(s_t)$	Set of possible actions in state s_t
$r(s_t, a_t)$	Reward function, giving the reward of taking action a_t in state s_t
$p(s_{t+1} \| s_t, a_t)$	Conditional probability distribution of subsequent state

$s_{t+1} := f(s_t, a_t)$	Deterministic state transition function
$\pi(a\|s)$	Policy function
P_t^X	Power provided or consumed by X at time step t
$P_t^{X,\mathsf{H}}$	Thermal power provided or consumed by X at time step t
Q_t^X	Amount of electric energy stored in X at time step t
$Q_t^{X,\mathsf{H}}$	Amount of thermal energy stored in X
η_C^X	Charging efficieny of X
η_D^X	Discharging efficiency of X
Θ_t^{HWT}	Temperature of hot water tank
$\Theta_t^{\mathsf{ambient}}$	Ambient temperature
$(m_{i,j}^{\mathsf{CHPP,E}})$	Matrix determining electric power
$(m_{i,j}^{\mathsf{CHPP,H}}), (m_{i,j}^{\mathsf{GCB,H}})$	Matrix determining thermal power
τ_t^{CHPP}	Dwell time in current mode
$\underline{\tau^{\mathsf{CHPP,on}}}$	Minimum running time
$\underline{\tau^{\mathsf{CHPP,off}}}$	Minimum off time

1 Introduction

In this chapter, the motivation behind this work, investigated research questions, and scientific contributions are presented. Additionally, an overview of previous publications and an outline of the document structure are provided.

1.1 Motivation

Until the year 2050, the European Union aims to reach zero net emission of greenhouse gases (GHG). This goal has been set by the *European Green Deal* [1] in 2019, tightening the previous target, which mandated a reduction of at least 80%. Decarbonization plays a critical role in reaching this goal [1] and requires the replacement of fossil fuels with other sources of useful energy, like wind and solar power. Consider Germany as an example. The development of the share of renewable energies in Germany is depicted in Figure 1.1. In 2004, only 6.2% of the gross final energy consumption was satisfied with energy from renewable sources. In 2020, energy from renewables made up 19.6%. The share for heating and cooling developed similarly, starting at 7.4% in 2004 and reaching 15.2% in 2020. While there also was an increase in the transport sector from 2004 to 2007, the share stagnated since then. The strongest growth can be observed in the share of consumed electricity, which grew from 9.4% in 2004 to 45.4% in 2020.

The major contributors to the consumed electricity from renewable sources in Germany are wind and solar power, which are both naturally fluctuating energy sources. In 2020, on-shore and off-shore wind power and solar power had shares of 41.2%, 10.8% and 20.3%, respectively [2]. Hence, with 45.4% consumed electricity from renewables, overall $45.4\% \cdot 72.3\% = 32.8\%$ of the consumed electricity originated from fluctuating sources. This percentage will increase in the future, as the current legislation, in the form of the German Renewable Energy Sources Act (Erneuerbare-Energien-Gesetz) [3], aims for a share of renewables of at least 80% in 2050 and 65% in 2030. With an increasing percentage of electricity produced from wind and solar power, the traditional paradigm of electricity supply following electricity demand is overthrown, since their availability is not guaranteed. While nuclear and fossil power plants may balance out the random variations in supply, maintaining them solely as reserve capacity is costly and using them would reduce the share of renewables in the energy consumption. The consequence is the need for buffering energy and influencing energy consumers, for instance, to shift their demand from periods with low electricity production to periods where there is excess energy. Such measures for influencing the demand side in the electrical grid, that is, the consumers of electricity, are provided by demand side management (DSM).

When there is a high share of volatile electricity supply, it is necessary to coordinate many consumers simultaneously in order to keep the balance between supply and consumption. For the overall balance it does not matter how the perceived change in consumption is achieved by the demand side. Increasing local electricity production, for instance, by starting a combined heat and power plant (CHPP), discharging a battery energy storage system (BESS), or stopping the operation of a heat pump (HP), all lead to a decreased amount of power drawn from the electrical grid. Devices able to provide such flexibility are called distributed energy resources (DERs). Another recent development intensifying the need for DSM measures is the transition towards electric mobility. Even

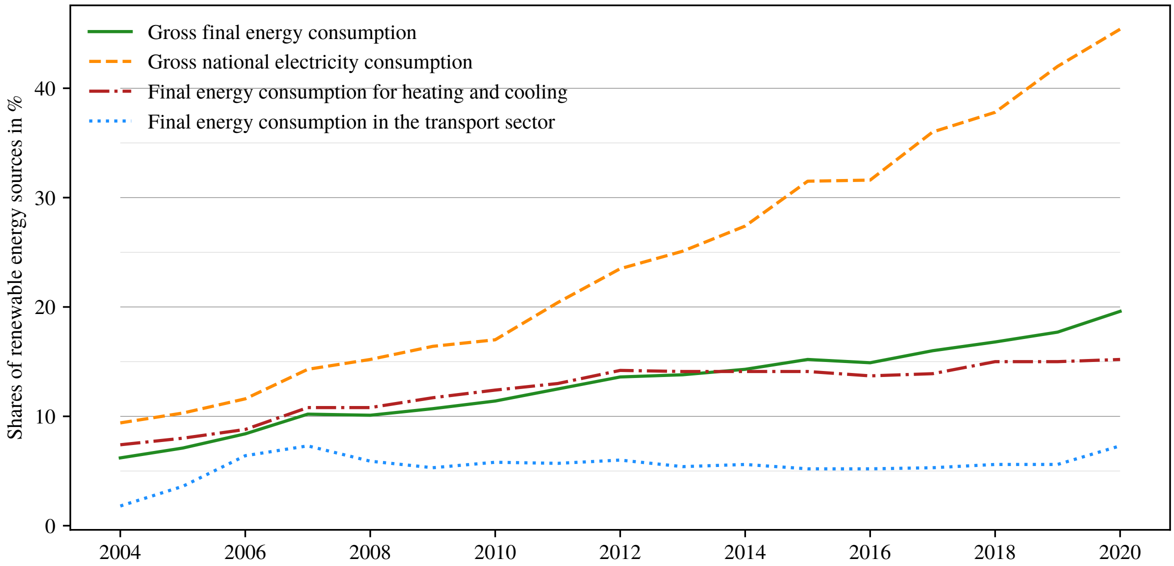

Figure 1.1: Share of renewable energies in Germany [2].

when battery electric vehicles (BEVs) are charged at home, the electric vehicle supply equipment (EVSE) needs comparatively high power for a prolonged time.

There are various potential options for coordinating DERs, ranging from decentralized mechanisms with minimal communication, to the centralized and direct control of the demand side. Most coordination approaches require models of the influenced DERs or descriptions of how they can be influenced, i.e., a model of their flexibility (a definition of the term “flexibility” is provided in Section 2.1.3). The question of how flexibility can be modelled and communicated has been occurring in various research projects. For instance, in the German research project *grid-control* **[310]**, multiple buildings were operated by automated building energy management systems (EMSs), which needed to describe their flexibility to some external entity trading on energy markets. Another example is the project *C/sells*, funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) as part of the SINTEG (Smart Energy Showcases — Digital Agenda for the Energy Transition) program[1]. In *C/sells* [4], the energy system is understood as a network of cells, such as smart buildings, city districts or even entire regions. How these cells are modeled and coordinated are two of many questions investigated in the project.

Standardized models are one way of describing DERs and their flexibility. While there are different propositions (see Chapter 4), such a standard model does not yet exist. Furthermore, even when there is a standard model, it is necessary to parameterize the model for every single DER in order to capture the individual local constraints and circumstances. An alternative could be the usage of automatically learned models, which are trained from observations. The automated generation of models has the potential of making manual model specification, implementation, and parameterization superfluous. By describing how a DER behaves, the learned model acts as a surrogate for the actual DER. Learning surrogate models for DERs and using them to influence the DERs' operation could simplify the process of making the flexibility of DERs available to third parties, such as aggregators

[1] `https://www.sinteg.de/en`

or system operators. Furthermore, with an automated process in place, there is the potential of gaining influence over many DERs in a short time frame, due to the reduced work needed.

The question of how the flexibility of DERs can be communicated and the goal of learning models for DERs represent the core of this thesis. The field of machine learning provides many types of learnable models. However, not every approach is suitable in the same way. Due to their versatility, in this thesis, artificial neural networks (ANNs) have been selected as a promising subject of investigation. In the following, the exact research questions are presented and explained, the contributions of this work are described, prior publications and their usage throughout the thesis are summarized, and the structure of the document is outlined.

1.2 Research Questions

The investigated subjects are divided into three major research questions **RQ1**, **RQ2** and **RQ3**. For each of these, subquestions have been formulated in order to structure the investigation systematically. The individual research questions build upon one another.

RQ1 *Which models can be used for communicating flexibility?*
The communication of the flexibility of DERs is investigated with the goal of developing a general framework for classifying exploitation and modeling approaches. This framework is the basis for deriving possible ways of modeling DERs with ANNs during the investigation of **RQ2**.

RQ1.1 *What are the motivations for communicating flexibility?*
Investigating the motivations may unveil aspects relevant for the subsequent development of a framework and the investigation of possible ANN-based surrogate modeling approaches.

RQ1.2 *How can flexibility be communicated and exploited?*
The communication of flexibility is generally a step in its exploitation. Therefore, the intended exploitation process, to a certain extent, dictates which information is required. Aiming to identify the general approaches for exploiting flexibility, this question lays the basis for the development of the framework.

RQ1.3 *Which modeling approaches do exist and how can they be categorized?*
Based on the results of **RQ1.2**, general modeling approaches for describing DERs are explored in a literature review and classified, resulting in the framework sought-after by **RQ1**. Additionally, for the purpose of demonstrating its application, the framework is used to classify exemplary exploitation approaches suggested in the research project *C/sells*.

RQ2 *What is the quality of approximation ANN-based surrogate models can achieve?*
Automatically learned models are only an alternative to manually specified ones, if they approximate the behavior of the real DERs sufficiently well. With this research question, the quality of approximation of ANN-based surrogates is investigated.

RQ2.1 *What are the advantages and disadvantages of using surrogate models?*
Exploring the general motivations and challenges associated to surrogate modeling may uncover requirements, issues associated to their implementation, possible solutions to these issues, and best practices. Additionally, the statements collected from the literature are translated to the topic of modeling flexibility.

RQ2.2 *How can ANNs serve as surrogates for the flexibility of DERs?*
This question entails two separate aspects. Firstly, the question of how ANNs can act as surrogates, which is answered on the basis of the framework developed during the investigation of **RQ1**. Secondly, the specification of the exploitation process, including how data is acquired and models are trained.

RQ2.3 *What is the quality of the trained ANN-based surrogate models?*
The goal is the assessment of how well ANNs can approximate the flexibility of DERs. For this purpose, multiple experiments are conducted in order to create a baseline for the performance of ANN-based flexibility models. The results are not only compared to other ANN-based models, but also to competing surrogate modeling approaches found in the literature.

RQ3 *How adequate are the different ANN-based modeling approaches for controlling schedules of flexibility providers?*
In order to make targeted use of DERs in a DSM application, the description of their flexibility must allow the derivation of some sort of control signal. Control, in this context, does not only refer to direct control, but also any other type of signal influencing their operation. Following up **RQ2**, this research question aims at investigating the suitability of the different identified approaches for the application in an exploitation scheme.

RQ3.1 *Which criteria can be used for assessing the adequacy of a modeling approach?*
In order to compare different ANN-based surrogate modeling approaches, it is necessary to identify evaluation criteria.

RQ3.2 *What is the assessment of the individual ANN-based modeling approaches?*
Using the previously identified criteria, the different approaches are assessed, including an outline of how the utilization of flexibility could be optimized. Afterwards, the general suitability of ANN-based flexibility surrogates for controlling DERs is demonstrated for one selected approach.

Subquestions **RQ1.1**, **RQ2.1** and **RQ3.1** are answered with the help of the literature, since they explore the motivations and serve as an introduction to the respective subject.

1.3 Contributions

This thesis continues and systemizes our previous published work on the subjects *flexibility* and *ANN-based surrogate models for DERs*. The utilization of already published concepts and results is disclosed in Table 1.1, and wherever previous insights are used in the text (see the subsequent section for more details). Overall, the major contributions of this thesis are:

Development of a framework for classifying approaches for the exploitation of flexibility: An extensive literature review is presented in Chapter 4, investigating and classifying the modeling of DERs in the literature, resulting in a list of general modeling approaches. The review is structured according to so-called *exploitation patterns*, which have been taken from one of our previous publications and refined in the review process. The updated patterns allow for a more precise distinction of approaches. Additionally, criteria for telling the patterns apart are presented. With the resulting framework, approaches for the exploitation of flexibility can be distinguished in terms of their modeling approach, abstractness, exploitation pattern, required information exchanges, and similarity. An exemplary application for a selection of approaches proposed in the research project *C/sells* is presented in Section 4.5.

Derivation of approaches for learning ANN-based surrogates of DERs: In order to provide guidance for future research and applications, the general process and requirements of ANN-based surrogate modeling are explained in Section 5.2. This includes an outline of how the data for training the models can be acquired. Furthermore, possible challenges and ways of overcoming them are discussed. Then, throughout Section 5.2, several approaches for encoding the flexibility of DERs into ANNs are derived and explained, and the most promising ones are pointed out. The results provide a basis for future research aiming at automating the generation of DER models with the help of ANNs.

Establishment of a baseline for the quality of approximation: By selecting a modeling approach that is able to emulate most remaining approaches and evaluating its performance, a baseline for the approximation quality achievable with ANN-based surrogates is established in Section 5.3. The results comprise multiple different experiments and performance indicators in order to provide a comprehensive picture. It is concluded that ANN-based surrogates in principle can be used for modeling DERs when a sufficient amount of data is available. Furthermore, potential trade-offs, for instance, between a comprehensive description of flexibility and the rate of errors, are pointed out.

Qualitative analysis of the applicability of ANN-based surrogates: With respect to two exemplary optimization problems it is outlined in Chapter 6 how each ANN-based surrogate modeling approach could be used to solve each optimization task. This provides researchers with approaches and guidelines to apply the models in the future. Furthermore, each approach is assessed in regard to its complexity and scalability, which provides further guidance in identifying and selecting the best suitable ANN-based surrogate modeling approach for a given problem.

First implementation and evaluation of a scheduling task: In Section 6.4, one of the two exemplary scheduling problems discussed in Chapter 6 is solved with the help of an ANN-based surrogate model. The results prove the general capability of ANNs to provide and predict information sufficiently detailed and accurate to solve simple scheduling tasks.

Software framework for evaluating ANN-based surrogates: The software framework developed in order to conduct the evaluation is published on GitHub at `https://github.com/kfoerderer/ANN-based-surrogates` and, together with this document, in the KITopen repository, alongside the experiments, data and trained models. It comprises simulation models, optimization models, an implementation of the "state-based simulation" surrogate (see Chapter 5), functionality for its training, and the algorithms presented in Section 6.4. With the framework and data being freely available, researchers can validate the results, and start their own experiments, for instance, exploring the untested approaches.

1.4 Previous Publications

Table 1.1 summarizes all of my scientific publications prior to this thesis and specifies where in this document which contents and results are incorporated. The major contributions by me to each paper are additionally pointed out in each summary. The utilization of insights gained in the past is also detailed in the introductions of each chapter, and, of course, referenced in the text. Throughout the document, the listed publications are formatted in boldface (e.g., **[314]**), while all other citations are not (e.g., [1]). By making this distinction, it is easily visible whether an older, own work is referenced or not.

Table 1.1: Publications prior to this thesis.

Reference	Summary	In this thesis
[314]	An abstract outlining the EMS developed for the research project *grid-control*. The EMS describes and communicates flexibility with the help of time-varying corridors for power and energy. **Contribution:** EMS and flexibility model.	The paper appears in the search results returned by Scopus and is hence discussed in the literature review in Chapter 4.
[316]	In this publication, we propose a definition for the term "flexibility" in the context of smart buildings and smart grids, present generalized patterns for the exploitation of flexibility, and shortly address the associated communication effort. **Contribution:** proposed definition.	The definition of flexibility is cited in Section 2.1.3 and used throughout this thesis. The exploitation patterns were the starting point for deriving the refined patterns presented in Section 4.2.
[313]	We propose the utilization of ANNs as surrogate models for DERs and suggest different approaches for the implementation, namely, a classifier, a generator, a repair mechanism, and a price-based load schedule forecast. **Contribution:** concept and approaches.	As this paper is the first presentation of the idea of ANN-based surrogates for modeling flexibility, it is referenced throughout this thesis. The proposed modeling approaches are incorporated into Section 5.2 and the paper can also be found in Chapter 4, where it is part of the literature review and the basis for an illustration.
[312]	In this paper, we evaluate the individual approaches that we proposed in **[313]** by means of multiple experiments. We conclude that the results show a good quality of approximation for the classification and generation approach. However, this is only a first proof of concept, and we merely outline possible ways of searching for feasible load schedules, which is a prerequisite to conducting optimizations. **Contribution:** concept and approaches.	The simulation models and parameters presented in Section 5.3 and used for the evaluation are based on those presented in **[312]**. Furthermore, the best results are cited in order to compare them to those achieved in this thesis.
[315]	We present a process for creating forecasts of power flows in low voltage grids and evaluate the quality of the predictions. The process was implemented in the project *grid-control* and, among other things, makes use of forecasts provided by the EMS outlined in **[314]**. **Contribution:** EMS specific forecasts.	Not used in this thesis.

Table 1.1 – continued from previous page

Reference	Summary	In this thesis
[309]	In the project *C/sells* we further discussed the definition of the term “flexibility” and related concepts. This paper presents the results and provides a set of flexibility related definitions. **Contribution:** discussion and feedback.	Referenced in the section on fundamentals.
[307]	The integration of automated EMS into the German Smart Meter Gateway architecture is the subject of this paper. Aside from illustrating the options for integrating EMS and receiving signals for implementing DSM measures, we provide an overview of the architecture and the most important configurations related to energy management. **Contribution:** equal collaboration.	Not used in this thesis.
[310]	The final report of project *grid-control*. In the report, the project, developed architecture and components, as well as the results are summarized. One of the presented components is the EMS outlined in **[314]** and discussed in **[315]**. **Contribution:** EMS and related results.	This publication is referenced when project *grid-control* is mentioned. Also, some data and parameters used in the evaluation originate from this project.
[311]	This abstract points out the lack of standardized environments for the testing of building energy management algorithms and motivates the development of such environments. **Contribution:** discussion and feedback.	Not used in this thesis.
[308]	In this paper, we present and evaluate a method for encoding the flexibility of DERs into ANNs. The approach is inspired by Markov processes. The evaluation is based on simulation models derived from **[312]** and the model performance is measured via the share of generated feasible load schedules. **Contribution:** concept and experiments.	The state-based approach for encoding flexibility is also presented and evaluated in this thesis. Therefore, it is referenced throughout this document. In comparison to the paper, a few adaptations are made, more DERs are evaluated, and more criteria are analyzed. Differences between the implementation in the paper and the implementation in this thesis are pointed out in the text.
[306]	While **[307]** provides only a theoretical analysis, in **[306]** we present a practical implementation. In the paper, we describe the functionality provided by an exemplary EMS, point out challenges that had to be overcome, and illustrate how the integration into the Smart Meter Gateway architecture is achieved. The implementation was tested by influencing the charging process of a BEV. **Contribution:** equal collaboration.	Not used in this thesis.

1.5 Structure

Before jumping into the subsequent chapters, in the following the structure of this thesis is outlined. Chapter 2 gives an overview of fundamental concepts and approaches, used and referenced throughout this thesis. The state of the technology is presented in Chapter 3. It focuses on the topic of communicating the flexibility of DERs. A broader overview of the literature and the subject of modeling flexibility is provided subsequently in the form of a literature review in Chapter 4. Chapter 4, Chapter 5, and Chapter 6 are dedicated to the three major research questions. Each chapter is devoted to one of the major questions and its associated subquestions. The thesis is concluded by Chapter 7, which summarizes the findings and provides an outlook for possible future research. Some further ideas and information are provided in the appendix.

All source code, including the implemented simulation and optimization models, the trained ANNs, and the logs from the training and evaluation runs have been published on GitHub at `https://github.com/kfoerderer/ANN-based-surrogates` and, together with this document, in the KITopen repository.

2 Fundamentals

This chapter summarizes concepts, methods, and algorithms utilized and mentioned throughout this thesis. It is not intended as an introduction to the named subjects, as this would easily go beyond the scope of this work. Instead, it clarifies the terminology, provides a basic understanding, and directs to learning materials. The chapter starts with a short introduction to smart grid related topics in Section 2.1, followed by a summary of control architectures in Section 2.2. In Section 2.3, general modeling concepts are introduced, before providing a concise overview of machine learning fundamentals in Section 2.4, and optimization approaches in Section 2.5. This chapter incorporates previous publications in Section 2.1.3, where the term “flexibility” is defined.

2.1 Smart Grids

This section summarizes the most important concepts related to smart grids and DSM. The primary source is Mauser [5], as it provides a comprehensive overview of smart grid related topics. The approaches investigated and developed in this thesis all belong to the domain of smart grids. Mauser proposes a smart grid definition considering all energy carriers, instead of focusing solely on electricity. A slightly generalized version of this definition is also utilized in this thesis:

Definition (Multi-energy smart grid)**.** “A multi-energy smart grid is an integrated energy grid comprising all energy grids that are used in the (distributed) generation, distribution, and consumption of energy using various energy carriers. It includes [. . .] technologies and entities that enable the intelligent interaction of all entities to provide [. . .] services, facilitate [. . .] functionality, and increase energy efficiency. The main characteristic is a two-way flow of information to and from all entities.” [5, pp. 38-39].

2.1.1 Ancillary Services

Ancillary services help to maintain a secure supply of electricity [6]. According to [5], typically, six services are distinguished. For a more elaborate description than provided in the following, please see [5].

Frequency control: Frequency control aims at keeping the grid frequency at its target value. There are three frequency control levels. However, the implementation varies from country to country. Please see [6] for an overview.

Primary control (frequency containment reserves, FCR [7]): Primary control has the purpose of stabilizing the frequency. It uses local sensors to detect deviations and reacts immediately and automatically [6].

Secondary control (automatic frequency restoration reserves, aFRR [7]): Once activated, the secondary reserve replaces the primary reserve in order to bring back the frequency and release the primary reserve [5, 6]. The secondary reserve is controlled automatically and centrally by the transmission system operator (TSO) [6].

Tertiary control (manual frequency restoration reserves, mFRR [7]): Tertiary reserve is controlled manually and releases the secondary reserve [6].

Voltage control: The voltages at different grid nodes vary depending on the local power flows and the grid topology. Voltage control has the goal to keep the voltages within an acceptable range [5]. TSOs are obliged to ensure that the voltage remains within an agreed range for transmission-connected DSOs and transmission-connected significant grid users [7]. The same applies for DSOs and their connected grid users [8]. Electricity generators, in turn, must follow given protection schemes in order to prevent the aggravation and limit the consequences of disturbances [9].

Reactive power control: Reactive power needs to be provided for technical reasons and can also be used for voltage control [5].

Phase balancing: Phase balancing aims at balancing the load between the different phases of AC grids [5].

Redispatch and congestion management: Congestions in the electrical grid happen when the infrastructure is overburdened. Redispatch means rescheduling power plants on a global scale, for the purpose of resolving congestions. Similarly, congestion management solves congestions on a more local scale by influencing the consumption and generation of electricity [5].

Restoration after power outages: After a power outage, the system has to be restored in an orderly fashion, to avoid new outages [5].

2.1.2 Demand Side Management

The demand side of the energy grid is typically found at the lowest voltage level and comprises all entities consuming energy. Influencing them is the goal of DSM. Nonetheless, the literature does not provide a consistent definition for DSM and the related topic of demand response (DR) [5]. Mauser [5] defines DSM and DR as follows:

Definition (Demand side management)**.** DSM describes "[. . .] all measures and methods that are applied at or influence the lowest level of the energy grids — the former demand side — for the benefit of the overall grid and energy system" [5, p. 40].

Definition (Demand response)**.** "DR refers to all measures that incentivize the demand side to adapt their consumption and generation because of additional costs, benefits, information, and education" [5, p. 40].

DR is therefore a part of DSM. An overview of DSM measures is provided by Figure 2.1, which was taken from [5]. There are different types of DR signals and aside from DR, DSM comprises energy efficiency and conversion measures. In this thesis, the measures themselves only play a subordinate role, since modeling approaches are investigated independently of the DSM measure. Per definition of DSM, a demand side manager (DSMgr) makes use of the listed measures and pursues the goal of influencing the energy consumption of the demand side. DSMgrs are not limited to a specific role. TSOs, distribution system operators (DSOs), as well as aggregators and other participants on energy markets, can all act as a DSMgr. How a change in consumption is achieved, that is, whether it

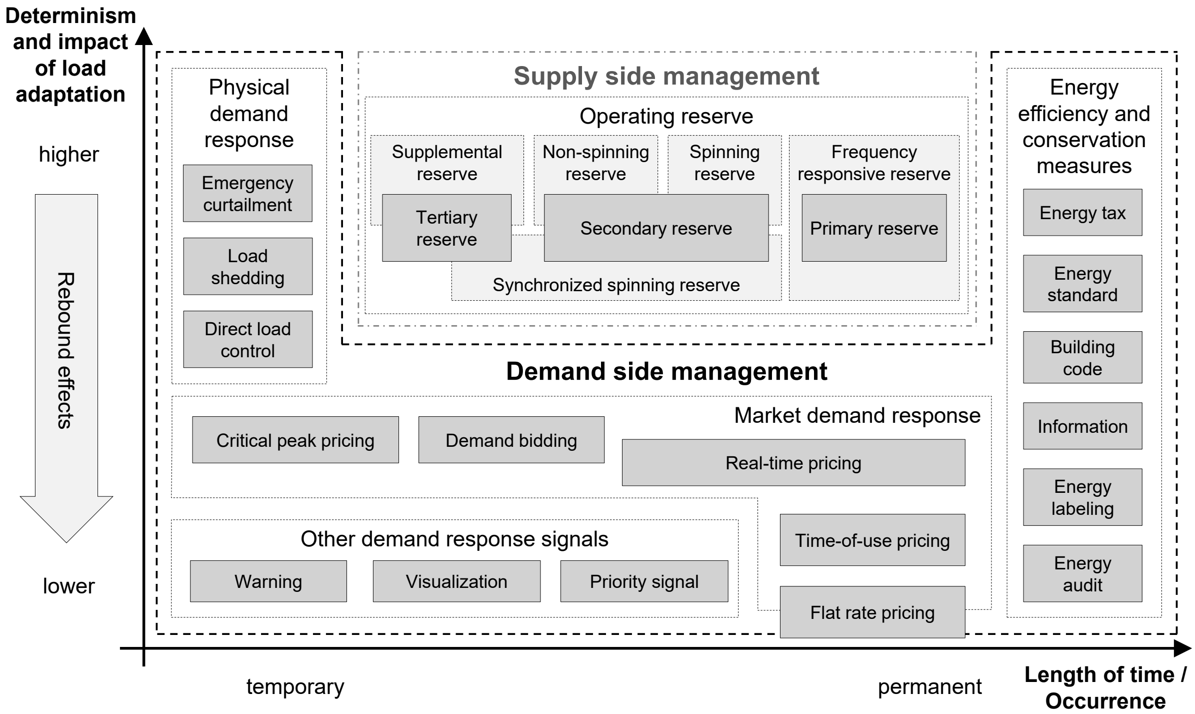

Figure 2.1: "Categories of measures of demand side management and operating reserve [. . .]" [5, p. 40].

is due to a change in local production, storage, or the actual consumption, does not matter to the DSMgr. Consumers with the additional capability of producing electricity are often called "prosumers".

As Figure 2.1 illustrates, there are different possibilities for a DSMgr to influence or control the demand side. Particularly popular in the literature are time-variable tariffs (time-of-use pricing) and dynamic tariffs (real-time pricing), as is shown in Chapter 4. In time-of-use pricing, there are periods with different predefined prices, for instance, peak and off-peak periods. The price only depends on the time period, all other circumstances are neglected. In dynamic pricing, in contrast, the prices can vary freely, reflecting the current market situation [5]. For more pricing schemes, please see [5].

A DSMgr using monetary incentives, such as a dynamic tariff, in order to influence the demand side may want to estimate the consequences of adapting energy prices. Price elasticities provide a simple solution for estimating changes in demand. The price elasticity is "[. . .] the percent change in quantity divided by the percent change in price" [10, p. 274], that is,

$$\epsilon = \frac{\frac{\Delta q}{q}}{\frac{\Delta p}{p}}.$$

Multiplying the elasticity with a relative price change yields the relative change in consumed energy. Vice versa, it is possible to estimate the required price change in order to achieve a certain change in consumption. Of course, this is only a very rough estimate, since it is assumed that the elasticity is a fixed parameter, which in reality it is not. In addition to the price elasticity, cross elasticities can be used to estimate the consequences of price changes in period j in regard to the consumption in period i:

$$\epsilon_{i,j} = \frac{\frac{\Delta q_i}{q_i}}{\frac{\Delta p_j}{p_j}}.$$

Signals demanding a restriction of the feed-in of decentrally generated power or requesting a reduction of consumption are another possibility to influence the demand side. Exemplary measures using such signals are emergency curtailment and load shedding. One way to implement reduction signals are quotas. Quotas are percentages of some defined reference power, such as a nominal power of a DER or an announced load schedule. While load quotas specify the maximum power allowed to be drawn from the grid, feed-in quotas set the maximum feed-in power **[310]**. Examples for their application can be found in the German research project *grid-control* **[310]** and its successor *flexQgrid*. Another exemplary application is the dynamic reduction of the solar power generation in Germany during grid congestions, which is mandated by § 9 of the Renewable Energy Sources Act [3] and currently implemented by restricting the maximum electricity generation to 0%, 30%, 60% or 100% of the installed nominal power.

2.1.3 A Definition of Flexibility

A term often found in the literature on smart grids and DSM is the word "flexibility". However, the definitions of "flexibility" in this context varies heavily. Often the definition has a certain use case in mind **[316]**. In **[316]**, we therefore propose a generic definition:

Definition (Flexibility)**.** "The flexibility of an energy system is the collection of valid combinations of system inputs and their state dependent outputs in terms of all energy carriers, i.e., all combinations that provide all mandatory energy services in a manner ensuring system stability." **[316]**

Here, the term "energy system" refers to any system supplying energy carriers, such as electricity, gas, and heat, or providing energy services, for instance, lighting and air conditioning. A general depiction of such a system is provided in Figure 2.2. With this understanding, the definition applies to single DERs, groups of multiple DERs, buildings, factories, cities, and many more. Systems combining multiple DERs are synonymously referred to as "aggregates" and "ensembles" throughout this thesis.

The above definition is inspired by the definition of the graph $G(f) := \{(x, f(x))|x \in \mathcal{X}\}$ of a function $f : \mathcal{X} \to \mathcal{Y}$. Analogously, flexibility can be expressed as $\{(x, f(x))|x \in \mathcal{X}\}$ with x being the system inputs, $\mathcal{X}$ the inputs required to fulfill the mandated energy services, and f specifying the resulting outputs. In contrast to the graph definition, the set $\mathcal{X}$ and function f depend on additional variables specifying the system state. Moreover, when there are N energy carriers and T time steps to be considered, the input x and the resulting output $f(x)$ are matrices of dimension $N \times T$. With this set at hand, someone seeking to exploit flexibility has not only a list of options of how the associated system may be operated, but also knows the consequences in terms of consumed and supplied energy carriers.

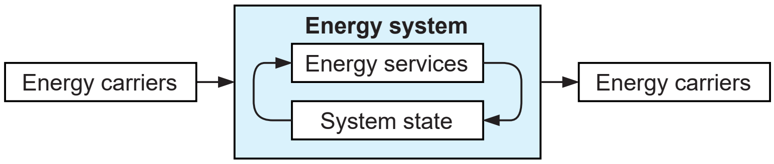

Figure 2.2: General depiction of an energy system. Based on **[316]**.

When looking only at electricity, system input and output are both electric load schedules. A load schedule is a sequence of planned average loads for consecutive time steps of a fixed length. Therefore, system input and output specify the average consumed (input) and provided (output) electric power. In case of a single coupling point, instead of considering input and output separately, it is sufficient to look at the total load schedule, that is, the sum of input and output $x + f(x) = (x_1 + f(x)_1, \ldots, x_T + f(x)_T)$. The result is a description of the flexibility in the form of load schedules. Each schedule in the resulting set can be achieved by the associated energy system, that is, each schedule in the set is feasible. Synonymously, feasible schedules are sometimes called valid or acceptable. A schedule that is not in the set is therefore infeasible, which means that the energy system is unable to reproduce the schedule given its current state and operational restrictions. However, in practice, generally only a subset of the true set of feasible schedules is known. Nevertheless, if provided with a feasible schedule, an energy system can derive which actions it should take in order to reproduce this target.

Please note that considering only electric power as system input and output does not mean that other types of useful energy are neglected. For determining the flexibility of an HP, for instance, it is still necessary to consider the thermal energy demand it has to satisfy. Furthermore, in this thesis, it is distinguished between load schedules and load profiles. While a load schedule consists of planned loads, the term load profile refers to a sequence of measured average load values.

We derived more specific definitions of flexibility and related concepts in the research project *C/sells*, in order to allow for a more detailed distinction **[309]**. However, for the topics investigated in this thesis, the general definition provided in this section is sufficient. Moreover, we also based our previous work on ANN-based surrogate modeling of DERs on this very definition **[313, 312, 308]** and other authors share a similar understanding, e.g., [11, 12, 13, 14, 15].

2.2 Communication Relationships

Influencing or controlling the demand side requires some sort of coordination mechanism. A major element in designing such a mechanism is the definition of communication relationships, that is, specifying who communicates with whom. Furthermore, it is necessary to look at what information is communicated.

2.2.1 Control Structures

In the field of control theory, it is distinguished between multiple control structures. The IEC 60050 series, that is, an encyclopedia by the International Electrotechnical Commission (IEC) which is also published online [16], defines centralized, decentralized, and hierarchical control structures. It is further possible to define a distributed control structure [17]. The definition of each is provided in the following:

Definition (Centralized control structure)**.** A centralized control structure is a "control structure with interconnected subprocesses, in which each partial control equipment takes into account the information of all subprocesses to form its output information" [16, reference number: 351-55-09]

Definition (Decentralized control structure)**.** A decentralized control structure is a "control structure with interconnected subprocesses in which each partial control equipment takes into account only the information from its associated subprocess to form its output information" [16, reference number: 351-55-10]

Definition (Hierarchical control structure)**.** A hierarchical control structure is a "control structure with several control levels placed one over the other, in which the control equipment assigned to a higher level coordinates the work of the control equipment assigned to the next lower level, providing for instance pre-determining control tasks, command variables, reference variables or final controlled variables" [16, reference number: 351-55-11]

Definition (Distributed control structure)**.** A distributed control structure is a decentralized control structure, but with controllers exchanging information [17, p. 21].

In summary, in a centralized structure, processes are controlled taking all information into account, including those of other processes. A decentralized structure, in contrast, has multiple controllers, where each controller only uses the information provided by the associated processes. If these controllers exchange information instead, a distributed structure is created. Finally, the hierarchical structure stacks decentralized control structures on top of each other until a central root is reached.

2.2.2 Stigmergy

Stigmergy is a concept for decentralized control derived from the behavior of social insects, such as ants and wasps. It explains how they achieve coordination, even though each individual acts autonomously [18].

Definition (Stigmergy)**.** "The concept of stigmergy explains the processes and self-organized behavior which results from the indirect communication between individuals through anonymous alterations on the environment. As a consequence of these alterations cooperation and coordination emerge spontaneously which enables the system to achieve global objectives in a self-organized manner." [18, p. 30]

This means that individuals base their actions on their perception of the environment and the only way of communication is the manipulation of said environment. In general, there are two ways of manipulating the environment. Firstly, leaving signs or markers as a signal for other individuals [18]. Secondly, physical modification of the environment towards some global objective [18]. The anonymous alteration is a key point in this definition. If some coordination mechanism allows determining actions of other individuals, it is not stigmergic. This distinction is relevant later in this thesis.

2.3 Modeling

According to the Oxford Dictionary, a model is "a simple description of a system, used for explaining how something works or calculating what might happen, etc.". A summary of modeling related concepts is given in this section, including an introduction to surrogate modeling.

2.3.1 Abstractness

Abstraction is an elementary step in the process of creating a model. When comparing different models, it becomes apparent that some models are more abstract than others. It may even be impossible to directly compare models, due to different forms of abstraction being employed (see, for instance, this thesis). In order to make distinctions, it is necessary to specify the concept of abstractness and define an abstractness relation. A series of formal definitions for comparing the abstractness of computer programs is derived in [19]. They are based on three core requirements an abstraction relation should satisfy:

1. "Abstract specifications say what a program does without necessarily saying how it does it." [19, p. 443]
2. "Abstraction is a process of generalization, removing restrictions, eliminating detail, removing inessential information (such as the algorithmic details)." [19, p. 443]
3. "Abstract specifications have 'more potential implementations', moving to a lower level means restricting the number of potential implementations." [19, p. 443]

While the formal definitions provided in [19] are not directly applicable to the topic of modeling, the requirements above are. In regard to modeling this means that a model A is considered to be more abstract than model B if it is obtained by removing details, such as constraints, from model B. As a consequence, simplifying a model and making it applicable to a broader range of systems is considered abstraction. This understanding conforms to the definition of "abstraction" as "the process of deciding on the appropriate level of detail for whatever problem is at hand [. . .]" [20, p. 33] in the context of mathematical modeling.

2.3.2 White-Box, Black-Box, and Grey-Box Models

The literature utilizes different criteria for classifying models. One such classification is the distinction of white-box, black-box, and gray-box modeling [21, 22].

White-box model: A white-box provides a clear view of the system inside, that is, all relevant dynamics of the system can be seen. The associated white-box model is a detailed and complete description of the system dynamics [21, 22]. It is therefore comparatively easy to interpret results obtained from such a model [22]. However, formulating physical relations requires extensive knowledge [22].

Black-box model: A black-box completely obstructs the view of the system inside. Only the system input and output are visible. A black-box model is a statistical model fitted to approximate the system dynamics in terms of the observed outputs [21, 22].

Grey-box model: A grey-box is the combination of a black-box and a white-box in the sense that some parts of the system inside are visible while others are not. A grey-box model is therefore a hybrid approach, using combinations of white-box and black-box models [21, 22]. The three major options to couple white-box and black-box models are the estimation of physical parameters of a white-box model using a black-box model, the generation of a black-box model from data produced by a white-box model, and replacing single components of a white-box model with black-box models [22]. The third option is especially relevant for components that are difficult to model, like user behavior [22].

2.3.3 Surrogate Modeling

Surrogate models, which are also known as approximation models [23], reduced-order models [24], regression surfaces [25], meta models, emulators, or response surface models, are a tool for reducing computational expenses [26] by approximating input-output relationships [23, 24]. They are applied in different contexts, including optimization, dynamic processes, feasibility analysis, parameter estimation, sensitivity analysis, and scheduling [25]. Due to the focus on input and output data, surrogate models are black-box models. It is possible to distinguish local and global surrogate modeling [26, 25]. A local surrogate is a small and low fidelity model used in optimization [26]. It provides rough approximations of the optimization surface [26]. A global surrogate, on the other hand, is created with the goal of constructing an accurate model with high fidelity [26]. Generating surrogates as descriptions of DERs, like it is investigated in this thesis, falls into the global surrogate modeling category.

Formally, the goal of global surrogate modeling can be defined as follows [26]: Let $f : \Omega \to \mathbb{C}^n$ be a multivariate function, mapping from the domain $\Omega \subseteq \mathbb{R}^d$ to the set of all n-tuples of complex numbers $\mathbb{C}^n$, $\mathcal{X} = \{x_1, \ldots, x_k\} \subset \Omega$ be a set of sample points of which the function values are known, and $\mathcal{S}$ be a given space of functions $s : \mathbb{R}^d \to \mathbb{C}^n$. Then, the goal is to find the best approximation $s^* \in \mathcal{S}$ of f according to some given criterion ξ, that is, $s^* = \arg\min_{s \in \mathcal{S}} \xi(s)$. Additionally, it is assumed that $f(x)$ is expensive to compute so that it should only be used parsimoniously to compute further sample points.

The field of machine learning provides many approaches for generating, selecting, and evaluating surrogate models. A concise introduction can be found in the subsequent section, including more information on selection criteria ξ.

2.4 Machine Learning

The capability of an artificial intelligence to extract its own knowledge from raw data is known as machine learning [27]. In this context, "learning" can be defined more formally:

Definition (Learning)**.** "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." [28, p. 2]

Based on [27], a short overview of experiences E, tasks T and measures P is given in the following. For a more elaborate introduction, please see [27]. Afterwards, ANNs and the support vector data description (SVDD) are introduced.

2.4.1 Experience

Typically, experiences are data points, which are given by some dataset [27]. In some cases, like reinforcement learning (RL), data points are observed dynamically. Each data point, also called example [27] or sample [29], is a collection of features and usually represented as a vector $x \in \mathbb{R}^n$ [27]. Depending on what should be learned from the data, it is possible to distinguish different types of learning algorithms:

Supervised Learning: In supervised learning each data point is associated with a response [29], that is, a label or target value [27]. The goal is to predict the response from the features provided by the data point.

Unsupervised Learning: Unsupervised learning aims at learning useful properties from the data points, such as the underlying probability distribution [27], and understanding the relationships between individual variables or observations [29].

Semi-Supervised Learning: In semi-supervised learning, a response is only known for parts of the data points [29]. The goal is to make use of the entire dataset, even though responses are missing [29].

Performing well on previously unseen inputs, that is, being able to generalize, is a central challenge in machine learning [27]. To evaluate how well an algorithm generalizes, it must be confronted with data points it has not seen during its training. When the data points originate from a dataset, the data is typically split into smaller subsets, in order to allow the selection and validation of a model [27, 29]: firstly, the training set, which comprises most of the data and provides the samples for learning a model. Secondly, if multiple models with different hyperparameters are generated, a validation set. Hyperparameters are parameters of the model or algorithm which are not learned from the data [27], e.g., the learning rate of the training algorithm or the depth of an ANN. With the help of a validation set, it is possible to evaluate hyperparameter choices and select the best performing combination. Finally, the test set for the actual evaluation of the model.

2.4.2 Tasks

The task defines the purpose of learning. Learning is what enables the algorithm to fulfill the task [27]. Common machine learning tasks include [27]:

Classification: Learning a function $f : \mathbb{R}^n \to \{1, \ldots, k\}$, which assigns data points x to one of k categories [27]. In case only two classes are distinguished, they are often labelled “positive” and “negative”.

Regression: Predicting a numerical response with a learned function $f : \mathbb{R}^n \to \mathbb{R}$ [27].

Structured output: Learning to derive a vector of multiple inter-related outputs, such as a textual description of a picture, from the given inputs [27]. This includes:

Transcription: Transcribing data, such as images, into text [27].

Machine translation: Translating a sequence of symbols from one language to another language [27].

Anomaly detection: Identifying unusual or atypical events or objects [27].

Synthesis and sampling: Generating new data points that are similar to those faced during training [27].

Imputation of missing values: Predicting features x_i missing in a data point $x \in \mathbb{R}^n$ [27].

Denoising: Predicting either a clean example $x \in \mathbb{R}^n$ or the probability distribution $p(x|\tilde{x})$ associated with a corrupted example $\tilde{x} \in \mathbb{R}^n$ [27].

Density estimation or probability mass function estimation: Learning a function $p_{model} : \mathbb{R}^n \to \mathbb{R}$ approximating the probability density function or the probability mass function [27].

There exist more tasks than those listed above and also many variations. Take, for instance, the multi-label classification, in which more than one category can apply to a data point. Another example is tasks in which different input features are missing [27]. One way to deal with missing input features or multiple output values is to train a set of models instead of a single one. However, learning different tasks together, which is called multi-task learning, can help to improve the generalization capability [27].

2.4.3 Performance measures

Performance measures quantify how well a task is fulfilled. There is a multitude of measures, e.g., the accuracy for classifiers or the mean squared error (MSE) for regression models, but not every measure is suitable for every task [27]. During the training of a model, the objective is to improve the value of a target function, that is, the quantified performance given by the measure. If the goal is minimization, the target function is also called a loss function, cost function, or error function [27]. The loss functions utilized in this thesis are introduced in Section 2.4.4. Additionally, the following standard measures can be found in the evaluation:

Share of true/false positives/negatives: In a classification task with a "positive" and "negative" class, the model output is either a true positive (TP), false positive (FP), true negative (TN), or false negative (FN), depending on the classification result (positive or negative) and whether it is correct (true) or not (false). The individual share is calculated by dividing the respective number by the total amount of tested samples.

False positive rate: The false positive rate (FPR) is the number of false positives in relation to the actual number of negative samples [29]:

$$\frac{FP}{\mathrm{FP} + \mathrm{TN}}.$$

False negative rate: The false negative rate (FNR) is the number of false negatives in relation to the actual number of positive samples [29]:

$$\frac{FN}{\mathrm{FN} + \mathrm{TP}}.$$

Mean absolute error: Let vector $y = (y_1, \ldots, y_N)$ contain the target values, and $x = (x_1, \ldots, x_N)$ comprise the associated model predictions. Then, the mean absolute error (MAE) is given by

$$l^{\mathrm{MAE}}(x, y) := \frac{1}{N} \sum_{i=1}^{N} |x_i - y_i|.$$

2.4.4 Artificial Neural Networks

ANNs are collections of so-called neurons, which are typically arranged in interconnected layers. Each neuron receives an input vector x, computes a function $f(x, \theta)$ of which some parameters θ are learned, and passes on the scalar output. In most ANNs, neurons perform affine transformations $w^\intercal x + b$ with learned parameters w and b, followed by a fixed, non-linear activation function g [27]. Such a neuron is shown in Figure 2.3. There are many activation functions, including the rectified linear unit $g(z) = \max\{0, z\}$, sigmoid activation function $g(z) = \sigma(z) = \frac{1}{1+e^{-z}}$, and the hyperbolic tangent activation function $g(z) = \tanh(z)$ [27]. The ANNs trained in the context of this thesis make use of the following activation functions:

Swish activation function: The swish activation function is given by $g(z) = z \cdot \sigma(\beta z)$ [30]. It is utilized in this thesis since it proved to work well across a variety of different tasks and datasets [30]. The additional parameter β is treated as a fixed hyperparameter.

Softmax activation function: The softmax function relates the elements z_i from the vector $z = (z_1, \ldots, z_N)$ to the vector itself:

$$g(z, i) = \text{softmax}(z)_i := \frac{e^{z_i}}{\sum_{j=1}^{N} e^{z_j}}.$$

One application of this function is the modeling of probability distributions over discrete values [27]: Let there be n different discrete elements, each assigned with an index 1 to n, and an ANN providing n outputs, that is, one output for each index. The task of the ANN is to predict the index i based on the input x. When the ANN output is interpreted as a vector of unnormalized log-likelihoods, the softmax function yields a vector $\hat{y}$ of predictions $\hat{y}_i = P(y = i \mid x)$ [27], specifying the probability of each discrete element.

In order to simplify the presentation of neural networks, neurons are typically bundled to layers and calculation steps are expressed in matrix notation. A layer of neurons is, for instance, given by $g(W^\intercal x + b)$, where the matrix W and vector b comprise the weights of each individual neuron and activation function g is applied to each element of the resulting vector. Aside from vectors and matrices, tensors find their application in ANNs. Like a matrix, but without being limited to two dimensions, a tensor is an array of numbers arranged on a regular grid with a variable number of axes [27]. Illustrations of ANNs are often simplified in a similar way, which means that instead of showing individual neurons, they depict tensor calculations, network layers, or entire ANN modules, i.e., sets of layers and calculation steps (see for example [27, 31, 32, 33]). Such a simplified illustration of network layers $g(W^\intercal x + b)$ is also depicted in Figure 2.3. The input layer holds the data passed to the ANN and the output layer the computed result. All layers in between are so-called hidden layers.

Neurons and layers can be arranged and connected in various manners. Depending on their topology and characteristics, it is possible to distinguish different types of ANNs. The most basic ones are the feedforward neural network, recurrent neural network, and convolutional neural network (see [27]). In feedforward neural networks, which are also called multilayer perceptrons, information flows only in forward direction [27]. The ANN depicted in Figure 2.3 is an example for a feedforward neural network. Recurrent neural networks (RNNs) are used for processing sequential data. They are characterized by some information flowing backwards and thereby forming a cycle in which the present value of a variable influences its future value [27]. The cycled information is called the “state” of the system [27]. Convolutional neural networks process data in grid-like topologies, like time-series or images [27]. They are neural networks that make use of convolution instead of general matrix multiplication in one or more layers [27]. For a comprehensive explanation of these types of ANNs, please see [27]. The ANNs

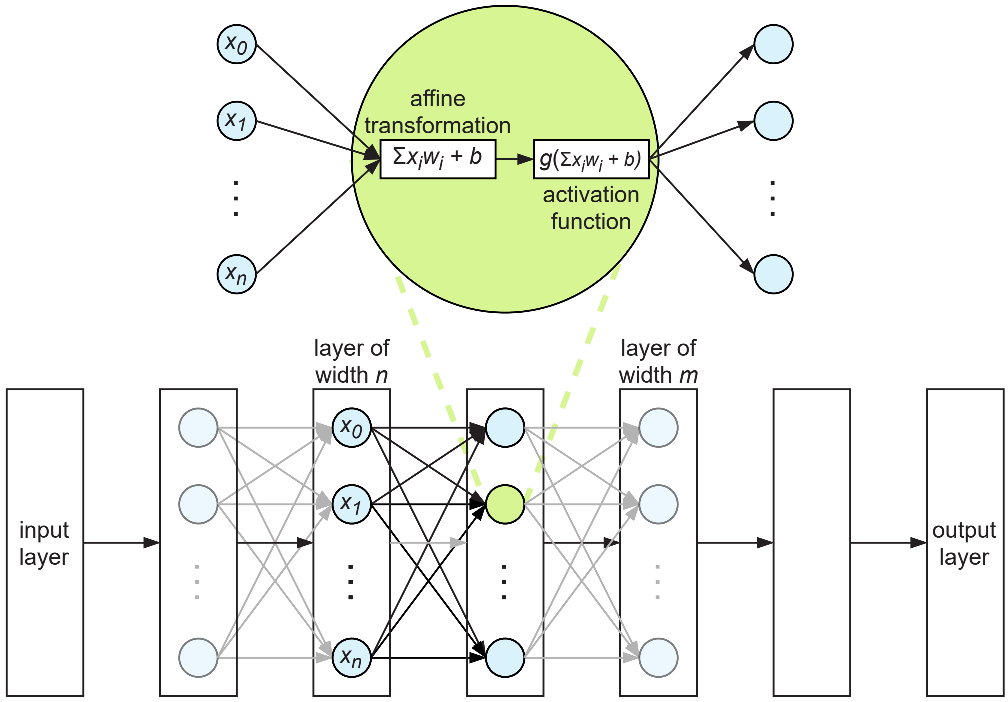

Figure 2.3: Illustration of an artificial neural network. A neuron receives inputs from its predecessors, performs computations and provides the result to its successors. For large ANNs, layers of neurons are depicted instead of showing thousands of neurons and their connections.

generated in the context of this thesis are all feedforward neural networks. However, some are used in a recurrent fashion.

ANNs are a very powerful tool for modeling. The universal approximation theorem states that "[...] standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available" [34], where the term "squashing function" means a non-decreasing function $f : \mathbb{R} \to [0, 1]$ with $\lim_{x\to-\infty} f(x) = 0$ and $\lim_{x\to\infty} f(x) = 1$. In simpler terms, any continuous function on a closed and bounded subset of $\mathbb{R}^n$ can be approximated to any desired degree of accuracy by a neural network [27, 34, 35].

In order to determine the weights of a neuron the ANN needs to be trained. The goal during training is typically the minimization of a loss function. There are plenty of loss functions to choose from. Those relevant for this thesis are presented in the following:

Mean squared error: Let vector $y = (y_1, \ldots, y_N)$ contain the target values, and $x = (x_1, \ldots, x_N)$ comprise the associated ANN outputs, then the MSE is[27]

$$l^{\text{MSE}}(x, y) := \frac{1}{N} \sum_{i=1}^{N} (x_i - y_i)^2.$$

Taking the root yields the root mean squared error (RMSE).

Cross-entropy loss: The cross-entropy between the empirical distribution $\hat{p}_{\text{data}}$ and the model p_{model} is [27]

$$-\mathbb{E}_{x\sim\hat{p}_{\text{data}}}[\log p_{\text{model}}(x)],$$

where $-\mathbb{E}_{x\sim\hat{p}_{\text{data}}}$ is the expectation under distribution $\hat{p}_{\text{data}}$. The minimum is attained if and only if both distributions are identical (discrete variables) or equal "almost everywhere" (continuous variables) [27]. Depending on the distributions, the negative log-likelihood can be resolved to different functions.

Take a classification task with C mutually exclusive classes, for instance. For a single data point, let $\tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_C)$ be a vector of normalized log-likelihoods and let y be the index of the true class. Then, the negative log-likelihood is given by $l^{\text{NLL}}(\tilde{x}, y) := -\tilde{x}_y$. Normalized log-likelihoods $\tilde{x}$ can be obtained by applying the softmax activation function to the ANN output x, followed by a logarithm. Alternatively, it is possible to incorporate both into the loss function, which yields

$$l^{\text{CE}}(x, y) := -\log(\text{softmax}(x)_y) = -\log\left(\frac{e^{x_y}}{\sum_{i=1}^{C} e^{x_i}}\right).$$

This loss is named "cross-entropy loss" in the machine learning framework PyTorch (see [36] for the official documentation) and used in this thesis to predict discrete variables. The losses of individual data points are combined by computing the average.

In case of $C = 2$, that is, a binary classification task, the negative log-likelihood of a single sample is given by

$$l^{\text{BCE}}(x, y) := \begin{cases} -\log(x) & , y = 1 \\ -\log(1-x) & , y = 0 \end{cases} = -[y \cdot \log(x) + (1-y) \cdot \log(1-x)],$$

where x is the predicted probability of $y = 1$, and $y \in \{0, 1\}$ is the actual observed class. Here again, losses of individual data points are averaged. PyTorch calls this loss function "binary cross-entropy" [36]. In this thesis, it is used for solving multi-label classification tasks.

With a loss function in place, which may be a weighted combination of multiple different loss functions, the ANN can be trained, i.e., the weights of the ANN can be updated in order to minimize the loss. Typically, a gradient-based optimization algorithm is used for this purpose. Gradient-based optimization is outlined in Section 2.5.3. In summary, the gradient of a function depends on the point it is evaluated at and points in the direction of the steepest ascent. Moving the point suitably far in the opposite direction decreases the function value. How far the algorithm is allowed to step is usually controlled with a learning rate hyperparameter. The smaller this hyperparameter, the smaller are the updates. For updating the weights of an ANN with a gradient-based algorithm, it is hence necessary to compute the derivative of the loss function with respect to each weight. This can be done with the so-called backpropagation algorithm, which involves two steps. Firstly, the forward propagation, in which the ANN output is computed and the objective evaluated. Then, in the second step, the derivatives are calculated layer by layer, moving backwards and using the chain rule of calculus [27]. Instead of determining the gradient for each individual data point, it is usually computed for a (random) set of data points at once. Such a set of data points is called a batch. For an elaborate explanation of the backpropagation algorithm, please see [27].

The ability to generalize is a desired feature in a learned model. Regularization provides strategies for training models with better generalization capabilities [27]. According to [27], regularization plays an important role as "[. . .] we might find—and indeed in practical deep learning scenarios, we almost always do find—that the best fitting model (in the sense of minimizing generalization error) is a large model that has been regularized appropriately" [27, p. 229]. Moreover, in our past experiments on ANN-based surrogate modeling (e.g., **[308]**), the ANNs often failed to generate meaningful output when no regularization was applied. The following regularization strategies are considered in this thesis:

Parameter regularization: The models created in the context of this thesis are trained with an L^1 parameter regularization. This means that the term $\alpha \|w\|_1$ is added to the loss function. The hyperparameter $\alpha \in [0, \infty]$ controls the impact on the total loss [27]. With this added regularization term, bigger (absolute) weights w lead to a higher loss. Hence, the optimization algorithm keeps the absolute weights smaller than without the term. In comparison to the L^2 regularization, the L^1 regularization generally leads to more parameters with a value of zero [27]. Please note that parameter regularization is usually not applied to the bias b and does require training experiments with different values of the hyperparameter α [27].

Early stopping: Typically, during the training of an ANN, the training set error and validation set error are evaluated periodically. While the training set error generally decreases over time, given the ANN is learning successfully, the validation set error can begin to rise again. Such a behavior is due to overfitting and can oftentimes be observed [27]. One strategy to battle overfitting is early stopping. With early stopping, a copy of the best known model measured by its validation set error is stored and the training is stopped if no better model is found within a given time frame [27].

Dropout: Dropout is an inexpensive approximation of bagging, i.e., training multiple models separately and combining their output [27]. In order to apply dropout, so-called dropout layers are integrated into the ANN, which determine whether the output values of neurons are passed on or not. For ANNs using affine transformations and non-linear activation functions, like those used in the context of this thesis, this is achieved by applying binary masks [27]. A simple example is illustrated in Figure 2.4. A binary mask is sampled for each data point, based on a hyperparameter specifying the fixed and independent probability for a neuron to pass on its output [27]. There can be multiple dropout layers in an ANN. Aside from introducing dropout layers, the ANN is trained like before. Given a trained ANN with dropout, there are different options for making predictions [27]: Firstly, computing the ANN output for different random binary masks and aggregating the result, for instance, by computing the average. Secondly, having each dropout layer pass on its expected output by replacing each value of the mask with its expected value. If, for instance, the probability of sampling a one, i.e., the probability of passing the value to the subsequent layer, is 75%, the expected value of the mask is 0.75.

Batch normalization: Batch normalization is applied to input or hidden layers by computing

$$\frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}]}},$$

where $x^{(k)}$ holds the neuron outputs associated with the k-th sample in the batch, $E[x^{(k)}]$ is the expected output, and $\sqrt{Var[x^{(k)}]}$ is the associated standard deviation [37]. During training $E[x^{(k)}]$ and $Var[x^{(k)}]$ are either running estimates or calculated per batch [27, 36]. Afterwards, both are fixed. Adding batch

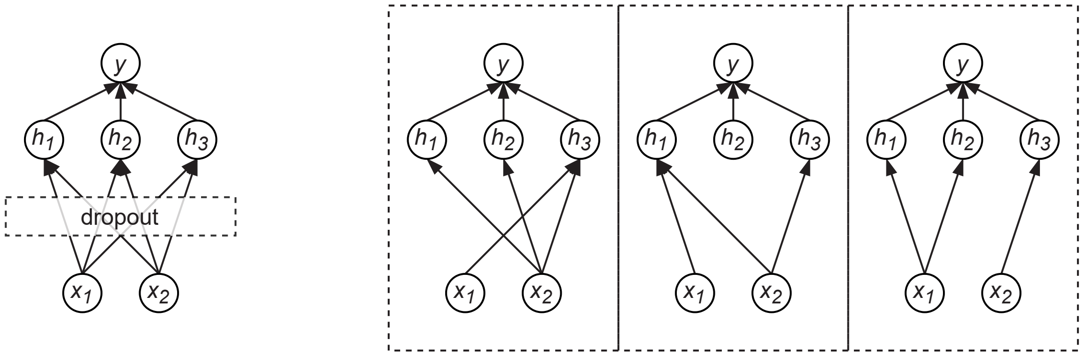

Figure 2.4: Illustration of a dropout layer. Neuron outputs are only passed on with a predefined probability. Which values are filtered changes randomly every new input. On the right-hand side, three exemplary results are depicted.

normalization prevents the gradient descent algorithm from unnecessary changes to the mean and standard deviation, since, with such a layer, it has no effect [27]. In PyTorch, the "batch norm" layer computes

$$\frac{x^{(k)} - E[x^{(k)}]}{\sqrt{Var[x^{(k)}] + \epsilon}} \cdot \gamma + \beta,$$

where ϵ is a small number added to avoid $\sqrt{0}$, as it has no derivative, and γ, as well as β are learnable parameters [36]. These two parameters are introduced to restore the expressive power of the ANN, which would otherwise be limited by the normalization [37, 27]. Even though it seems like multiplying γ and adding β cancels out the normalization, it changes the underlying learning dynamics: while prior to the normalization $E[x^{(k)}]$ is determined by the ANN layers, the mean of the batch norm output is β [27]. This makes learning the mean easier, since it eliminates the need to consider the interrelations of all previous layers [27]. Aside from that, batch normalization sometimes makes dropout unnecessary [27].

2.4.5 Support Vector Data Description

Although the focus lies on ANN-based surrogate modeling, the SVDD plays a prominent role in this thesis. SVDD is inspired by support vector classification and closely related to support vector machines (SVMs). It was introduced in [38], building on an idea presented in [39], and is explained in great detail in [40]. The following summary is based entirely on [40].

The idea of SVDD is to describe a set of data points $\{x^{(k)}\}$ by enclosing them in the smallest possible sphere with radius R around the center point a. Every point within the resulting sphere is considered to belong to the same class. It can be determined by solving

$$
\begin{aligned}
\min_{R,a} \quad & R^2 \\
\text{s.t.} \quad & \left\| x^{(k)} - a \right\|^2 \leq R^2, k = 1, 2, \ldots .
\end{aligned}
$$

This minimization problem can be adapted to consider outliers by adding a variable ξ_k for every data point $x^{(k)}$, resulting in

$$
\begin{aligned}
\min_{R,a} \quad & R^2 + C \sum_k \xi_k \\
\text{s.t.} \quad & \left\| x^{(k)} - a \right\|^2 \leq R^2 + \xi_k \quad , k = 1, 2, \ldots \\
& \xi_k \geq 0 \quad , k = 1, 2, \ldots .
\end{aligned}
$$

In order to penalize outliers, the ξ_k are added to the target function. Their impact can be controlled with the parameter C. With an increasing C, fewer data points are considered to be outliers, as the radius grows. Using the method of Lagrange multipliers (see Section 2.5.2), this problem can be reformulated to

$$
\begin{aligned}
\max_{\alpha} \quad & \sum_k \alpha_k (x^{(k)} \cdot x^{(k)}) - \sum_{k,l} \alpha_k \alpha_l (x^{(k)} \cdot x^{(l)}) \\
\text{s.t.} \quad & 0 \leq \alpha_k \leq C \quad , k = 1, 2, \ldots .
\end{aligned}
$$

The center point a is then given by

$$
a = \sum_k \alpha_k x^{(k)} .
$$

Depending on the position of the point $x^{(k)}$, α_k varies. A value of 0 means that $x^{(k)}$ lies within the sphere. If $0 < \alpha_k < C$, it lies on the surface, and for $\alpha_k = C$ it is outside. The radius is determined by computing $\left\| x^{(k)} - a \right\|$ for any point on the surface, i.e., any point $x^{(k)}$ with $0 < \alpha_k < C$. In order to test some point z, it is sufficient to compute its distance to a and check whether it is greater than R or not. Since only those points with $\alpha_k > 0$ have an impact on a and R, they are called support vectors.

By replacing the inner product $(x^{(k)} \cdot x^{(l)})$ with a kernel function $K(x^{(k)}, x^{(l)}) = (\Phi(x^{(k)}) \cdot \Phi(x^{(l)}))$, more flexible descriptions can be generated. The function Φ maps the data points into some other space, which is called feature space. Ideally it would map to a space where the data points are all within a sphere and all outliers are outside. There are different options to choose from, like the polynomial kernel, but in this thesis only the Gaussian kernel is relevant. The Gaussian kernel is given by

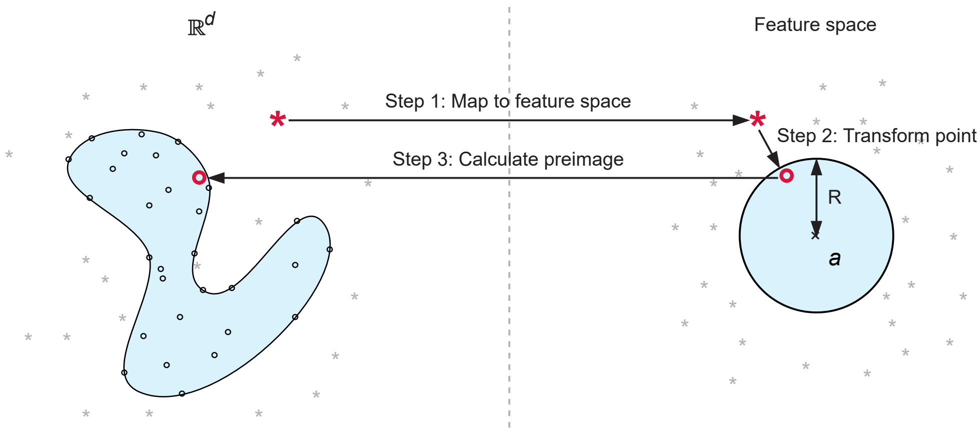

Figure 2.5: Illustration of the support vector decoder approach. Given an SVDD, data points can be mapped into the feature space, where the decision boundary is a sphere. If a point lies outside the sphere, it can be moved closer to the center point in order to change the classification result. Based on [11].

$$K(x^{(k)}, x^{(l)}) = e^{\frac{-\left\|x^{(k)}-x^{(l)}\right\|^2}{s^2}}.$$

The variable s is, like C, a hyperparameter. It determines how tight the decision boundary envelopes the support vectors. With an increasing s, the decision boundary expands and the description contains a growing portion of the vector space.

Support Vector Decoder Approach

Whether a data point lies within the decision boundary or not is determined by its distance from the center point a of the sphere in the feature space. Given the data points' feature space location, the center of the sphere, and the radius, it is possible to move the point towards and across the decision boundary. This concept is the core of the SVDD-based surrogate modeling of DERs and called "support vector decoder approach" in [11]. Throughout this thesis this approach is referenced multiple times. Figure 2.5 illustrates the general process. For the sake of brevity, the required calculations are omitted. Understanding the basic concept outlined here is sufficient for comprehending all SVDD related statements throughout this thesis. A detailed explanation, including the associated mathematical formulas, can be found in [11].

In the first step, an outlier is mapped to the feature space, where its distance from the center is larger than the radius of the sphere. Then, inside the feature space, the data point is moved in the direction of a. By setting the step size appropriately, the data point can be projected onto the surface or pulled inside the sphere. In the final step, the preimage is calculated. However, depending on the kernel, the preimage may not exist. In case of the Gaussian kernel, there are only approximate preimages, which can be calculated iteratively [11].

2.5 Optimization

This section provides a short overview of optimization related concepts mentioned in this thesis. It is meant as a concise recapitulation of the basic concepts in order to clarify the terminology rather than a proper introduction. For an actual introduction, please see the respective literature.

In mathematical programming an optimization problem has the general form [41]

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq b_i, i = 1, \dots, m \\ & x \in \mathcal{X} \subset \mathbb{R}^n. \end{aligned}$$

The function $f(x)$ is called the objective, and the constraints are determined by the real-valued functions $g_i(x), i = 1, \dots, m$ [41]. These constraints define the “feasible set” $\{x \in \mathcal{X} | g_i(x) \leq b_i, i = 1, \dots, m\}$ by further restricting the design vector x within its domain $\mathcal{X}$ [41]. If x is restricted to integer values, the optimization problem is called an integer programming problem [42]. In case some elements must be integers and others not, it is a mixed integer programming problem [42]. Depending on the objective function and constraints, it is possible to distinguish different classes of optimization problems. In the following, linear, quadratic, and nonlinear forms are outlined. There exist many further classes and characterizations, which are not relevant for this thesis, for instance, convex [42], conic, and semidefinite programming [41].

In linear programming, an optimization problem has the form [41, 42]

$$\begin{aligned} \min_x \quad & c^\intercal x \\ \text{s.t.} \quad & Ax \leq b \\ & x \geq 0, \end{aligned}$$

where vector $c \in \mathbb{R}^n$ holds the coefficients of the objective function $c^\intercal x$, and the $(m \times n)$-matrix A the coefficients of all constraints [41]. The vector $b \in \mathbb{R}^m$ is the so-called right-hand side of the constraints [41]. A quadratic program, in contrast, has a quadratic objective function $c^\intercal x + \frac{1}{2} x^\intercal Q x$, where Q is an $(n \times n)$-matrix, and linear constraints [43, 42]. Finally, the nonlinear program is the most general class and may be written as [44]

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \leq 0, i = 1, \dots, p \\ & h_j(x) = 0, j = 1, \dots, q. \end{aligned} \tag{2.1}$$

2.5.1 Computational Complexity

The difficulty of solving an optimization problem, that is, its computational complexity, varies from problem to problem. There are different factors influencing the computational complexity [45]:

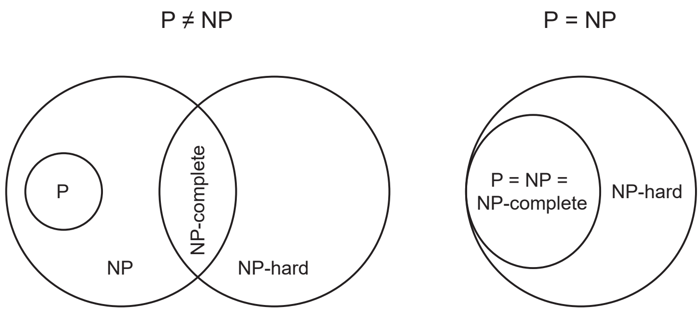

Figure 2.6: Computational complexity classes in case **P** ≠ **NP** and **P** = **NP**. Based on [45, 46].

Problem size: The number of variables and different values the variables can attain.

Running time: The hardness of a problem is linked to the running time of an algorithm solving it, that is, the number of elementary steps required until its termination. It is expressed as a function of the problem size and specifies an upper bound for the worst-case running time.

Problem reduction: It may be possible to transform the problem into another one.

It is generally distinguished between the problem classes **P**, **NP**, **NP-complete**, and **NP-hard** [45, 46]. For problems of class **P**, there exists an algorithm that, in the worst case, needs polynomial time [45]. In other words, the running time of the algorithm is a polynomial of the problem size [45]. Problems of the class **NP** can be solved and any given solution can be verified within polynomial time by some other algorithm [45]. This is also true for **P**, which makes **P** a subset of **NP** [45]. The class **NP-complete** contains problems of the class **NP** for which there is any other problem in **NP-complete** that can be reduced to this problem in polynomial time [45]. Lastly, a problem that is at least as hard as any problem in **NP-complete**, that is, a problem from **NP-complete** can be reduced to said problem, belongs to the class **NP-hard**. These relationships are illustrated in Figure 2.6.

2.5.2 Lagrange Multipliers

An optimization problem written in closed form may be solved with the help of the Karush-Kuhn-Tucker conditions [44]. Given a problem in the form 2.1, these conditions are [44]

$$
\begin{aligned}
\frac{\delta}{\delta_{x_k}} L(x, \lambda, \mu) &= 0\,, k = 1, \ldots, n \\
\lambda_i &\geq 0\,, i = 1, \ldots, p \\
\lambda_i g_i(x) &= 0\,, i = 1, \ldots, p \\
g_i(x) &\leq 0\,, i = 1, \ldots, p \\
h_j(x) &= 0\,, j = 1, \ldots, q.
\end{aligned}
\tag{2.2}
$$

The function $L(x, \lambda, \mu)$ is the so-called Lagrange function. It is given by [44, 47, 48]

$$L(x, \lambda, \mu) = f(x) + \sum_{i=1}^{p} \lambda_i g_i(x) + \sum_{j=1}^{q} \mu_j h_j(x)$$

and contains the Karush-Kuhn-Tucker multipliers λ and μ, which are often also called Lagrange multipliers [48]. By solving the equations 2.2, optimal solutions of the associated optimization problem can be identified [44]. For an explanation of the underlying assumptions and as to why these conditions define optimal points, please see [47] or [48], for instance.

2.5.3 Gradient-Based Optimization

The gradient $\nabla_x f(x)$ of a function f is a vector containing all partial derivatives of f with respect to x_k [27]. It points in the direction of the steepest ascent [44]. Gradient-based optimization leverages this information and may also make use of higher order derivatives.

The method of the steepest descent, makes steps in the opposite direction of the gradient, i.e., the direction of the steepest descent, using the update rule [27]

$$x' = x - \epsilon \nabla_x f(x).$$

How far subsequent points lie apart is controlled with the step size ϵ. In the context of machine learning, ϵ is the learning rate [27]. The step size can be a given parameter or determined dynamically, e.g., by analyzing $f(x - \epsilon \nabla_x f(x))$ or using some form of momentum [27].

In machine learning applications, usually the goal is to minimize the loss, which depends on the individual training samples. Often it is too expensive to compute the loss for each sample in the dataset and for each single gradient step. Stochastic gradient descent provides a solution, by treating the gradient as an expectation [27]. The gradient is not computed for the entire dataset, but instead for a batch of samples drawn uniformly from the training set [27].

2.5.4 Meta-Heuristics

Meta-heuristics are search algorithms which gather information in some sort of memory and use this information to decide where to search next in the feasible set [49]. New solutions are generated on the basis of the collected data and evaluated, yielding new information [49]. There are various meta-heuristics, which are often inspired by nature, including, evolutionary algorithms, simulated annealing, taboo search, ant colony optimization [49], and particle swarm optimization. Out of these, evolutionary algorithms are the only ones relevant for this thesis.

Evolutionary Algorithms

Evolutionary algorithms (EAs) belong to the field of evolutionary computing, which emerged from evolutionary programming, genetic algorithms, and evolution strategies [45]. Evolutionary computing, like the name suggests, is inspired by the natural evolution of species[45]. The following summary is entirely based on [45].

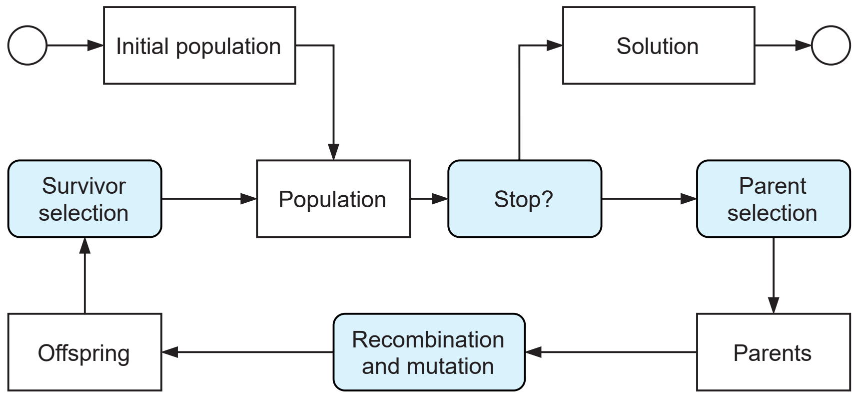

Figure 2.7: Outline of an evolutionary algorithm. Based on [45].

In the evolution analogy, the optimization problem constitutes the environment in which individuals (solution candidates) must demonstrate their fitness (value of the objective function). The idea is that the individuals compete for resources and by natural selection, the fitness of the population rises. Starting from some arbitrary initial population, new individuals are created by combining (cross-over) and altering (mutation) existing individuals randomly based on their fitness. In the next step, the new individuals compete with the existing ones for a place in the population. Figure 2.7 illustrates this process. It is repeated until a stopping criterion is met, e.g., the objective function value exceeding a specified bound or reaching a given computational limit.

Often, evolution in an EA takes place in a simplified and abstracted problem-solving space in which solution candidates are encoded. How they are represented varies between the different dialects of evolutionary computing. Nonetheless, no matter if it is strings over some finite alphabet (genetic algorithm), real-valued vectors (evolutionary strategies), or some other encoding, the encoded individual is called a genotype, whereas the solution candidate in the original problem context is a phenotype. The fitness of an individual is typically assessed in phenotype space rather than genotype space, as they can be very different from each other. Hence, in order to evaluate an individual, it is mapped from genotype to phenotype.

There are various options to design a genotype space, as well as the parent selection, recombination, mutation, and survivor selection steps. However, the basic understanding provided with this short summary is sufficient for the purposes of this thesis. A thorough introduction to evolutionary computing can be found in [45].

2.5.5 Black-Box Optimization

In a black-box optimization problem it is assumed that the objective and/or constraint functions are black-boxes [50]. The overall goal, like before, is to search for the x within the feasible set minimizing (or maximizing) $f(x)$. Black-box optimization is applied in different situations, for instance, when the optimization problem requires extensive simulations or laboratory experiments, or when the objective function is noisy [50]. The following summary is based entirely on [50].

Black-box optimization problems can be solved with either heuristic or non-heuristic algorithms. EAs, particle swarm optimization, and simulated annealing are only some available options. The non-heuristic approaches include direct search methods and model-based methods. An exemplary direct search method is the generalized pattern search, which makes use of a mesh of points and a set of directions. It repeatedly checks points of the mesh, as well as points around the current location determined with the set of directions. If a better point is found, the location is updated. The algorithm is proven to converge under some weak conditions. Model-based methods, in contrast, construct models of the unknown functions and derive information to guide the optimization. Please see [50], for an introduction to black-box optimization.

2.5.6 Online Optimization

In many situations there is a need to optimize some process without yet knowing all relevant information. Optimization problems with incomplete or no knowledge of the future belong to the field of online optimization. In an offline optimization problem, in contrast, the future is completely known in advance [46]. The performance of online algorithms is typically measured by means of competitive analysis [46]. An online algorithm ALG is called c-competitive if for any input sequence $\sigma \in \Sigma$ the resulting target function value satisfies

$$\mathrm{ALG}(\sigma) \leq c \cdot \mathrm{OPT}(\sigma) + a,$$

where OPT is an optimal offline algorithm and a a constant value [46]. The competitive ration c_τ is defined as [46]

$$c_\tau = \inf\{c \geq 1 \mid \text{ALG is c-competitive}\}.$$

2.5.7 Model Predictive Control

Model predictive control (MPC) provides a general framework for solving optimization problems [51]. It makes use of a feedback mechanism in order to handle uncertainty and disturbances [51], which may for instance be caused by incomplete knowledge. The basic steps of MPC are [51]: firstly, the utilization of a model to predict the future. Secondly, determining a sequence of decisions, that is, a control sequence, optimizing the objective function. Lastly, implementing only the first decision of the control sequence. Each of these three steps is repeated every single time step. There are no limitations to the implementation of the optimization step, that is, any type of optimization algorithm may be used [51].

2.5.8 Sequential Decision-Making

Sometimes optimization tasks involve sequential decision-making or can be modelled as a sequence of decisions. In a sequential decision-making problem, an intelligent agent is interacting with its environment with the goal to maximize its long-term reward [52]. The agent is the entity making decisions based on the observed state of the environment and implementing them through its actions. These actions influence how the state of the environment evolves. The state of the environment, in turn, determines the reward received by the agent. Figure 2.8 illustrates this process. Although only one agent is depicted, there may be other, external agents within the environment [52]. As an example, consider a robot moving blocks on a plane surface and sorting them by their size. In this example, the agent is the robot. It interacts with the environment by moving on the plane and manipulating the blocks. Hence, moving around, picking up blocks, and placing them down are actions performed by the robot. Not every action is possible at any time. For instance, there needs to be a block, in order to pick it up. Which action to

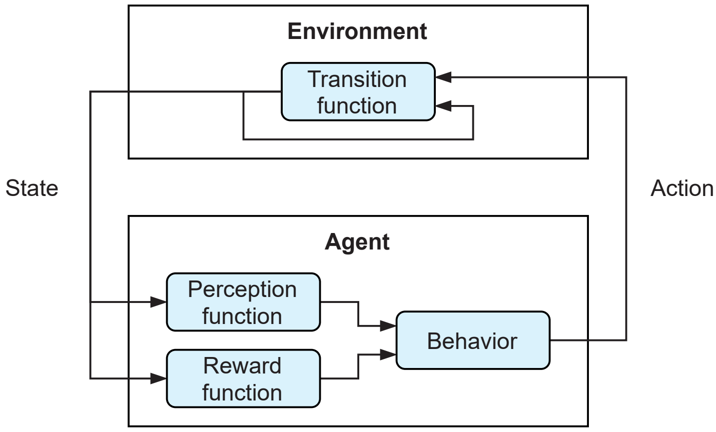

Figure 2.8: An agent interacting with its environment. Based on [52].

perform is decided by the robot, based on its perception of the environment provided by its sensors. The reward in this scenario would be determined by some measure reflecting how well the robot performs in its ordering task.

How an agent behaves is determined by its policy. There are multiple types of policies, ranging from fixed sequences of actions to stochastic action choices [52]. A policy can be generated in different ways, depending on the available information about the environment. It is possible to distinguish the planning scenario and the reinforcement learning (RL) scenario. In the planning scenario, the policy is derived by a planner from a known and complete model of the environment [52]. The planner is not necessarily the agent [52]. In RL, the agent itself is the planner, deriving the policy based on the experience gathered from interacting with the environment [52]. The involved learning process can be augmented, for instance, adding simulation models (simulated RL) or learned models of the environment (model-based RL) [52].

Dynamic Programming for Solving Markov Decision Processes

In dynamic programming, problems are solved by determining the solutions of subproblems and combining them into a solution for the original problem [53]. In contrast to divide-and-conquer algorithms, where all subproblems are independent, dynamic programming subproblems may share subsubproblems. Dynamic programming can be used for deriving optimal policies from a perfect environment model given in the form of a Markov decision process (MDP) [54]. The summary provided in this section is entirely based on [54].

An MDP is defined by the sets S, S^+, A, R, and the probabilities $p(s', r|s, a)$. The state of the environment is $s \in S$, followed by $s' \in S^+ \supseteq S$, after taking action $a \in A(s)$. In case of an episodic problem, $S^+ \supset S$ contains all states in S plus a terminal state. Which actions are valid is generally depending on the state s, hence $a \in A(s)$. The joint probability of reward $r \in R$ and the subsequent state s' is $p(s', r|s, a)$. Usually, a finite environment, i.e., finite sets S, S^+, A, and R, are assumed.

In an MDP, the subsequent state s' is only influenced by the current state s and the action a. The state s must hence contain all information relevant for the future, including details of past interactions. This requirement is called Markov property. There are methods to construct states having the Markov property from non-Markov observations.

The goal of dynamic programming is to maximize the expected total reward collected throughout each time step. This can be achieved by determining either $v_*(s)$ or $q_*(s, a)$ satisfying the Bellman optimality equations

$$\begin{aligned}
v_*(s) &= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s', r|s, a)[r + \gamma v_*(s')] \\
q_*(s, a) &= \mathbb{E}[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') | S_t = s, A_t = a] \\
&= \sum_{s',r} p(s', r|s, a)[r + \gamma \max_{a'} q_*(s', a')]
\end{aligned}$$

for all $s \in S$, $s' \in S^+$, and $a \in A(s)$. The parameter $\gamma \in [0, 1]$ is a fixed discount factor. The index t determines the time step. Both functions provide the maximum expected total reward achievable when starting from state s in time step t and, in case of $q_*(s, a)$, taking action a. Fortunately, approximations of $v_*(s)$ and $q_*(s, a)$ can be calculated and improved with the help of these particular equations. With either $v_*(s)$ or $q_*(s, a)$ at hand, it is possible to determine the action a yielding the maximum expected total reward.

A policy $\pi(a|s)$ maps the state s to probabilities of selecting each action. The expected total reward achieved with policy π, starting from state s in time step t, is given by the so-called state-value function for policy π:

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')].$$

The action-value function for policy π, i.e., $q_\pi(s, a)$, is defined analogously. Policies can be ranked according to the expected total reward. An optimal policy π_* yields the maximum expected total reward. There may be more than one such policy. The optimal state-value function is $v_*(s) = \max_\pi v_\pi(s)$. For further explanations and concrete algorithms, it is referred to [54].

Reinforcement Learning

RL also has the objective of determining an optimal policy. However, the policy is derived from experience rather than a complete model of the environment. There are various approaches to RL, each with its own benefits and challenges [55]. While some approaches explicitly learn a policy function $\pi(a|s)$, for instance in the form of an ANN, others do not. If no explicit policy function is learned, the actions are selected on the basis of other models, e.g., a learned approximation of $q_\pi(s, a)$ or a learned model of the environment's dynamics [55]. Some approaches even combine a learned policy with additional models (see [55, 54]).

Learning a model of the environment's dynamics and using it in order to derive or improve a policy is called model-based RL. A model of the dynamics provides either s' [55] or (s', r) [54], given the state s and action a. Only the subsequent state $s_{t+1} = s'$ is relevant in the following. In the deterministic case, the transition function $f(s_t, a_t) = s_{t+1}$ provides this state. In a stochastic environment the dynamics are given by $p(s_{t+1}|s_t, a_t)$. Therefore, trained with the data collected from the environment [55, 54], a learned model of the dynamics approximates either

$f(s_t, a_t)$ or $p(s_{t+1}|s_t, a_t)$. There are again various options for implementing model-based RL. The dynamics model may, for instance, be utilized in an MPC approach, or provide gradients via backpropagation to compute policy updates [55]. For an introduction into RL, including model-based RL, please see [55] and [54].

Monte Carlo Tree Search

Monte Carlo tree search (MCTS) is a planning algorithm which simulates how a sequence of decisions unfolds, makes estimates of the value function, and uses this information to direct the search towards better trajectories. Simulation in this context means the repeated selection of actions according to some policy until a terminal state is reached. With the help of these simulation results, a decision tree is built, step by step. Starting from the initial state at the root, each state equates to a node and each action to an edge. An iteration of MCTS involves the following steps [54]:

1. **Selection:** Starting from the root, the tree is traversed following a tree policy until a leaf is reached. Which edges, i.e., actions, are selected is influenced by the estimates of the value function.
2. **Expansion:** In this step, an arbitrary number of child nodes can be added to the selected leaf.
3. **Simulation:** Using a rollout policy and starting at either the selected leaf or the newly added child nodes, a simulation run is conducted.
4. **Backup:** The simulation result is used to update the estimates of the value functions in the generated tree.

These steps are repeated until a given time or computational budget is exhausted [54]. Then, one action leaving the root node is selected and the entire process is repeated once the environment reaches its subsequent state [54], solving the optimization problem in an MPC fashion. MCTS is a component of some artificially intelligent system, such as AlphaGo, which was able to win against the 18-time world champion in the game Go [54].

3 State of the Technology

In order to control DERs, their flexibility needs to be modeled and communicated. Plenty of models are presented in the literature, usually in the context of some specific use case. The proposed models are mostly byproducts, created in process of investigating some other topic. Such modeling approaches are listed and compared in Chapter 4, where an extensive literature review is presented. This chapter focuses on related publications that explicitly investigate the exploitation, communication, and modeling of flexibility.

3.1 Exploitation of Flexibility

Exploitation of flexibility means making use of DERs by controlling their operation. Control can be achieved in different ways. Mauser [5] distinguishes three different types of exploitation: the direct physical DR, indirect market DR, and other DR signals. As the name already indicates, in direct physical DR, DERs are controlled directly, for instance, by means of load reduction signals. The indirect market DR, in contrast, provides market related incentives, such as real-time prices, to adapt DER schedules. Exemplary other DR signals that do not fall under the previous two categories are warning notices and visualizations of the grid-state [5]. These three categories are the basis for the exploitation patterns proposed in **[316]** and which are explained in Section 4.2.

Another, similar classification is proposed in [56]. It is distinguished between autonomous control, indirect control, transactional (marked-based) control, and direct control. In this classification, direct control is equated to central decision-making and makes use of either one-way communication or two-way communication [56]. The other three types are attributed to local decision-making, where autonomous control means no communication at all, indirect control means one-way communication, and transactional control makes use of two-way communication [56]. This distinction in terms of decision-making and communication provides simple criteria for classifying exploitation schemes. However, the literature provides many coordination approaches for which this framework fails, especially for multi-agent systems, where agents act autonomously but often exchange information (see Section 4.3.5 for some examples).

3.2 Modeling Flexibility

While the literature makes use of a wide variety of models for DERs and their flexibility, the topic of modeling itself is only discussed occasionally, typically in the form of a review, a newly proposed generic model, or a combination of both (like in this thesis). For an extensive review of modeling approaches, please see Chapter 4. A selection of reviews presented in the literature is showcased in the following.

Applications of different white-box, grey-box, and black-box approaches for the modeling of buildings are discussed in [22]. The presented black-box approaches include ANNs and SVMs, but only for the purpose of generating forecasts. Furthermore, white-box, grey-box, and black-box modeling is discussed in general. In [57], a short review of building energy models in the context of DR is presented. The discussion of the modeling techniques mainly focuses on the distinction of white-box, grey-box, black-box models. The authors conclude that building characteristics need to be taken into account in DR applications and propose the investigation of clustering techniques [57].

A review of techniques for the modeling of heating, ventilation, and air conditioning systems is presented in [58]. It discusses several white-box, grey-box, and black-box approaches, and many associated data-driven models. One of the presented techniques is the "subspace state space identification (4SID)" [59], which makes use of a discrete time, linear state space model. It predicts the state and an output vector, given the previous state and an input vector. At a first glance, this is similar to the *state-based simulation* approach, which is presented and evaluated in this thesis. However, both approaches are very different from each other, since the state vector holds different kinds of information. In [59] the state vector reflects an internal process state and is only estimated from the observed system inputs and outputs, like the state of a hidden Markov model. Furthermore, the method for estimating this internal state vector depends on the fixed and specific model structure.

The flexibility of the heat demand in buildings and district heating systems is investigated in [60] by means of a literature review. It discusses a variety of definitions for the term "flexibility", presents several metrics and quantification methods, and names different sources of flexibility in such systems.

Agent-based modeling and simulation in smart grids is reviewed in [61]. Aside from a discussion of key concepts, like the definition of an agent, it presents a variety of publications that make use of multi agent systems. Almost all the presented literature considers DR, some kind of energy market, or a combination of both. Furthermore, some of these DR mechanisms and markets are briefly summarized and explained.

The literature provides more such reviews. From a modeling perspective, they mainly describe different approaches for modeling the flexibility of DERs. Sometimes these descriptions are more technical (e.g., [22] and [58]), discussing equations and concrete models, and sometimes more general (e.g., [57]), merely naming a few abstract categories.

3.3 Surrogate Modeling

Surrogate modeling for the purpose of describing the flexibility of DERs is not a novel idea. In [62], the authors propose the utilization of SVDDs in order to encode the space of feasible load schedules. With the help of an SVDD, feasible schedules could be identified by classifying randomly created schedules and filtering valid ones. This idea was later refined by considering the preimage of the SVDD sphere [63] instead of making random guesses. Then, in the next step, the concept was augmented by a mechanism able to project an infeasible schedule onto the set of feasible schedules [64]. A description of this mechanism can be found in Section 2.4.5. This mapping from infeasible schedules to feasible ones has been integrated into different optimization algorithms (see [11]) and coordination approaches (e.g., [65]).

Potential alternatives to the SVDD that have been proposed are classifiers [12, 66, 67, 68], and Chi-shapes [69], i.e., concave hulls around a set of points described in the form of a polygon.

In Section 5.2, surrogate modeling in the context of EAs is addressed. Using surrogates in an EA is not a novel idea either. An illustration of an EA and the steps in which surrogates may be used are presented in [70]. Applications for the initialization, recombination, mutation, local search, and fitness evaluation are pointed out. However, in the named examples, the surrogates mostly provide estimates of the fitness values of individuals. In this thesis, in contrast, the idea of substituting an entire step of the EA by surrogates is considered.

3.4 Learning About Flexibility

One motivation for the investigation of ANN-based surrogate models for the flexibility of DERs is the potential of learning them from recorded data. The utilization of energy time series to assess a flexibility potential is subject of [71] and [72]. In the presented procedure, so-called motifs, that is, similar sequences within a time series, are detected and used to compute flexibility measures. However, as the authors point out, the derived ratings are merely indicators for a potential flexibility, as the underlying processes that generated the energy time series are unknown. It is elaborated more on this limitation in Section 5.2.

Applications of RL in energy management may also be seen as a related concept. Depending on the RL algorithm, the models have to learn about the dynamics of the controlled DERs. An extensive review of the utilization of RL in DR applications is presented in [73]. Model-free RL in the form of Q-learning is the most popular RL approach in this context [73]. It involves learning the action-value function $q(s, a)$ (see Section 2.5.8). As the name suggests, model-free RL does not learn an environment model. Model-based RL, on the other hand, involves learning a model of the environment and therefore learning a model of the controlled DERs.

3.5 Flexibility of Chemical Processes

The concept of operational flexibility is by no means exclusive to the field of electrical engineering. In 1985, Swaney and Grossmann [74] proposed a flexibility index which measures the size of the parameter space over which chemical plants can operate in a feasible steady-state. This index is the root of the feasibility analysis for chemical processes [25]. However, even though the identical terminology is used, the notion of flexibility is slightly different. The operational flexibility of a process relates to a set of uncertain parameters. These parameters influence the process, but do not control it. In contrast, in the context of smart grids and with the definition of flexibility used in this thesis, flexibility relates to the consumed and provided power, that is, the control variables themselves. While it is possible to exploit flexibility in smart grids indirectly, for instance using price signals, the underlying flexibility is still understood in terms of energy and not money or some other quantity. Nevertheless, the basic concepts are still related and, thus, the field of chemical engineering may provide ideas which may be transferred to smart grid applications.

Surrogate models also find their application in the field of chemical process engineering. An overview of their different uses is presented by [25]. Several applications, including optimization, dynamic process modeling, and the already outlined feasibility analysis, are named. In case of dynamic process modeling, the surrogates are used to forecast the state of a process, based on prior states and the control variable inputs [25]. One example for this is presented in [75], where a kriging model provides forecasts of the state in the discrete time step $t+1$, given the state in time step t and the control input. Other types of surrogates named in [25] are polynomial regression models, and ANNs. This utilization of surrogates is closely related to the *state-based simulation* approach presented in this thesis.

4 Communicating the Flexibility of Distributed Energy Resources

Many different concepts of how the flexibility of DERs could be utilized are proposed in the literature. The corresponding architectures and exploitation mechanisms vary vastly from approach to approach. By addressing the first principal research question and its subquestions, this chapter aims to organize the approaches found in the literature in order to build a foundation for more precise evaluations and comparisons of the different concepts. The primary question **RQ1** "Which models can be used for communicating flexibility?" is answered by means of investigating the three more specific questions **RQ1.1** to **RQ1.3**, and then putting them into context. In regard to the individual questions, the chapter is structured as follows.

RQ1.1 What are the motivations for communicating flexibility?
This is the primary question of Section 4.1, and is answered based on current literature. The question is investigated in order to make sure no relevant aspects for the subsequent analyses are missed.

RQ1.2 How can flexibility be communicated and exploited?
Building upon the classification scheme we proposed in **[316]**, more elaborate patterns for the exploitation of flexibility have been developed. The patterns are presented and discussed in Section 4.2, incorporating concepts we previously published in **[313]**, **[312]** and **[308]**. The refinement has been conducted alongside the literature review for question **RQ1.3**.

RQ1.3 Which modeling approaches do exist and how can they be categorized?
A literature review has been conducted for answering this question. Its results are structured with the help of the newly defined patterns. The methodology of the review, the retrieved and classified literature, and generalizations are presented in Section 4.3. Then, using the findings of the literature review, Section 4.4 discusses general approaches for modeling flexibility and structures them in the context of communication. Thereby **RQ1** itself is answered. Two of our previous publications **[314, 313]** are included in the review. Additionally, concepts proposed in **[312]** and **[308]** are referenced.

After answering these research questions, Section 4.5 applies the developed systematics for classifying approaches to the different concepts for exploiting flexibility proposed in the German research project *C/sells*.

Before discussing the motivation for communicating flexibility and therefore the need to model it, it should be pointed out that the discovery of flexibility is not subject of this thesis. Discovering flexibility, i.e., detecting the potential for altering the schedules of DERs, is a fundamental and essential step prior to the exploitation itself. It is assumed that at some point flexibility has been discovered and shall now be made available to some external party. Nonetheless, Chapter 5 can provide starting points to automate the discovery of flexibility, as it introduces modeling approaches where the descriptions can be automatically generated and encode the flexibility learned from the provided data. The major prerequisite is that the available data itself must reflect the actually available flexibility.

4.1 Motivations for Communicating Flexibility

This section is based on [5] and [76], as both present elaborate discussions and reviews on the usage of flexibility in the energy system. Since providing flexibility to some external entity often involves communicating a quantification of it, there is an overlap of the respective motivations. Hence, it is first necessary to look into motivations for the general provision, before deducing incentives for the communication itself.

Fundamentally, flexibility is required to deal with fluctuations of the net load caused by uncertainty and variability, and to resolve contingencies in the electrical grid [76]. Therefore, the (qualitative) security of supply is one possible incentive. Regarding the demand side, looking into DSM and DR can provide more possible motivations. Goals of DR include the integration of RESs, faster load balancing by adjustment of consumption, reduction of cost due to lower capacity requirements, balancing of fluctuations in the production of electricity, avoiding costly units in the merit order, reduction of GHG emissions, and reduced market prices when there are DR market participants [76]. Multiple of these points are related to costs, i.e., monetary incentives. In each previously named motivation the exploitation of flexibility is assumed. This means that the potential to influence the operation of DERs is utilized by deriving suitable control signals. However, it is also possible that a flexibility provider receives a compensation solely for delivering a description of their DERs' abilities. This could be the case when data is required for a planning process, for example, in the computation of dynamic prices. All indirect measures summarized as "market demand response" (see also [5]) in Figure 2.1 could make use of the DER descriptions in order to derive more impactful price signals. When considering the individual descriptions, it is possible to assess whether a price change is likely to have an influence on the behavior or not. Depending on the model, it may even be possible to make precise estimations of the resulting load. The resulting tariff itself may already act as a form of compensation. Another example could be a service that reads and processes the descriptions and derives suggestions, for instance in regard to future investments in complementary DERs. The compensation in this case is non-monetary, in the form of advice, or the owner of the DERs may even pay for such a service.

Emissions of GHG may be summarized under the more general category of environmental incentives. Measures of DSM listed in [5] are warning notices, visualization, priority signals, as well as energy efficiency and conservation. Aside from taxes, standards, and building codes, efficiency and conservation measures include the provision of information and energy audits. Based on these, possible motivations could also include the derivation and provision of information, for example in form of a label, comparable to energy labels for household appliances.

Although these motives have been derived from the goals of DSM and DR, they are not limited to the demand side. System operators may want to exploit flexibility to reduce grid investments [76], which is again a monetary incentive. Suppliers generally aim to be profitable and hence have monetary motivations, too. The goal to participate in various energy markets can be derived from this motivation. Furthermore, these companies may also have environmental targets or aim to operate transparently. A method for visualizing operational flexibility in power systems is proposed in [77]. Such visualizations can help in assessing available operational options and thus help in planning processes. This may constitute a desire to be able to request data from decentralized DERs.

A closer look at energy sharing communities provides another perspective on the subject. Different incentives for shared ownership of DERs are discussed by [78]. Those applicable to the topic in question are mainly monetary compensations, acceptance, as well as ethical and environmental motivations. Acceptance is closely related to transparency, as transparency may be used to generate acceptance. Other motivations related to communities and collaboration may be the usage of "locally generated" energy (in a spatial sense), or simply altruism. Local balancing, which is basically the result from covering demand by consuming locally generated energy, is for example an important aspect considered in cellular energy systems. Such cellular energy systems are, for instance, investigated in the project *C/sells* and in the context of the project GOFLEX [79]. Altruism in the community mainly means supporting neighbors and other members of the community.

Finally, there could always be a law requiring announcing and offering flexibility. An example is the remotely controlled reduction of solar power feed-in in Germany as governed by § 9 of the Renewable Energy Sources Act [3]. For small installations affected by the law, there is no feedback, i.e., the flexibility is not communicated in this case, and hence only estimated. For large installations of more than 100 kWp, on the other hand, communication is bi-directional. In this case, by requesting the current feed-in power, the system operator knows the exact consequences for each possible control signal.

Potential barriers for DSM and DR are discussed in [76]. Two concerns closely related to the exploitation of flexibility are privacy and data security. The subsequent sections in this chapter introduce a set of patterns and present the results of a literature review. From these it is apparent that whether and to what extent these concerns may be an issue depends on how the exploitation is achieved. Nevertheless, these concerns may constitute the motivation to voluntarily offer highly abstracted descriptions instead of, for example, allowing direct access to a DER.

A summary of identified motivations for quantifying and communicating available options for influencing local DERs' operation to some external entity is given in the following:

Monetary incentives: Most of the identified incentives are related to money. They include receiving compensations for allowing DERs to be influenced, favorable tariffs, reduced market prices, reduced cost for balancing load and supply, and reduced infrastructure costs. Additionally, in communities, cost may be avoided by balancing load and supply within the community.

Security of supply: The contribution to the security of supply comes primarily from participating in a scheme for flexibility exploitation. Those cases generally require communication in order to participate. Nevertheless, security of supply may still increase even if only information is provided and DERs cannot be influenced. Given the additional information other agents in the power grid are able to incorporate these data in their planning.

Environmental incentives: Reduction of environmental damage, e.g., by averting the usage of peak load plants [76]. Also, in the context of local balancing of supply and demand, potentially reduced need for transmission infrastructure.

Altruism: Especially in communities, DER operation may be coordinated so that DERs work collaboratively even though it involves personal cost or discomfort.

Legal compulsion: The legal obligation to participate in any affair requiring the communication of DER information.

Visualization: Visualization of flexibility may be used for multiple purposes, like assessing options to react on fluctuations. Furthermore, it can help with the subsequently named motives.

Awareness: Given a suitable presentation, the data may be used to raise awareness on energy related topics like distributed generation, the impact of installing heat pumps, or scheduling decisions. While this, of course, may be done with any public data set, the topics are presumably more comprehensible when actual personal data is used and personal implications are shown. When it comes to scheduling, a visualization of alternative schedules can illustrate the consequences of certain decisions.

Planning of investments: General knowledge on available flexibility may be considered in grid planning, but also when it comes to investments in other DERs. The latter could be supported by online services which receive the description and derive suitable investment offers.

Transparency: Quantifying and publishing the range of operational alternatives could be employed as a measure for increasing transparency. This data can then be used to reenact and analyze past situations and would be especially interesting for scientific applications.

A mail-back survey conducted in New Zealand investigated the motivation of residents to participate in peak demand response. The results have been published in the year 2011 in [80]. Participants could value the importance of price, environmental aspects and security of supply on scale from 1 to 5. Out of these options, price was valued most important, closely followed by security of supply. The valuation of environmental benefits was significantly lower than the other two. These findings indicate that monetary incentives and security of supply may also be the main motivation for communicating flexibility. However, it is important to note that in the context of the recent global climate protests, repeating the study today could result in different findings.

4.2 General Patterns for the Exploitation of Flexibility

The literature makes use of various approaches for the coordination of DERs, which require the exchange of all kinds of information. Often multiple approaches for the exploitation of flexibility are used in a single publication. For the purpose of structuring the findings of the literature review presented in Section 4.3 and enable further analysis, a detailed classification is necessary. Focusing on demand response, in **[316]** we presented four general patterns for communicating and exploiting flexibility. The four patterns can be described as follows:

Physical demand response: The DSMgr receives states and further information from DERs, and uses these to send control signals, such as on/off commands and set points, directly to each DER.

Direct market demand response of abstracted flexibility: The flexibility of DERs, either individually or combined, is explicitly described in a model. This model is sent to the DSMgr, who chooses how to use the offered flexibility, employing model specific algorithms. The resulting choice is sent back to the EMS managing the DERs.

Indirect market demand response of implicit flexibility: In this pattern flexibility is exploited by providing incentives, e.g., dynamic electricity prices. The pattern can be implemented without explicitly modeling the flexibility. Following the incentive is generally not mandatory, but deviation generally results in higher cost.

Decentralized market demand response: Coordination is achieved in a distributed manner, for example by auctions or distributed heuristics.

In order to allow a more detailed analysis with regard to the modeling, communication, and coordination, these four patterns have been refined for this thesis. Figure 4.1 depicts the five patterns that have been derived. The sources of flexibility are the DERs owned by the flexibility provider. Flexibility is provided to one or multiple external entities, for example, but not limited to, an aggregator, a system operator, or some other kind of DSMgr. Since the individual DERs are usually not equipped to communicate with any external parties themselves, in general, some kind of gateway is required. In simple scenarios, this gateway forwards messages between the DERs and the external party. More advanced coordination schemes require intelligent systems that handle the communication, evaluate the results, and manage the DERs accordingly. EMSs are able to provide both functionalities. For this reason,

Figure 4.1 depicts EMSs as communication interfaces on the side of the flexibility providers. The external parties also provide an interface for exchanging the pattern specific data. In the case of stigmergy (see Section 2.2.2 for a short introduction), the shared medium is considered to be the external entity and the device used for observing its state embodies the interface. There are two prominent changes in comparison to **[316]**. Firstly, the *direct market demand response of abstracted flexibility* pattern has been split into two separate patterns. Secondly, the *decentralized market demand response* pattern is replaced with a more generalized pattern for the exploitation based on state information. Even though this new pattern matches the general implementation of distributed coordination, this distinction avoids unifying all distributed and decentralized control structures in one single communication pattern. Take a model that describes feasible schedules, for example. Such a model may be used in centralized, as well as distributed coordination approaches. To facilitate this distinction, whether an approach is centralized or not is seen as an additional attribute, but not as a criterion for the classification itself. The patterns and therefore the classification primarily distinguishes the type of information that is communicated. In Figure 4.1 only basic information that is exchanged is depicted. Communication is of course not limited to exactly this data and the two depicted steps of data exchange. Generally, the communication from the flexibility provider to the external entity may be preceded by other messages, for instance, the external entity announcing that they are in need for offers from the flexibility providers. Additionally, after the depicted information exchange, the flexibility provider may respond and provide feedback. Such feedback messages can be used for implementing iterative approaches, e.g., for updating electricity prices as found in [81] and [82].

Direct exploitation: The first pattern shown in Figure 4.1 shows the direct exploitation of flexibility. In this pattern the external entity has detailed information on the current states and is provided with measured load profiles and forecasts. Control signals directly target the individual DERs. The signals are usually derived using explicit models of the controlled DERs. Hence, the local usage of DERs is fully transparent for the external party during the provision of flexibility. Depending on the type of DER it may be possible to infer private information concerning the flexibility provider. A closer look on the types of models is provided in Section 4.3.1.

Exploitation of abstracted flexibility: In the exploitation of abstracted flexibility more abstract models are used to describe the DERs. This raises the question of how this pattern is different from the direct exploitation pattern. Generally speaking, models are simplified representations of systems, i.e., involve some level of abstraction. Hence, the direct exploitation pattern also uses abstract descriptions of flexibility. The difference is that for the direct exploitation the DERs are represented by individual DER specific models. Models for the exploitation of abstracted flexibility are more general and can usually be applied for different kinds of DERs. Nevertheless, this criterion alone does not suffice for a clear distinction between the two patterns. Take the ability to start and stop a DER, as an example for a very general representation of any device. While many different DERs may be described in such a way, especially generators and loads, this exact model may still be used exclusively for one specific device and thus in a DER specific way. Hence, the context in which the model is utilized plays a crucial role. There is undoubtedly a grey area between the first two patterns. However, there are approaches using highly abstracted representations, such as an SVDD (e.g., [62]). For these, the whole process from generating the model to deriving and communicating suitable flexibility choices is different from the direct exploitation. Hence, the introduction of a pattern for the exploitation of abstracted flexibility is required to allow a more differentiated analysis. In order to clearly distinguish and classify approaches from the literature two simple criteria have been developed and used. Since abstraction is a process of generalization, it involves eliminating details [19]. Hence, a more abstract model is suitable to represent a wider range of DERs classes. Therefore, an approach is deemed to exploit abstracted flexibility, if it makes use of a specific model for more than only one type of DER. The basic types of DERs are

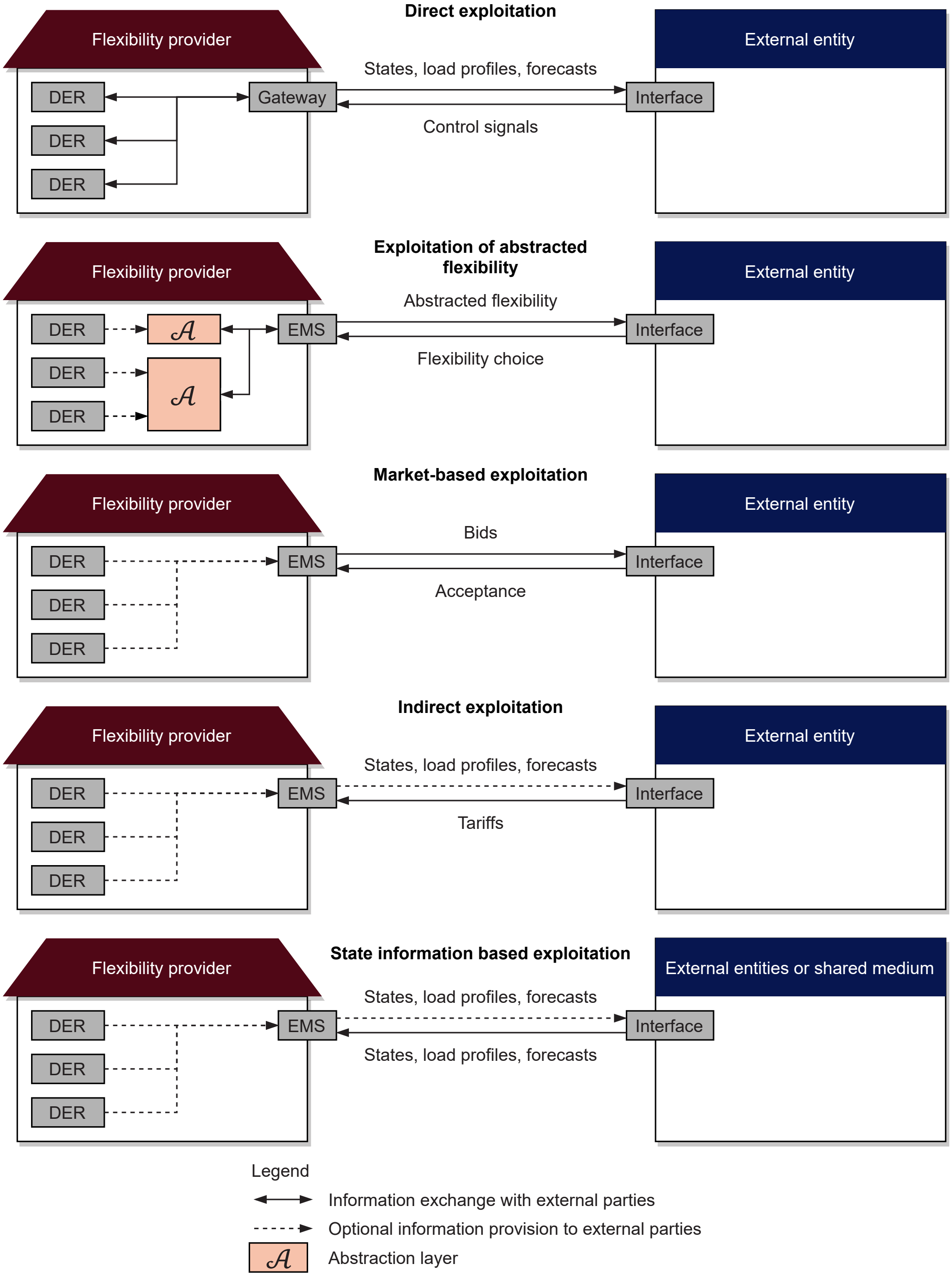

Figure 4.1: Patterns for the exploitation of flexibility.

considered to be generators, storage systems, and loads. If in a publication a model is used for only a single type, even though it could also describe others, it is not considered to be exploitation of abstracted flexibility. In other words, the utilization of a model able to represent multiple types of DERs is a necessary property, but not sufficient on its own. Whether the model is used in this way or not, is considered the determining factor in this thesis. The second criterion is the aggregation of multiple DERs. If in a publication a model is used for communicating the aggregated flexibility of DERs, it is regarded as exploitation of abstracted flexibility. It is important to note that patterns are often combined. For instance, there are approaches in the literature that optimize or place market offers based on aggregated models, but nevertheless use the direct exploitation pattern for controlling the DERs (e.g., [15] and [83]). Hence, if existent, the different levels in the coordination mechanism have to be distinguished.

Market-based exploitation: Closely related to the exploitation of abstracted flexibility is the pattern for market-based exploitation. As markets generally utilize standardized products, which can be interpreted as a shared model for all types and aggregates of DERs, the major difference from the exploitation of abstracted flexibility lies in the existence of a market. Models used in such markets are often highly abstracted and may not suit every DER. After collecting offers, the market is cleared using a clearing algorithm. The result, in its simplest form a note that the offer has been accepted, is then communicated to the flexibility provider who submitted the offer. The clearing process may also generate more complex output, such as a target power selected from an offered interval (e.g., [84]). Reserve markets are a special case in the market-based exploitation pattern. Depending on the type of reserve, the actual provision of flexibility is achieved with the help of direct signals or observed information like the power line frequency. This secondary exploitation pattern has been omitted in the classification in Section 4.3, as it is tied to the respective market and product. It is important to note that since it is possible to mix multiple exploitation mechanism into a singe approach, classification can sometimes be difficult and influenced by personal views, especially in regard to whether a market exists or not. In this thesis, requests and offers are only considered market bids, if prices are transmitted alongside. A fixed price is simply seen as a compensation and does, on its own, not constitute market-based exploitation. The same applies for requests and offers without prices.

Indirect exploitation: The indirect exploitation pattern covers all cases where some kind of incentive is offered to a flexibility provider in order to influence their behavior. It is up to each flexibility provider if they react or not. Figure 4.1 depicts tariffs as a response to the optionally provided states, load profiles, and forecasts. These optional data may be used to make targeted adaptations of the individual tariffs. It is important to note that this pattern is by no means restricted to electricity tariffs. Aside from the manifold existing pricing schemes it also includes any other approach that is based on monetary incentives. Please note that demand bidding programs, in contrast to [5], are interpreted as market-based exploitation, since they involve submitting a bid and acting when requested to. When there is no information sent to the external party, the communication can be considered one-way, in contrast to the two-way communication generally used (compare [85]) in the direct, abstracted flexibility and market-based patterns.

State information based exploitation: Finally, the state information based exploitation is closely associated with decentralized and distributed coordination, as the flexibility providers do not receive explicit requests for changing their behavior and, consequently, the DERs' schedules. In the case of decentralized coordination (as defined in Section 2.2), no data is exchanged between flexibility providers [86]. Hence, the only source for external information is a shared medium which reflects the global state. An example for such a medium is the electrical grid which is observable by metering devices. In distributed coordination architectures, data is shared with one or multiple external entities which may be flexibility providers themselves. However, distributed approaches are not restricted to this exact pattern. The pattern involves the exchange of the local information with external entities, the observation and manipulation of a shared medium, or both. Local information refers to the current states, measured load profiles and forecasts. After retrieving this information from the respective source, it is utilized by the flexibility provider to update the DER schedules. When using a liberal interpretation of the term “state”, any data connected to the DERs’ operation can be exchanged in this pattern.

Table 4.1 helps to gain a better understanding of the individual patterns. The type of the data received from the external entity plays a crucial role in the distinction. In the first three patterns the external entity more or less directly instructs the DERs (direct exploitation) or the EMS (exploitation of abstracted flexibility and market-based exploitation) on what to do. For the remaining two patterns, a reaction is generally voluntary, but not adhering may cause disadvantages. Incentives are offered for the indirect exploitation. For the exploitation based on state information, it is possible to distinguish even further. If the received data is associated with individuals, consider load schedules of flexibility providers, for instance, agents can observe their surrounding and coordinate their actions directly. This does not necessarily mean a peer-to-peer communication. Such information may easily be shared via some form of shared memory, like a virtual blackboard. In this thesis, this is called “exchange of state information”. If, on the other hand, the received information is an aggregated signal, i.e., it is not possible to know individual contributions to this signal, the approach is considered to be stigmergic. There is sometimes only a fine line between approaches based on stigmergy and others based on exchanging state information. This point is illustrated in the argumentation outlining why the presented mechanism in [18] is stigmergic [18, pp. 68-70].

Table 4.1: Distinction of patterns based on communication endpoint.

<table>
<tr><th colspan="4">External entity. . .</th></tr>
<tr><th>. . . demands action</th><th>. . . sends incentives</th><th colspan="2">. . . provides state information</th></tr>
<tr><td rowspan="3">Direct,
Abstracted flexibility,
Market-based</td><td rowspan="3">Indirect</td><td colspan="2">State information is associated with . . .</td></tr>
<tr><td>. . . individuals</td><td>Exchange of state information</td></tr>
<tr><td>. . . aggregate</td><td>Stigmergy</td></tr>
</table>

4.2.1 Communicating Models for the Flexibility of Distributed Energy Resources

The actual exchange of flexibility models is only indicated in Figure 4.1. While the exploitation based on exchanging and observing state information, as well as the indirect exploitation do not necessarily require exchanging models for the DERs’ flexibility, the remaining patterns do need them. In the market-based exploitation the offers are generally standardized. Therefore, every market participant knows the common data structure. The direct exploitation, on the other hand, requires the external party to have sufficient knowledge of the DERs to interpret the received data and derive control signals, e.g., by using a standardized interface. The interface provided by the control boxes offered for the smart metering architecture in Germany is such an example. In-between lies the exploitation of abstracted flexibility, where models are versatile and standardized to some extent, but not DER specific. In some cases the models for the exploitation of abstracted flexibility are individually generated, for example, in an automated process, which allows an even more detailed differentiation of approaches. Figure 4.2 shows the general process of communicating models that are created by the individual flexibility providers. The model with its parameters, after being generated, is sent to the external entity. The external entity stores this information for later use. In order to

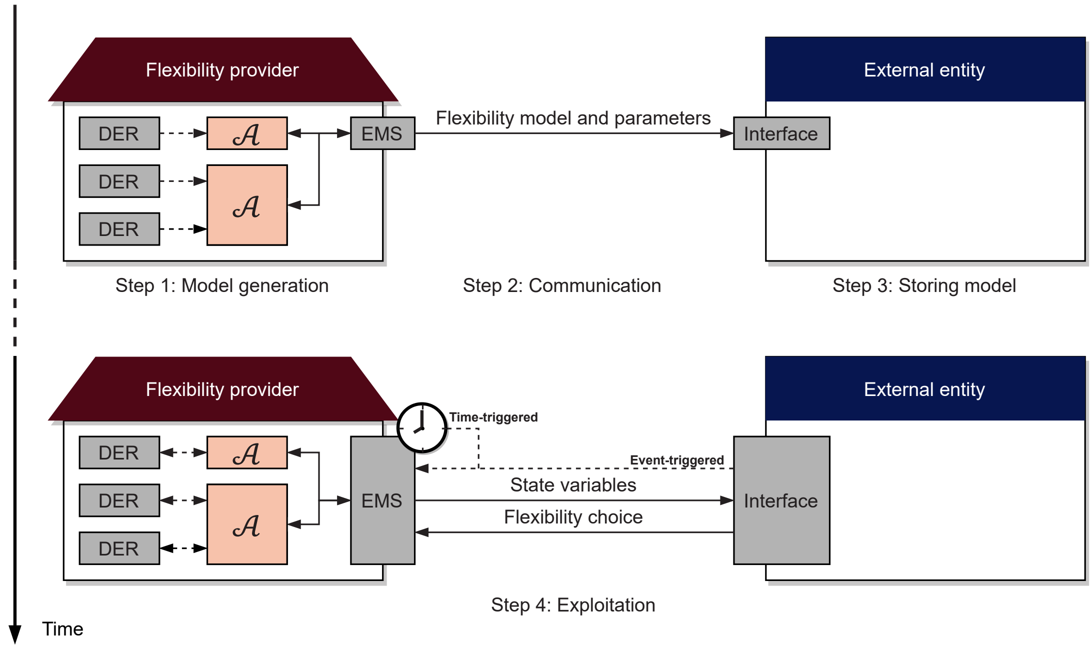

Figure 4.2: Communicating individually generated models, based on **[313]**.

determine the currently available flexibility, the current state is required in addition. Once triggered by a schedule or another defined event, the current state is sent to the external entity, who is then able to derive a flexibility choice if required. Whether generation and actual exploitation can be separated, depends on how long the generated model remains a valid representation. Approaches like the cascade classification [12, 67] and SVDD [62, 63, 87] encode the current state within the model and, thus, need to be generated whenever the state has changed. For such models, all the steps depicted in Figure 4.2 have to be repeated every time. Other approaches that allow to pass the current state as an input argument, e.g., when using ANNs **[312, 308]**, can omit the repeated model generation and communication steps as long as the system parameters don't change.

4.3 Literature Review

The refinement of the exploitation patterns presented in the previous section has been conducted in conjunction with a literature review. This section presents the results of the review using the revised patterns as basis for structuring the findings. In order to select a literature database, different popular databases have been tested with search queries, resulting in the choice of Scopus. Scopus is advertised as being "[...] the largest abstract and citation database of peer-reviewed literature: scientific journals, books and conference proceedings". Results include matches from IEEE Xplore, Springer, and, of course, Elsevier. The search query the review is based on is

Table 4.2: Terms used in the search query.

Word A	Word B	Word C	Word D
communicat*	model*	flex*	“smart grid”
coordinat*	encod*	“search space”	“micro grid”
exchang*	quantif*	feasib*	“demand side”
exploit*	characteri*	operab*	“demand response”
framework			“distributed energy resource”
taxonomy			“load management”
categor*			“load balancing”
			“load shifting”

$$\Big(\texttt{\{Word A\}}\ \textbf{W/16}\ \texttt{\{Word B\}}\ \textbf{W/16}\ \texttt{\{Word C\}}\Big)\ \textbf{AND}\ \texttt{\{Word D\}},$$

with `Word A` to `Word D` being either of the terms found in the respective column of Table 4.2. In other words, all terms in a single column are combined using the OR conjunction. The proximity operator **W/16** results in a match when two terms are within a distance of 16 words. The number 16 was an arbitrary choice with the intention to be able to find matches in rather long sentences. Please note that **W/16** is used twice in conjunction, meaning two of the three words may be separated up to 32 words. While `Word D` is supposed to restrict the findings to publications of the relevant scientific field, `Word A` to `Word C` are intended to filter for the topic. The terms within each single column are related in their meaning or implication, and have been chosen by testing different queries. `Word A` expresses the need to structure and/or communicate. The different alternatives for `Word A` are not strictly restricted to communication, since communication itself is rarely discussed. `Word B` requires that there is a model, and `Word C` stands for the various terms for the concept of flexible DERs. The asterisk (*) is a wildcard character, which stands for any sequence of letters. It was used heavily to catch the various phrasings and terminology found in the literature. Possible matches for “flex*”, for example, are “flexible”, “flexibility”, “flexibilities”, and “flex-offer”.

The search with Scopus using the presented query has been conducted on December 12, 2019, resulting in 791 matches. Out of these, 100 could not be taken into consideration, as they were either not accessible or only accessible in a language other than English or German. The papers falling in either category were almost exclusively in Chinese language. Some results are entire conference proceedings. For these, the table of contents and abstracts were screened, and relevant papers included in the analysis. The accessible 691 results were then filtered in a three-step process. Firstly, if it was apparent from the title that the publication has no relevance for this thesis, it was dismissed. Secondly, the same was done based on the abstract. Lastly, the content was reviewed as is outlined in the next paragraph. Even though every possible `Word D` is a term relevant for this thesis, many search results stem from other fields. Results related to the field of computing are particularly prominent in the filtered publications, presumably caused by including the terms “load management”, “load balancing”, and “load shifting” as options for `Word D`.

The search query has been formulated to return results from different research fields dealing with flexibility in electrical grids. Focus is the exploitation of flexibility provided by DERs, rather than the flexibility of the electrical grid itself as is for example discussed in [88]. The presented publications are, however, not limited to this particular use case. For this reason, there are many papers that are somewhat relevant, but do not provide additional insights when presented in this review. To keep the results manageable and exclude most irrelevant findings, the content based filtering has been conducted using a variety of criteria outlined in the following: For a paper to be considered relevant, it is sufficient that it either meets at least one criterion for relevancy, or does not meet any criterion for irrelevancy. Firstly, if the reviewed work makes use of highly abstracted models for flexibility, like discussed for the exploitation of abstracted flexibility in the previous section, it is immediately considered relevant. Please note that "highly abstracted" refers to utilizing a single model to represent either different types of DERs or aggregates of multiple DERs. Secondly, papers that focus solely on the processes within a given system, for example a building or micro grid, have been deemed irrelevant if they neither mention the purpose of demand side management nor the need for some sort of communication. Simply using terms like "control signal", "price signal" or "request", or depicting data exchanges between different agents is already sufficient for a paper to be considered relevant. Thirdly, not considered are grid operation centered works, if they only use very simplified models for DERs. Fourthly, publications focusing on the planning of investments or targeted installation of hardware have been filtered, as they analyze long term or predominantly local effects. Lastly, papers are only considered if they at least conclusively outline how flexibility is modeled. Several works only present highly conceptual architectures and are therefore not included in this review. As all papers were screened to test for these criteria, no paper from the results that is highly relevant for the topic of this thesis could have been missed. Based on these rules, in total 173 results have been identified as being relevant.

In the next step, all relevant papers have been classified according to the patterns presented in the previous section. Often a mixture of different approaches is combined, e.g., an aggregator directly exploiting the flexibility of the DERs under their control to trade on the energy market. In these cases multiple patterns are listed for a single entry in the tables showing the results. Although examples of other patterns are mixed into the tables, papers are listed in the table associated with the predominantly utilized pattern. Alternative options would have been separating the entries or adding them multiple times in the different tables. The former comes at the cost of comprehensibility, as for each paper all lists would be needed to gain a full understanding of the respective approach. The latter option would inflate the presentation of the results, with almost no added benefit. Please note that the analysis is considering all examples for a given pattern, no matter in which table it is listed. In publications, markets or tariffs, and the resulting signals are sometimes simply assumed to exist. In such cases, they are often only mentioned and not further specified. Consequently, the pattern associated was not listed here, as no insight could be gained. For the direct exploitation pattern, entries without added benefit have also been omitted. Generally speaking, a form of direct exploitation can always be found at some level of the grid hierarchy, as usually DERs do not control themselves. To see this, consider a smart building with an automated building EMS optimizing the energy cost. While the predominant pattern is either the indirect or the market-based exploitation (assuming a suitable market exists and is accessible), the local control within the building can be interpreted as a direct exploitation from the DERs' point of view. From this perspective, the DERs are the flexibility providers and the EMS is the external party. However, since typically EMS and DERs are in possession of the same entity, this is a liberal interpretation of the direct exploitation pattern. For this exact reason, these cases are not listed in the results, except if they involve some noteworthy modeling approaches. The aforementioned direct exploitation by an aggregator and comparable instances, on the contrary, are included in the lists, since for these the external party is truly external. From these examples it is apparent that not only may there be different exploitation patterns used at the same time, they may also vary throughout the hierarchical levels of the electrical grid.

The tables presenting the results are all structured identically. In the first column, the pattern is given. Keep in mind that multiple patterns may be associated with a single publication. In these cases there are multiple consecutive rows, providing more details for each pattern. Patterns are indicated with a single letter. Those are derived from (D)irect, (A)bstracted, (M)arket, (I)ndirect, and, to further characterize the state information based exploitation, either (E)xchange or (S)tigmergy. The next four columns provide more information on the DERs that have been controlled or influenced in the paper. Having these data helps in analyzing the modeling approaches more closely. It is distinguished between generation units, storages, loads, and a generic DER type, which unifies all the others. Ticks indicate whether the respective types of DERs are considered. Although many different DERs can be found in the relevant search results, some more specific classes than just the general type appear repeatedly. In the review process the additional distinction of aggregates (Agg), renewable energy sources (RES), EVs, and thermostatically controlled load (TCLs) emerged. If, in a paper, the operation of one of these is influenced, the respective column explicitly states Agg, RES, EV, or TCL. In these cases an additional tick means that there are further DERs considered. The reasons for adding these three additional distinctions lies in their characteristics. For RES and EVs it is the restricted availability and for TCLs their storage like properties. Furthermore, the EVs may be either storages or loads, depending on whether they may be discharged to the grid or not. If a DER simply exists but is not controlled with the specified patterns, it is not listed in the tables. Common examples are RES like solar power, when there is no curtailment. The order of the results does not reflect any kind of rating, it rather mimics counting with binary numbers based on the DER type columns. Exceptions are made when beneficial for the understanding of consecutive entries. Next is a column that provides more specific information on how the pattern is implemented. If there are any noteworthy approaches in the sense that they are outstanding compared to the most basic ones found in the literature, they are listed in the table. Whenever an aspect was not clearly outlined in a paper, it was resorted to classify according to the provided mathematical formulations.

In regard to the abstraction of the DERs, there are several concepts for modeling that repeatedly appear, but sometimes use different terminology. Examples are “compressible” [89], “reducible”, and “curtailable” load meaning the same, and different types of shiftable loads sometimes being distinguished and sometimes not. In an effort to provide a more intelligible review, the terminology has been unified. Therefore, reducible loads are simply listed as *curtailable* loads and shiftable loads are classified as either *shiftable* or *time shiftable*. Some papers consider shiftable loads as proportionately shiftable, i.e., some controllable part of the load may be shifted in time. Others use a more restrictive definition, only allowing the load to be shifted as a whole. Since such a time shift may be expressed by simply specifying the amount of time the load is shifted, this property is called time shiftable in this thesis. The distinction emphasizes that these two types of shiftable loads are modeled differently throughout the literature. While a shiftable load reacts to a continuous signal determining changes in the schedule, a time shiftable load generally receives a discrete signal defining the time period a DER runs. With continuous time, the signal would be continuous as well, but the literature usually makes use of time discrete models.

4.3.1 Direct Exploitation

The direct exploitation is the by far most commonly used pattern in the relevant literature returned for the search query. Even though the majority of publications at least acknowledges that there is a need to communicate, they mostly do not state any technical details. All those that do elaborate on technical details are listed in Table 4.3. These papers all discuss how the standards IEC 61850 or IEC 61970 may be extended in order to include the types of DERs specified in the table. The most generic concepts are given by [90] and [91]. In [90] the combination of CIM, which is defined in IEC 61970, and ISO 15926, a standard for power plant operation, is proposed to enable multi-domain studies. Extending IEC 61850 with the ability to provide detailed specifications of functions and related data, as described in IEC 61499, is discussed in [91]. A more specific concept is outlined in [92], where an adapter integrating CIM with OpenADR is presented. All entries in Table 4.3 are considered direct exploitation, since the external entity receives detailed data from the DERs and is able to send control signals. Since OpenADR is able to provide price signals, it is at the same time an example for the indirect exploitation of flexibility.

Table 4.3: Direct exploitation of flexibility in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
D			EV		IEC 61850		[93]
D			✓		IEC 61850		[94]
D			✓		IEC 61970		[95]
D		✓			IEC 61970	ISO 15926	[90]
D	✓				IEC 61850	IEC 61499	[91]
D&I	✓				IEC 62746	OpenADR-CIM adapter	[92]

Table 4.4 and Table 4.5 list papers utilizing direct exploitation that are not associated with any existing standard. Parentheses are used in the "Noteworthy Abstractions" column in order to indicate tuples and to eliminate ambiguity, if multiple options are listed. Take the entry "Requests (energy, duration, and time preferences)", for example. The word "requests" indicates that control signals are derived and sent by the external entity in the form of requests and "(energy, duration, and time preferences)" shows the structure of the requests. Additionally, it is distinguished between discrete and continuous signals, referring to the set of possible values sent to the EMS and not time. Discrete values usually include, but are not limited to, the options of turning a device on and off. Continuous signals are typically chosen from an interval of admissible values.

Overall, the most considered DERs in the direct exploitation pattern presented here are flexible loads. This picture is, however, to some extent skewed by the design of the filtering process. Also, as can be seen in the tables, the most variation concerning the employed models is found in loads. Therefore, papers only including generation units and storages, e.g., in a unit commitment problem, have a lower chance of being considered relevant for this thesis as the models are all very similar and often simple.

The results show that for the direct exploitation, DERs are often clustered according to common characteristics. Flexible loads are frequently modeled as being shiftable, time shiftable, and curtailable. Also, load shifting is often restricted to an average (reduction at time t_0 and increase at time t_1) shift of zero, i.e., the total amount of energy consumed cannot be influenced. More uncommon classes are exchangeable loads and interruptible loads. Exchangeable loads are loads that can be replaced, usually by using another energy carrier like gas instead of electricity.

A more abstract approach is to model the flexibility of DERs based on at least one interval from the options power, energy, time, and power ramping. This allows the application of the model to DERs of all types, e.g., generation units [96], EVs capable of discharging [97], and EVs incapable of discharging [98]. In those papers using such intervals for representing the space of feasible schedules, most commonly an interval for power is found. When assuming time dependent intervals, the model becomes even more versatile. Please keep in mind that, due to using this kind of model in a DER specific way, the three listed examples are considered direct exploitation. Nevertheless, there are examples using such models in a generic way, for instance [99] and **[314]**, which are listed below as examples for the exploitation of abstracted flexibility.

Table 4.4: Direct exploitation of flexibility in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
D				EV	Discrete		[100]
D				EV	Continuous	Max power to either be consumed or shifted	[101]
D				EV	Continuous	Time and energy interval	[98]
D				TCL	Continuous	Bucket	[102]
D				TCL	Discrete	Turning off considering a set of rules	[103]
D				TCL	Discrete	Minimum on- and off times	[104]
D				TCL	Discrete	Forecasting load from shifting signal using an ANN	[105]
D				TCL	Discrete	Optimization using a virtual battery model	[106, 15]
D				TCL	Discrete	For climbing, stable and recovery period regressions of response on time and outside temperature	[107]
D				TCL	Continuous		[108]
D				TCL	Continuous	Parameterization of a common model	[85]
D				TCL	Continuous	Up-/downward flexibility with hourly capacities	[109]
D				TCL	Continuous	Up-/downward flexibility	[110]
D				TCL	Disc. & Cont.		[111]
D				TCL	Discrete	Virtual battery model	[112]
D				TCL	Disc. & Cont.	Virtual battery model encoded in an ANN	[113]
D				✓	Discrete		[114]
D				✓	Discrete	Time shiftable	[115, 116]
D				✓	Discrete	Requests (energy, duration, and time preferences)	[117]
D				✓	Continuous	Up-/downward flexibility	[118]
D				✓	Continuous	Shiftable loads, interruptible loads	[119]
D				✓	Discrete Continuous	Time shiftable loads Curtailable loads, shiftable loads with average zero	[120]
D				✓, TCL	Discrete	Time shiftable	[121]
D				✓, TCL	Continuous	Shiftable loads, curtailable loads	[122]
D				✓, TCL	Discrete Continuous		[123]
D				✓, EV, TCL	Continuous		[124]
D				✓, EV, TCL	Continuous	Loads provide a power interval and availability for each time step and optionally either (minimum total power) or (comfort requirement, allowing zero consumption when met)	[125]
D				✓, EV, TCL	Continuous	Time shiftable loads and curtailable loads send requests	[89]

Table 4.5: Direct exploitation of flexibility in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
D			EV		Continuous		[126]
D			EV		Continuous	Time dependent power and energy intervals, energy usage	[97]
D			✓		Discrete		[127]
D			✓, EV		Continuous		[128]
D			✓	EV	Continuous		[129]
D			✓	TCL	Continuous		[130, 131, 132]
D			✓	✓, EV, TCL	Continuous	EV with availability and discharge, loads as (power, duration, time interval) or (load profile, timer interval)	[133]
D		✓		TCL	Continuous		[134, 135]
D		✓		✓	Continuous		[136]
D		✓		✓	Continuous	Shiftable loads with additional restrictions	[137]
D		✓		✓	Continuous	Aggregated load response function $f(t, t_r)$ giving power for time t and response duration t_r	[138]
D		✓, RES		✓	Continuous		[139]
D		✓, RES		✓	Continuous	Power interval for loads	[140]
D		✓, RES	EV		Continuous		[141]
D		✓	✓	✓	Continuous	Load curtailment	[142]
D		✓	✓	✓	Continuous	Up-/downward flexibility (loads)	[143]
D		✓	✓	✓	Continuous	Shiftable loads, curtailable loads, loads within ramping bounds	[144]
D		✓	✓	✓	Discrete Continuous	 Shiftable loads with average deviation of zero	[145]
D		✓	✓	✓	Discrete Continuous	Loads are time shiftable	[146]
D		✓	✓, EV	✓	Continuous	Load curtailment	[147]
D		✓, RES	✓	✓	Continuous	Load curtailment	[148]
D	Agg				Continuous	Power interval	[149]
D	Agg	✓, RES	✓	✓	Continuous	Aggregated load with ramping and power limit	[150]

Closely related approaches found in the review are the communication of upward and downward flexibility, the concept of the bucket, battery, and bakery [151], as well as the virtual battery model [112, 113]. Upward and downward flexibility of power, which is not to be confused with ramping flexibility, may be expressed in terms of intervals, which is essentially equivalent to providing a power interval. If, for example, the power interval allows control of the DER from 0 kW to 10 kW and a power of 3 kW is scheduled, then the up-/downward flexibility is given by the interval $[-3, 7]$. In other words, the intervals sent to the external entity are simply shifted by the scheduled power. The bucket, battery, and bakery are fixed intervals for power and energy width added constraints [151]. Lastly, virtual battery models are used for DERs witch characteristics similar to batteries, which includes TCLs, EVs, and, of course, batteries [152]. The virtual battery model is, again, the combination of an interval for power and one for energy with an added constraint. In contrast to the bucket/battery/bakery taxonomy, the virtual battery model considers dissipation [152, 106].

Aside from these results, the search query yielded some more particular approaches. For example, schedules may be generated from requests for consuming power, as seen in [117] and [89]. A request in [117] is defined as a tuple of consumed energy, the duration over which the energy is consumed and an array specifying preference levels for each time period. The already discussed distinction of curtailable and time shiftable loads is the foundation for scheduling in [89]. Another approach is letting the flexibility providers determine parameters of an advanced standardized model known by the external party, think of a MILP, for example. In [85] this is done with TCLs. The distinctive feature is that the flexibility providers themselves are obliged to send the required parameters and also update them if necessary. Interval based models like virtual batteries may also be used as standardized models, but are listed separately to point out their distinctiveness, as an advanced standardized model can be formulated to capture more complex relations. In [138] a function $f(t, t_r)$ called “response characteristic” describes the change in load for a request at time t with duration t_r. This load response function is known for each load. For the optimization, an aggregate of several response characteristics is computed. Since, here, the aggregate is computed by the external entity for the purpose of optimization rather than communication, this is an example for the direct exploitation instead of exploitation of abstracted flexibility. A related approach is the three-stage regression model proposed in [107], which predicts the response of TCLs to a DR signal. The duration of the DR event is split into a climbing, a stable response, and a recovery period, each covered by its own regression function. Given the current time and temperature, the individual regression models predict the response as a percentage. Using these models, it is possible for an external entity to estimate the impact of sending a (discrete) DR signal and therefore plan out the operation. In [113], a virtual battery model is encoded in an ANN. The ANN, which uses a long short-term memory architecture, forecasts the state of the virtual battery based on the previous state and the control signal. This can be used for estimating the consequences for a given series of control inputs. As this ANN model is created by the external entity, following the same logic as before, [113] is another example for the direct exploitation. Please note that this is closely related to **[308]**, in which, however, the ANNs are trained for the sole purpose of transmitting them to the external entity. The latter is therefore an example for the exploitation of abstracted flexibility.

To summarize the findings, direct exploitation is commonly achieved by one of the following options:

Precise model: Detailed DER specific models, closely replicating the real dynamics.

Classification of characteristics: Utilization of common models for DERs, especially loads, with shared characteristics as a compromise between detailed modeling and abstractness. Common classifications consider the ability to shift (proportionately or as a whole) and curtail loads. Less common classes differentiate exchangeability and the ability of being interrupted.

Intervals: Intervals for power, often combined with intervals for energy, time, or ramping capabilities, are often found in the literature. Together they define the feasible region for the respective variables. Using multiple intervals requires keeping track of the individual variables as they are all connect, e.g., energy as the integral of power. A higher abstraction is possible by making the intervals time dependent. For more specific use cases, additional constraints may be added. Both, the bucket/battery/bakery taxonomy, and the virtual battery approach are based on a combination of energy and power intervals with added constraints.

From this discussion, it is apparent that many of the approaches listed above could be used to describe multiple types of DERs. Then again, the listed papers do not utilize these models in a generic way in the sense of applying them for heterogeneous DERs.

4.3.2 Exploitation of Abstracted Flexibility

Papers investigating the exploitation of abstracted flexibility are listed in Table 4.6. It is further distinguished between two cases, the employment of a shared model for multiple types of DERs and the utilization of an aggregated model for multiple DERs. As several approaches for the direct exploitation are also suitable for this pattern, it is not surprising to find them again in Table 4.6.

Most common are models based on intervals. The bucket/battery/bakery taxonomy is used in several works. New, in comparison to the direct exploitation, is the power node model, which combines generators, storages, and loads in a single formulation. It uses intervals for energy, produced and consumed power, and the ramping of both. Virtual battery models can also be found. The parameterization of a given, well-known model plays a more prominent role in the exploitation of abstracted flexibility than in the direct exploitation. While multiple identified papers propose the use of generic model formulations [96, 153, 154, 155], the use of such a model for the purpose of communication is only discussed in [96]. Nevertheless, the remaining papers make use of generalized models and are thus relevant for this pattern. Of all examples, the authors of [154] propose the most sophisticated model. Their approach allows the description of DERs by selecting and combining features from a set of 14 different properties. The options include, but are not limited to, time restrictions, interruptibility, losses, multiple operation modes, and ramping. Therefore, all types of DERs can be described with this model. Scheduling on the basis of requests is found more often in this pattern, too. As explained before, in this thesis, the existence of requests and offers is not considered as constituting a market unless their prices may vary. In [156] and [157] the utilization and aggregation of so called flex-offers is discussed. A flex-offer is a tuple comprising a start time interval and a series of slices, which consist of a time and an energy interval [156]. The possibility of establishing a market for flex-offers is outlined in [157], hence the additional classification as market-based exploitation. Packetized energy management, with DERs sending requests for consuming or injecting energy, is addressed in [158]. Services, offered as tuples determining the amount of energy, maximum power, and the time of availability, are a subject of [159].

The remaining publications discovered with the search query are based on learning abstract representations, determining cost, demanding schedules within given intervals, and building highly descriptive models. Abstract surrogate models are learned in [62] and **[313]**. The descriptions are explicitly created for the purpose of communication and are therefore prime examples for the exploitation of abstracted flexibility. In [62] the feasible region of a DER is encoded by learning an SVDD from a set of feasible load schedules. The SVDD may either be used as a classifier, or to compute approximate projections of infeasible load schedules onto the set of feasible schedules. Please see Section 2.4.5 for more details on SVDD.

Table 4.6: Exploitation of abstracted flexibility in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
A				TCL	Aggregated	Bucket	[160]
D					Discrete		
A			✓	TCL	Shared	Packetized energy management (requests for consuming or injecting energy) as long as complying with QoS bounds, default control else	[158]
A			✓	EV, TCL	Sh. & Agg.	Power interval	[99]
D					Continuous	Power interval	
A			✓	✓, EV, TCL	Aggregated	Virtual battery model	[152]
D					Disc. & Cont.		
A			✓, EV	✓, TCL	Aggregated	Time dependent power and energy intervals	[161]
A		✓		TCL	Aggregated	Power node	[162]
A		✓	✓		Shared	Parameterization of a standardized virtual battery model with ramping and time dependent energy interval	[96]
A		✓	✓	✓	Aggregated	Power node	[77, 163, 164]
A		✓	✓	✓	Aggregated	Modified power node	[165]
A		✓	✓	✓, TCL	Shared	Parameterization of a common model, shiftable loads	[153]
A	Agg				Shared	Parameterization of a common model	[155]
A	Agg				Aggregated	Microgrid optimizes within target power interval	[166]
A	✓				Shared	SVDD encoding feasible regions	[62]
A	✓				Shared	Time dependent power and energy intervals with time constraint, and proposed additional constraints	**[314]**
A	✓				Shared	Approaches for exploiting flexibility using ANNs: (Classification), (Generation), (Repair), (Classification using deviations)	**[313]**
A&I					Sh. & Prices	(Forecasting load for a given price)	
A	✓				Shared	Bucket/battery/bakery	[151, 167]
A	✓				Sh. & Agg.	Flex-offer	[156]
A	✓				Sh. & Agg.	Assignment (Flex-offer, interval for total energy)	[157]
M					Sh. & Agg.	Flex-offer market	
A	✓				Shared	Cost curve giving cost for deviation in energy	[168]
A	✓				Shared	Services as tuple (energy, max power, time horizon)	[159]
A	✓				Shared	Loads (+/-), storages, and dependencies	[169]
A	✓				Shared	Parameterization of a common model	[154]

ANNs are another option for learning and encoding flexibility, which we proposed in **[313]**. In **[313]** we present five approaches for the ANN-based exploitation of flexibility. While we also call them "patterns" in **[313]**, they are not to be confused with the five general exploitation patterns discussed in this chapter and thesis. All five ANN-based modeling approaches allow the external entity to generate a set of feasible load schedules, which can then be used to select the most beneficial option. A more elaborate discussion is part of Chapter 5.

Cost curves which specify the cost for deviating from the schedule by a given amount of energy are investigated by [168]. An alternative to fixed target schedules is the consideration of target intervals. In [166] microgrids optimize their own operation to achieve an aggregated schedule within the provided target intervals. Finally, a noteworthy approach for the description of DERs and their flexibility is presented by [169]. The paper proposes a data model which was built with the aim of being generic, but at the same time allowing a detailed description of DERs and their interdependencies. In its conclusion, the need for deriving a suitable optimization model is acknowledged. Apart from that optimization isn't addressed.

In summary, similar to the direct exploitation, the exploitation of abstracted flexibility is often achieved on the basis of only a few different approaches.

Intervals: Most common is the utilization of intervals to describe multiple types of DERs. Aside from the bucket/battery/bakery taxonomy, the power node model makes a frequent appearance in the results.

Requests and offers: As requests and offers are often highly abstracted, they can be used to achieve exploitation of abstracted flexibility. The flex-offer is most prominent in the results. As noted before, the examples listed in Table 4.6 are not considered market-based exploitation, as they are submitted without prices.

Parameterization of a common model: When a model needs to capture more complex relations than possible with the two previous options, sophisticated standardized models come into play. In such a case, the flexibility provider and the external party have knowledge of a standardized, i.e., common, model. The flexibility provider determines the parameters and sends them to the external party. If required, updated parameters are computed and transmitted.

Examples of data-driven methods are scarce in the list of papers returned by the search query. When it comes to models that are trained from data for the purpose of communication, of all results, only [62] and **[313]** remain. Some more examples of such models uncovered during further research are presented throughout this thesis. Nevertheless, purely data-driven approaches are currently rather uncommon.

4.3.3 Market-based Exploitation

The second most common pattern found in the results is the market-based exploitation. It is not surprising that often real existing markets are considered. As upward and downward reserve can be seen as intervals, interval based models also appear frequently in the results. When it comes to custom requests and offers, the results are more diverse. Usually offerings include a combination of either power or energy, and a point in time. Sometimes the power or energy is expressed as an interval. A noteworthy abstraction is given in [170], where offers comprising a series of load levels and durations are considered. In few cases like [171], [172], [173], and [174] different offer and request types are distinguished. Aside from the one instance already listed in the section on abstracted flexibility [157], the search yielded another example of a flex-offer market [79].

Only few identified papers stand out in the sense of taking unique approaches. The most different approaches are presented in [175], [176], [171], and [177]. With a focus on markets, a derivative financial instrument that allows to interrupt or fully decline the feed-in of electricity at delivery time is investigated in [175]. In [176] an inversion of the indirect exploitation pattern is developed. The authors propose a price signal, sent from the flexibility provider to the external entity, in order to create a dynamic regulation mechanism which is interpreted as a market mechanism. Prices are computed from the marginal cost and the gradients of Lagrange multipliers evaluated at the current set point. Then, on the basis of the individual price, the subsequent set point is determined which in turn leads to another price feedback. Lastly, [177] and [171] describe multi agent architectures. In [177] consumers request energy from prosumers and "generator companies", and in [171] single agents are authorized to establish temporary markets and select which bids are accepted.

To sum up these findings, the following common approaches to markets have been identified:

Existing Markets: Selling and buying energy on markets existing in reality.

Intervals: Especially upward and downward flexibility, which can be expressed as intervals, are frequently considered. This category overlaps with real reserve markets.

Requests and offers: Usually a combination of power or energy, and a time span is requested or offered.

Overall, the market-based exploitation shows many similarities to the exploitation of abstracted flexibility. On a second thought, this is not surprising, as markets generally use highly abstracted models for placing requests and offers. The major differences are the existence of the market itself, as well as the level of abstraction. For the exploitation of abstracted flexibility, much more advanced models have been proposed in the identified literature.

Table 4.7: Market-based exploitation in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
M				TCL	Shared	Flex-offer market derives target schedule	[79]
M				TCL	Shared	Day-ahead market	[178]
D					Continuous	Control by aggregator	
M				TCL	Shared	Offer (power reduction, duration, time)	[179]
I					Price		
M				TCL	Aggregated	Tertiary control power	[180]
D					Continuous	Control by aggregator	
M				✓, TCL	Shared	Offer (energy within specified limits based on forecasts)	[181]
D					Continuous	Shiftable loads, curtailable loads	
M			✓		Shared	Offer (load series >= 0, duration series, price)	[170]
M			✓	TCL	Aggregated	Generalized battery model	[83]
M			✓	✓	Specific	Consumers request energy from prosumers Loads have priorities	[177]
M			✓	✓	Shared	Multi-energy market	[182]
D					Continuous	Shiftable loads, curtailable loads, exchangeable loads	
M			✓	EV, TCL	Aggregated	Up-/downward reserve	[183]
M			✓	✓, TCL	Shared	Offer (price, quantity, quality, time)	[184]
M		✓			Shared	Option to interrupt or decline energy at delivery time	[175]
M		✓			Shared	Reserve markets	[185]
M		✓, RES			Shared	Up-/downward reserve	[186]
M		✓, RES			Shared	Power interval (active and reactive)	[84]
M		✓		✓, EV, TCL	Shared	Set point with price feedback	[176]
M		✓	EV		Aggregated	Up-/downward flexibility	[187]
M		✓, RES	✓		Shared	Up-/downward reserve	[188]
M		RES	✓	✓	Aggregated	Up-/downward flexibility	[189]
I					Price	Tariffs (ToU), (CPP) or (reduced tariff with use restrictions)	
M		✓	✓	✓	Sh. & Agg.	Offers (energy, price) or (energy interval, price)	[174]
M		✓	✓	✓	Shared	Buying energy from markets and aggregators	[190]

Table 4.8: Market-based exploitation in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
M		✓, RES	✓	✓	Sh. & Agg.	Requests and DER specific bids for fulfilling requests	[171]
M		✓, RES	✓	✓	Shared	Up-/downward reserve	[191]
I					Price		
M		✓, RES	✓	✓	Shared	Up-/downward reserve for generators and loads, demand bidding	[192]
D					Disc. & Cont.		
M		✓	✓	EV, TCL	Shared	Offer (price, quantity, time)	[193]
M		✓	✓	✓, EV, TCL	Shared	EVs and TCLs offer reserve capacity	[194]
I					Price	Prices for loads	
D					Continuous		
M		RES	EV	✓, TCL	Shared	Offers (energy) or (up-/downward reserve)	[173]
D					Continuous	Control by aggregator	
M	Agg				Aggregated	Upward flexibility (production)	[195]
M	Agg				Shared	Reserve markets	[196]
M	Agg				Shared	Offer (load curtailment, price, time)	[197]
I					Price		
M	✓				Shared	Offer (time interval for consuming or producing one unit of energy) and bonuses if given bundles of offers are allocated	[198]
M	✓				Shared	Offers (provision of energy in time interval), (provision of power in time interval) or (provision of shifting energy within time interval)	[172]
M	✓				Shared	Offers (single period power interval) or (multi period power interval, average modulation of zero)	[199]
M		✓			Shared	Offers as in [199] within dynamic ranges specified by DSO	[200]
M	✓				Shared		[201]
M	✓				Shared	Trading on energy markets	[202]
A					Sh. & Agg.	Bucket/battery/bakery	
M	Agg	✓			Shared	Ramping capacity	[203]

4.3.4 Indirect Exploitation

Some sort of dynamic electricity pricing scheme is commonly assumed in the literature. For the sake of brevity, it is not further distinguished between different implementations like, for example, time-of-use tariffs, critical peak pricing, or real-time pricing. Instead, they are all subsumed under the term "dynamic prices". Table 4.9 and Table 4.10 show those examples of indirect exploitation obtained in the search process. Often, there are no particular mechanisms associated with the dynamic prices, merely their existence is assumed and modeled. Although they are using tariffs that are not further specified, two papers [204, 205] stand out from the other search results by their utilization of learning algorithms for the purpose of optimizing schedules. While [204] uses Q-Learning, policy function approximation based on ANNs is investigated in [205]. In those instances that do discuss the formation of prices, iterative approaches and price elasticity-based approaches are popular. Iterative approaches repeatedly compute price updates, until some criterion for stopping is fulfilled. For example, a DSO may set nodal prices for load serving entities (LSE) [206]. On this basis, the LSE compute their optimal power as an aggregate of the controlled DERs' powers. This aggregated power is combined with a description of the feasible region for reactive power and then sent back to the DSO, who in turn may update the nodal prices. This procedure is repeated until the prices converge. More examples for iterative price updates can be found in the results. A noteworthy implementation of the indirect exploitation pattern is presented in [207]. It replaces the energy market with a continuous and strictly monotonically increasing price function, which computes the price for a given total demand. Such an approach may be implemented in a centralized or a distributed manner. As mentioned before, price elasticities are also a popular choice for determining prices. Although the pattern for indirect exploitation would allow the flexibility providers to determine their price elasticities on their own and share them, in the literature, elasticities are usually assumed to be either known or estimated by the external entity. Price elasticities can be used for estimating the change in load, based on a given change in price. When additionally using cross price elasticities, e.g., [208], [209], and [210], even intertemporal effects can be estimated. Albeit the approach shines through its simplicity, precision is, however, limited. This is especially the case when modifying prices for multiple time periods at once.

More specialized approaches are DER specific controllers incorporating cost in their optimization [211, 212], and promised schedules [213]. Such a promised schedule is defined day-ahead and deviation is penalized. Flexibility provision entirely without automated control of DERs is presented in [214], where the DR signal targets humans and requires their interaction for generating a response.

The general approaches identified in the review process can be summarized as follows.

Unspecified tariffs: The existence of dynamic tariffs is often assumed and modeled without discussing price formation or the origin of the tariffs.

Iteration: In iterative approaches prices are updated until a given stop criterion is reached. The loop typically involves updating prices based on schedules and computing schedules based on prices.

Price elasticity: Given a price elasticity the impact of price changes can be predicted. If additionally cross price-elasticities are available, intertemporal consequences, i.e., changes in load in one period in reaction to price changes in another period, can be estimated.

The approaches discussed here feature the great advantage of usually relying on few data, as simply some prices and sometimes schedules need to be communicated. Furthermore, the data is simple to interpret, and it is not possible to coerce a flexibility provider into acting, as long as prices are not allowed to skyrocket.

Table 4.9: Indirect exploitation in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
I				TCL	Price		[215, 216]
I				✓	Price	Q-Learning	[204]
I				✓	Price	Price elasticity	[217]
I				✓	Price	Promised schedules with penalties for deviating	[213]
I				✓	Price	No automation, needs human interaction	[214]
I				✓, EV	Price	Price and schedule updates	[207]
I				✓, TCL	Price		[14]
I				✓, EV, TCL	Price	Iterative update of load schedule and price	[218]
D					Discrete	Shiftable loads with given consumption, time shiftable loads, interruptible loads, exchangeable loads	
I			EV		Price	Feedback loop updates prices	[81]
I			EV		Price		[219]
D					Continuous	Time dependent energy constraints	
I			✓		Price		[220]
D					Continuous		
I			✓		Price	Policy function approximation using ANN	[205]
D					Continuous		
I			✓	TCL	Price		[221]
I			✓	✓, TCL	Price	Feedback loop updates prices	[82]

Table 4.10: Indirect exploitation in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
I		RES		TCL	Price		[222]
D					Discrete		
I		✓		✓	Price	Contracts with time variable prices and energy consumption dependent penalties	[223]
D					Continuous		
I		✓		✓	Price	Cross elasticities for load	[209]
M					Shared	Up-/downward reserve	
D					Disc. & Cont.	Generator schedule	
D					Continuous	Load curtailment	
I		✓, RES		✓	Price	Price updates until convergence	[224]
M					Shared	Up-/downward reserve	
I		✓, RES	✓	✓	Price	DER specific controllers	[211]
I		✓, RES	✓	✓, EV, TCL	Price	Cross elasticities and reduction potential for load	[210]
D					Disc. & Cont.		
I		✓, RES	✓, EV	✓	Price	Price elasticity for loads	[225]
D					Disc. & Cont.		
I	Agg				Price		[226]
I	✓				Price	(Cross) elasticities	[208]
I	✓				Price		[227]
I	Agg	✓			Price	(Aggregated) EVs (storages)	[228]
D					Continuous		
I	Agg	✓, RES		✓	Price	Until convergence 1. DSO generates nodal prices 2. LSEs optimize and return schedule and feasible set	[206]
D					Continuous	DERs send constraints to LSE and receive dispatch signal	

4.3.5 State Information Based Exploitation

Results classified as exploitation based on state information are further divided into two subclasses. Instances of the first class are presented in Table 4.11, showing results that rely on exchanging agent specific information. Stigmergy based exploitation can be found in Table 4.12. As a controller is generally not able to observe the world by itself, it is depending on the input of sensors. Therefore, on a technological level, there can still be communication involved in the case of stigmergy. A more detailed description how these two cases can be distinguished is given in Section 4.2. Some papers compare centralized and decentralized coordination mechanisms, and therefore discuss both. For these, the centralized versions are omitted, since they usually fall into the direct exploitation pattern and are very similar to the already presented examples.

Beginning with the exchange of information, it is further differentiated whether there is a coordinator present in the approach or not. The first case is labeled “coordinator”, the second “local update”. A coordinator is a centralized entity facilitating the data exchange and sometimes performing additional computations required in the exploitation process. In its simplest form, the coordinator is a shared database which holds and provides information, like a blackboard. It is important to analyze which information is provided by the coordinator, as stigmergy based coordination can also be built around a coordinator (see [18, p. 53]). Roughly speaking, when changes of the provided information can be attributed to individual agents, it is not stigmergy [18, p. 45].

Often, updates are calculated successively by one flexibility provider at a time. The coordinator then receives the result, performs own calculations, and passes the results to the next flexibility provider. This is repeated until a given stopping criterion is met. In the papers collected in the review process, most commonly states or schedules are exchanged. Potential objectives include achieving a given aggregated schedule, e.g., [229] and [230], and minimizing cost, e.g., [231]. Depending on the design of the information flow, aggregated power can either be estimated or precisely computed. In regard to the flow of data, the approaches presented in [232] and [229] are especially interesting. Both involve the stepwise propagation of knowledge. While in [232] constraints are shared until every agent knows all individual constraints, in [229] estimates are shared and improved. Some distributed methods, such as the “alternating direction method of multipliers”, are derived by splitting the optimization problem into subproblems and updating the Lagrange multipliers. Examples identified in the review are [233], [234], and [235]. Blackboards are used in [236] and [237]. In both instances the load schedules of all agents are visible on the blackboard. Therefore, they are not considered to be stigmergic. If, instead, the blackboard computed and provided only the aggregated schedule, the two examples would be based on stigmergy. A rather unconventional two-step optimization of grid areas is presented in [238]. Firstly, each grid-area optimizes on its own, and the results are exchanged. Then, each grid-area optimizes again, this time considering the external cost while manipulating the generation of the other areas. The best result computed by one of the grid-areas is implemented. The entries in Table 4.11 suggest that the consideration of aggregated flexibility is popular in this pattern.

For the observation based approaches, the “Via” column specifies which quantity is being observed. Combinations of multiple quantities are possible. The majority of entries in Table 4.12 relies on measurements of either voltage or frequency. Signals derived from the aggregated load are used in [239] and [240].

In summary, approaches using this pattern are often exchanging states or schedules. A minority of the identified papers rely on a central coordinator or a blackboard for making information available. When it comes to measurements, usually voltage or frequency is considered.

Table 4.11: Exploitation based on the exchange of state information in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
E				✓	Local update	Reaching aggregated target schedule by 1. Neighbors exchange their states 2. Agents estimate the total load 3. Agents adapt their schedule by a formula after waiting for a random time	[229]
E				EV, TCL	Coordinator	Alternating direction method of multipliers	[233]
E				EV, TCL	Coordinator	Until total power is acceptable, DERs read schedules, update their schedule and publish the result	[236]
E			✓	TCL	Local update	Aggregated schedule for buildings and BESS, buildings follow slowly, BESS covers deviations fast	[230]
E			✓	TCL	Local update	Battery controller receives predicted load from building controller	[212]
I					Price	Both know the electricity tariff	
E			✓	✓, TCL	Coordinator	Alternating direction method of multipliers	[234]
E			✓, EV	TCL	Coordinator	Until total power is acceptable, DERs read schedules, update their schedule and publish the result	[237]
E		✓			Coordinator	Iterative local updates by derived formula Coordinator can ask a generator for update	[241]
E		✓			Local update	States and output voltages of neighbors	[242]
S					Voltage & Frequency		
E		✓	✓	✓	Coordinator	Sequential optimization	[243]
E	Agg				Local update	Constraints repeatedly exchanged with neighboring TSOs, distributing knowledge	[232]
E	Agg				Coordinator	1. Grids optimize locally 2. Global optimization based on results 3. If necessary, grids update local results and step 2 is repeated	[244]
E	Agg				Local update	1. Grid areas optimize locally 2. Grid areas optimize considering external costs and manipulating external generation 3. The best result is implemented	[238]
E	Agg				Local update	Microgrids optimize, exchange schedules, and then repeat the optimization	[245]
D					Continuous	Control by microgrid	
E	Agg		EV		Coordinator	Aggregators send schedules to DSO. Lagrange multiplier based updates of aggregator schedules on violation of network constraints	[235]
M					Aggregated	Up-/downward reserve offered by aggregator	
E	Agg		✓	✓, TCL	Local update	Repeated optimization based on forecasts and aggregated schedule until equilibrium is reached	[231]

Table 4.12: Stigmergy based exploitation in the literature.

Pattern	Generic	Generation	Storage	Load	Via	Noteworthy Abstraction	References
S				TCL	Frequency		[246]
S				TCL	Voltage & Feeder load	Controller prioritizing voltage ≻ feeder load ≻ tariff	[239]
I					Price		
S				EV, TCL	Voltage & Delta power	HP	[240]
D					Continuous	EV	
S				✓, TCL	Frequency	Loads with discrete and continuous control signals	[247]
S		RES			Voltage		[248]
S		RES	✓		Voltage		[249]

4.4 Modeling Flexibility with Regard to Communication

On the basis of general modeling considerations, the specified exploitation patterns, and the literature review presented previously, this section aims to identify and categorize general approaches to the modeling of DERs and their flexibility. By relating these approaches to the individual exploitation patterns, a systematic overview is created.

4.4.1 General Modeling Considerations

When developing a model, some fundamental questions should be answered in the process. A list of such questions is given by [20], with the first two asking why the model is needed and which information is sought-after. For the topic of controlling DERs, the "Why?" question can be answered with the motivations identified in Section 4.1. Given the motivation, the required information is a description of how a DER may be influenced. However, there is no general rule stating which data is needed for this purpose. Even solutions employing identical exploitation patterns can be very dissimilar and there is a strong dependence between the pattern and the required information, as the literature review shows. In the discussion of how DERs are abstracted for the individual exploitation patterns, general approaches have been identified and outlined. These form the basis for the following discussion.

Typically, three methodologies for modeling are distinguished in the literature: white-box, black-box, and grey-box modeling. An explanation of each is provided in Section 2.3.2. With respect to the description of DERs and buildings, this classification has been used for reviewing thermal modeling [22], HVAC modeling [58], and the modeling of buildings [57]. White-box models are the most commonly found type in the presented literature review, where equations are often linear or quadratic. In general, equations are by no means limited to these two types. Black-box models are data-driven [58, 22]. Examples from the literature review are [62], **[313]**, and [113]

It is also possible to distinguish top-down and bottom-up approaches, which can be further differentiated into a macro-economic and engineering-economic perspective [57]. Interdependencies between the TSO and DSO level and their implications for objectives and constraints are discussed in [250]. In the top-down view, the objective is governed by the TSO who seeks to activate DERs and the constraints are posed by the DSO, in the bottom-up view this relation is inverted, the DSO specifies the objective and the TSO the constraints. This understanding can easily be generalized to flexibility providers and external entities. From this perspective the direct exploitation is closely associated with a top-down view, while the market-based exploitation and especially the indirect exploitation generally follow the bottom-up view. For the remaining two patterns, both views are commonly found.

4.4.2 Condensed Results of the Literature Review

The findings of the literature review regarding how flexibility is modeled are summarized in Figure 4.3. It shows the individual general approaches to modeling (white colored boxes) used in each pattern (blue colored boxes) ordered by their abstractness. The ordinate is split in three lanes representing the number of DER types that is generally targeted with identical model formulations, i.e., whether a model describes one type of DER or multiple types. Essentially, the ordinate can also be interpreted as degree of abstractness. Green colored boxes are specific to the implementation of certain exploitation patterns and highlight special schemes of communicating information, indirectly related to modeling. However, the patterns and approaches are by no means limited to these communication schemes. How much area is covered by each box does not convey any meaning. Overall, the figure aims to depict a generalized perspective on modeling, outlining universal capabilities. It is easy to create situations not depicted: Imagine, for instance, a DSMgr who controls a portfolio of DERs consisting only of CHPPs and does so by directly switching them on and off. While the employed model is highly abstract and listed under the exploitation of abstracted flexibility, the scheme itself falls under the direct exploitation pattern.

Direct Exploitation

As established before, white-box models describe physical relations. However, the detailedness may vary vastly. While linear equations are quite popular, they are usually greatly simplified representations. With the help of quadratic, nonlinear, or even differential equations, much more detailed descriptions can be achieved. Highly detailed and *precise models*, independent of the types of equations, are one possibility to model flexibility identified in the review of the direct exploitation pattern. Since such a model captures very much information and is very specific, this approach also exhibits the least level of abstraction.

In the context of modeling buildings, the selection of representative instances is a method to reduce complexity caused by diversity [57]. By grouping together buildings with similar characteristics, a single model may suffice to describe each instance of the group. Such a grouping can be achieved by classification or clustering [57] and is essentially the *classification of characteristics* approach identified for the direct exploitation pattern. Since the process of defining such a common representation requires the harmonization of DER specific constraints, individual details are lost. Therefore, common models are more abstract than precise models. As, in this thesis, the line between direct exploitation and exploitation of abstracted flexibility is drawn based on the types of DERs that are modeled identically, per definition, a generic generator, a generic storage, and a generic load are the most abstract models used in the direct exploitation. Once a model is shared for multiple types, for example, a virtual battery that describes BESSs and TCLs, it is considered exploitation of abstracted flexibility. There are various proposals for such common models in the literature. Classifications like curtailable, shiftable, and interruptible have already been pointed out in the review. The majority of these classifications focuses on loads and are therefore associated with the direct exploitation pattern. However, there are more comprehensive taxonomies. An example not identified by the search query is the so-called "energy flexibility interface" [251]. It distinguishes uncontrollable and time shiftable devices, buffers, and unconstrained DERs. As the buffer class is not only meant to be used for storages, but also thermal systems, this is an example for abstracted flexibility.

The review suggests that representing flexibility with the help of *intervals* is also a popular approach. Like the already mentioned virtual battery (e.g., [112, 106, 15]), interval-based models also fall into the shared characteristics category. Nevertheless, an extra category is introduced in this thesis in order to draw a distinction, especially since intervals on their own are more abstract. Models based on intervals make use of one or multiple intervals, e.g., power and energy, and may add some additional constraints, e.g., a given minimum amount of energy consumed. The virtual battery model, and also the bucket/battery/bakery taxonomy are regarded as such interval based models. In summary, for the direct exploitation, the three identified general approaches to modeling are *precise models*, *classification of characteristics*, and *intervals*. However, as the discussion shows, the last two options may also be used for the exploitation of abstracted flexibility.

Exploitation of Abstracted Flexibility and Market-based Exploitation

In regard to abstracted flexibility, the classification of characteristics, as well as intervals have already been named. In the review, the *parameterization of a common model* has been identified as another possibility for implementing

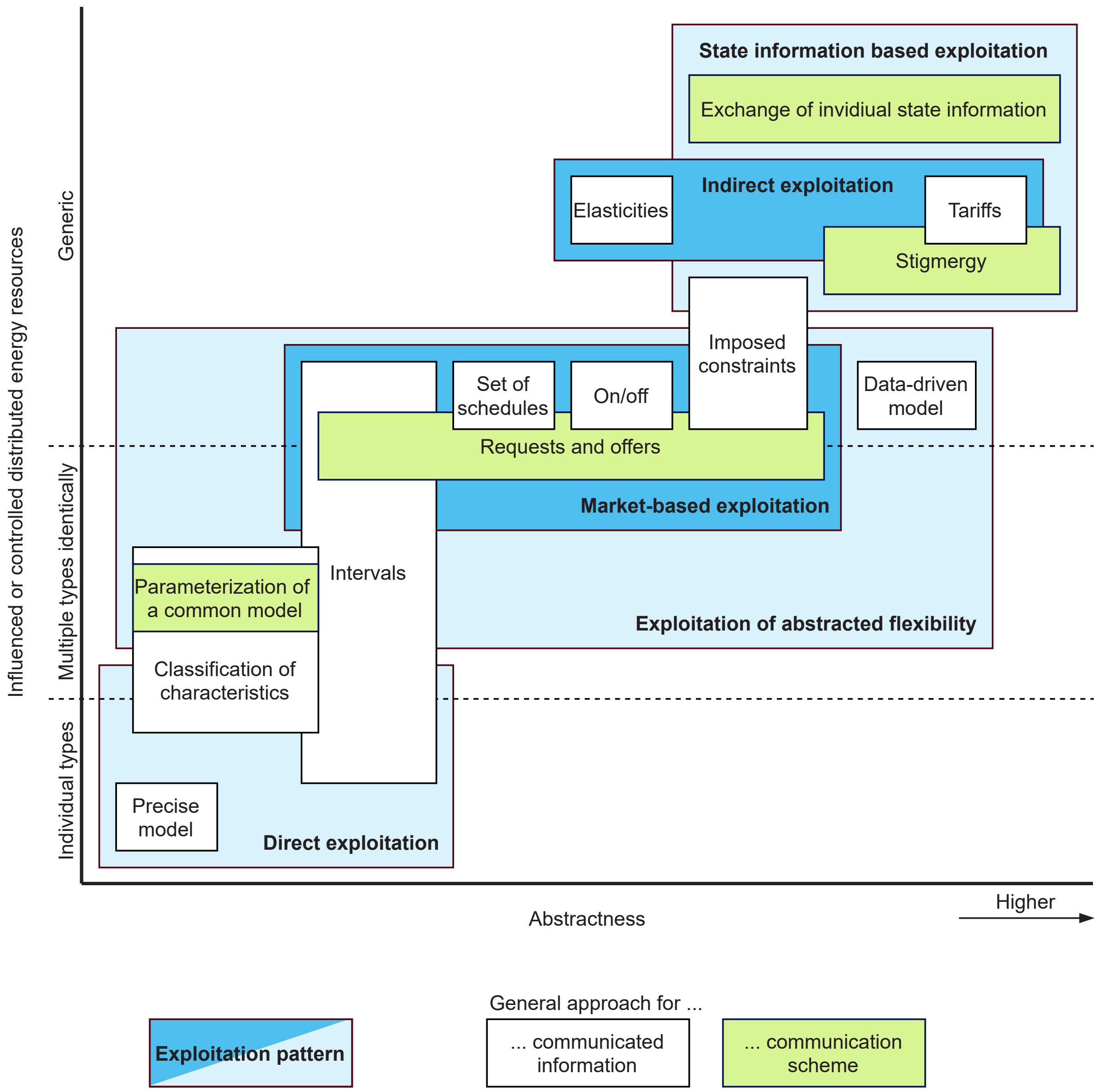

Figure 4.3: Modeling of flexibility in regard to communication.

this pattern. On a closer look, the parameterization of a common model is essentially an implementation of the classification of characteristics. The major difference is that the parameters for multiple DERs are bundled in a single description. Consider, for example, a given, well-known MILP with constraints $Ax \leq b$. For the parameterization of a common model, the EMS of the flexibility provider determines and transmits the matrix $\mathbb{A}$ and vector b to the external entity. By using a formulation with adaptable constraints, e.g., with an option to deactivate unnecessary ones, the model itself can be applied more flexibly.

Another popular approach is the coordination based on *requests and offers*. This is again a specific implementation of the pattern and not directly limiting modeling options. Nevertheless, it is an important mechanism, as it is also the core of the market-based exploitation. Remember that the market-based exploitation requires bids to be priced. If bids are submitted without prices or any other measure of utility, the approach is considered to be an exploitation of abstracted flexibility. Therefore, the market-based exploitation is a special case of the abstracted flexibility pattern, and hence, requests and offers are very relevant to both patterns. In contrast to offers, which are, by the meaning of the word itself, bids or proposals, requests may either be compulsory orders or not. Offers are usually involved in market processes and submitted by flexibility providers. Since they are offers, the acceptance and implementation is not mandatory. For requests, on the other hand, it is important to consider the origin. Since for the exploitation, the flexibility providers must be influenced, requests submitted by the provider to the external entity are generally proposals asking for permission to perform certain actions. Requests sent from the external entity to the flexibility provider, e.g., asking to turn a device off, are usually compulsory.

Obligations resulting from requests or offers being accepted are not reflected in Figure 4.3. When offering control reserve, for instance, the acceptance of a bid does not necessarily mean that it will be called. Also, the mechanisms for actually providing flexibility may greatly vary and even follow other exploitation patterns. Take, for instance, primary control reserve, which observes the grid frequency and activates proportionally to the measured deviation. This kind of provision is an example for stigmergy. Hence, for a detailed analysis, it is also necessary to identify such combinations of patterns. Similar to the abstracted flexibility pattern, the review revealed many examples for interval based modeling of flexibility. For instance, offers for upward and downward flexibility, e.g., in the form of spinning reserve, are often found. Since these offers allow the market, that is, the external entity, to choose any power level within the minimum and maximum power, it is considered an interval based model. In comparison to the interval based models for the abstracted flexibility pattern, the market-based patterns tend to be more abstract, i.e., they use fewer intervals and fewer additional constraints.

Regarding abstractness, one of the most abstract ways of actively quantifying flexibility is to provide a *set of load schedules*. Assuming there is at least one valid load schedule, which must always be the case for an error-free operation, a set of load schedules can describe any constraints. However, this comes at tremendous costs, as there are several questions which need to be answered for making this approach truly feasible. Firstly, there is no best practice to the generation of alternative load schedules. One possible way could be switching on and off additional constraints that manipulate the aggregated load during the optimization. Nevertheless, this requires additional optimization runs for each added alternative. It also immediately poses the question of how these constraints should be formulated and varied. Next are the questions of how many profiles must be generated and how it can be measured from a set of load schedules, in case there is a need to measure the flexibility. Although these questions are not answered, there are some papers using this approach. For instance [252], which assigns disutility to the individual alternatives, which may be interpreted as a price and therefore market-based exploitation.

By reducing the length of the schedules the set of load schedules becomes more and more abstract, until at some point there is only one time period, i.e., one single power value, left. Hence, the most abstract option, in terms of lost details, is to describe the possible operations for a single time step. While this could again be an interval, the interval based models are often not as simple. Therefore, Figure 4.3 lists the ability to switch a DER *on or off* as the most abstract offer on a market. This is also equivalent to submitting a bid for a specified amount of energy. The acceptance of the bid can essentially be interpreted as an on/off signal.

Another option which has been identified for the exploitation of abstracted flexibility is the utilization of *data-driven models*. Since these black-box or grey-box models can be used to approximate any data more or less exactly, they are considered to be the most abstract approach. It is, for instance, possible to train an ANN to provide information on whether and when DERs can be switched on or off, as **[308]** shows, for example. Given the various options for implementing a data-driven model [58] and the realization that there is often more than one way of representing flexibility with such a model **[312, 308]**, the total number of implementation options is enormous.

Imposing constraints is a method for controlling flexibility providers that bridges the gap between the (market-based) exploitation of abstracted flexibility and the state information based exploitation. In this approach, the external entity defines and communicates constraints, which the flexibility providers must meet. Thereby the allowed options to control DERs are restricted. Quota based restrictions of the power drawn from or fed into the grid are a popular implementation of this approach. The quota itself is a percentage of a nominal power. This means, a quota of 100% poses no restrictions, while a quota of 0% forces the DERs to shut down their operation. Since either feed-in or consumption need to be reduced, and never both at the same time, they are restricted by individual quotas. However, there are no implemented examples for this in the literature review. Nonetheless, even though quotas are not discussed in **[314]**, it is related to a quota based approach. In project *grid-control*[1], which **[314]** belongs to, quotas were imposed upon a market (see [253]) and therefore indirectly upon the smart buildings, which offered flexibility to a market participant. Quotas play an even more prominent role in *grid-control*'s successor *flexQgrid*[2], which directly targets smart buildings with all their DERs. In *flexQgrid*, the building itself is a market participant and thereby able to perform quota based trades in order to mitigate the imposed restrictions or profit from selling unused capacity. Overall, imposing constraints is not restricted to quotas. On the one hand, as the project *flexQgrid* shows, it is possible to build a market around this approach, hence, imposing constraints can be part of a market-based exploitation. On the other hand, since the flexibility provider is still able to act more or less freely as long as they adhere to the additional constraints, this approach is also considered to be state information based exploitation. Whether it is classified as stigmergic or not depends on the exact implementation. A non-discriminating quota, i.e., a quota that is identical for all flexibility providers, can be considered stigmergic, as it provides aggregated information for all agents and it is not possible to trace the actions of individual agents.

Indirect Exploitation

With the direct, abstracted, and market-based exploitation, all three approaches building on the communication of models, parameters and states have been discussed. The remaining two patterns, the indirect and the state information based exploitation, usually do not involve the communication of models or parameters. It is, however, easy to incorporate the transmission of models into such approaches, e.g., an EMS estimating its price elasticity and informing an external entity. Regarding the abstractness of *elasticities*, switching between discrete operation modes, like turning a DER on and off, is ranked equally abstract as allowing to control a continuous set point. Here, the term continuous is again referring to the available control options and not time. The reasoning is that these two alternatives are complementary. While a single (non-degenerate) interval cannot model discrete options without adding discrete variables, discrete options for control can only approximate a continuous interval. Since

[1] `http://projekt-grid-control.de/`
[2] `https://flexqgrid.de/`

elasticities are used to estimate a continuous response, the approach has been ranked to be equally abstract as an approach allowing discrete control. However, elasticities on their own do not cause any response of the flexibility provider. They must be accompanied by a signal, most commonly *tariffs*, able to incentivize a change in behavior. Typically, elasticities are assumed to be known or determined by the external entity, in which case flexibility is not communicated. In contrast, the previously mentioned EMS estimating and transmitting its current elasticity is indeed communicating flexibility. When tariffs are not tailored to single individuals, that is, when they are not discriminating, they generally reflect the current state of the electrical grid and energy system as a whole. Therefore, they can be interpreted as a type of state information, and hence the overlap of the indirect pattern with the state information based pattern. Nevertheless, if tariffs are directly targeting single flexibility providers they cannot be considered state information based exploitation.

State Information Based Exploitation

Finally, the state information based pattern is regarded as the most abstract way of coordinating flexibility providers, since it does not need incentives to control behavior and may even involve no communication at all. The two central schemes in this pattern directly relate to the exchanged information. The *exchange of individual state information* allows sharing data from which information on single individuals can be deduced. A popular example would be the individual load schedules. Given this information, each agent is able to see how its neighbors react and how everyone contributes to the solution. The complementary scheme, using aggregated information, is *stigmergy*. While stigmergy can be implemented without any active communication, there may still be a central entity providing signals (see also Section 2.2.2).

A tool to achieve coordination that has not yet been mentioned in this discussion is iteration. Iteration until a certain criterion is reached is commonly used in the state information based exploitation, but also found in the indirect pattern. However, other patterns may also profit from iteration, especially when using highly abstracted models. While the direct exploitation, for example, directly offers all required data to assess all available options, this is often not the case for highly abstracted models. In cases where it is unclear if a good solution or even a feasible option is found in a single run, iteration can help to narrow down the search space and improve the solution.

4.4.3 Modeling Flexibility from a Data-Driven Perspective

As only few data-driven flexibility models were included in the literature review, this section presents more such models and attributes them to the basic tasks arising in the context of statistical and machine learning (see also Section 2.4.2), with the goal of contributing to a better understanding of the relevant concepts. A more elaborate discussion of such surrogates and a systematic derivation of ANN-based approaches follows in Chapter 5. Generally speaking, any model capable of generating feasible load schedules can be used to represent flexibility, when understanding the flexibility of DERs as a set of feasible load schedules (see Section 2.1.3 and **[312, 308]**). The term “load schedule” can be used without loss of generality, since, in the case of single period flexibility, individual power levels are simply represented by schedules consisting of only one element.

Classification: Classification can be used in multiple ways to model flexibility. For example, a classifier may be used to distinguish feasible from infeasible load schedules. With the help of such a classifier model, it is possible to identify valid options for controlling DERs. This can be achieved by classifying whole schedules **[312]**, or by using multiple classifiers covering smaller parts of the schedule [12]. The latter approach has the advantage to significantly simplify the generation of load schedules, as the individual parts of a schedule are generally easier to guess than a whole schedule at once. Another application of classifiers could be the detection of certain predefined properties or events which indicate how DERs may be influenced, e.g., by identifying valid control options for the following time period **[308]**.

Regression: In general, regression means the estimation of a one dimensional output given some input [27, p. 101]. It is hence especially useful for myopic flexibility calls, as only a single time period needs to be considered. One option could be to estimate a (one period) load response for a given signal. Alternatively, a price elasticity could be estimated from the current state provided by the flexibility provider.

Structured output: Structured output produces vectors in which the single elements are related [27, p. 101]. In the context of modeling flexibility, structured output can be seen as a generalization of regression, returning a vector instead of a scalar. Examples for structured output are the patterns for generation, load profile forecasting, and repair presented in **[313]** and **[312]**. Other approaches that can be named are the SVDD based repair of load schedules [87], the step-wise creation of load schedules using multiple classifiers outlined above, and state-based models like state machines.

Synthesis and sampling: Synthesis and sampling is closely related to structured output. Its goal is to generate samples similar to the training data [27, p. 101]. Models like generative adversarial networks could be trained to synthesize load schedules, as we already proposed in **[312]**. In [254], general adversarial networks have already been used to synthesize load profiles in non-intrusive load monitoring tasks.

Probability density/mass function estimation: Tools which can be used for modeling uncertainty are probability density and probability mass functions. Generally speaking, approaches from the regression and structured output categories can be transformed to use probability density/mass functions.

Clustering: Clustering itself does not produce a model with inputs and outputs, but a set of clusters. Clustering of load profiles, for example, can help in evaluating DSM decisions [255]. In the case of [255], clustering is proposed as a tool for analysis, not communication. Nevertheless, a flexibility provider could perform a clustering and communicate the resulting clusters. Previously it was stated that there is no known best practice for selecting representative load schedules, i.e., generating a set of feasible schedules, to bundle them in an offer. Depending on the particular use case, clustering could be one approach for generating such a set.

In summary, regression, structured output, as well as synthesis and sampling comprise models with the potential to directly generate load schedules. Some schedules created with these approaches can be interpreted as forecasts, others are more closely related to random samples. If suitable algorithms are used, classification approaches like the SVDD or the cascade classifiers can also be used as generative models. As already stated, the utilization of such a model on its own does not constitute a certain exploitation pattern. It is important to distinguish whether the model is communicated or not. One example is [87]. While the authors propose SVDD as a way to communicate sets of feasible schedules, their proposed coordination scheme is an example for state information based exploitation, since the model is never communicated.

4.5 Exploitation of Flexibility in the Research Project C/sells

In this section, approaches developed and discussed in the German research project *C/sells*[3] [4], funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) as part of the SINTEG (Smart Energy Showcases – Digital Agenda for the Energy Transition) program, are classified and discussed on the basis of the previously presented patterns and findings. This does not only contribute to a better understanding of the approaches developed in *C/sells*, but also show how the framework of patterns and approaches derived in this chapter can be applied. While the project is focusing on a cellular energy system, none of the concepts discussed here is limited to a cellular system. Overall, with trying to pool as many DERs as possible, there is a strong tendency to highly abstracted models.

Exploitation of abstracted flexibility: Abstracting DERs is one of the central approaches in *C/sells*. A quantification for the flexibility of heat generating units, such as CHPPs and HPs, is proposed by [256]. It uses metrics derived from predicted thermal energy corridors. Such a corridor basically describes the set of feasible thermal load schedules of a DER.

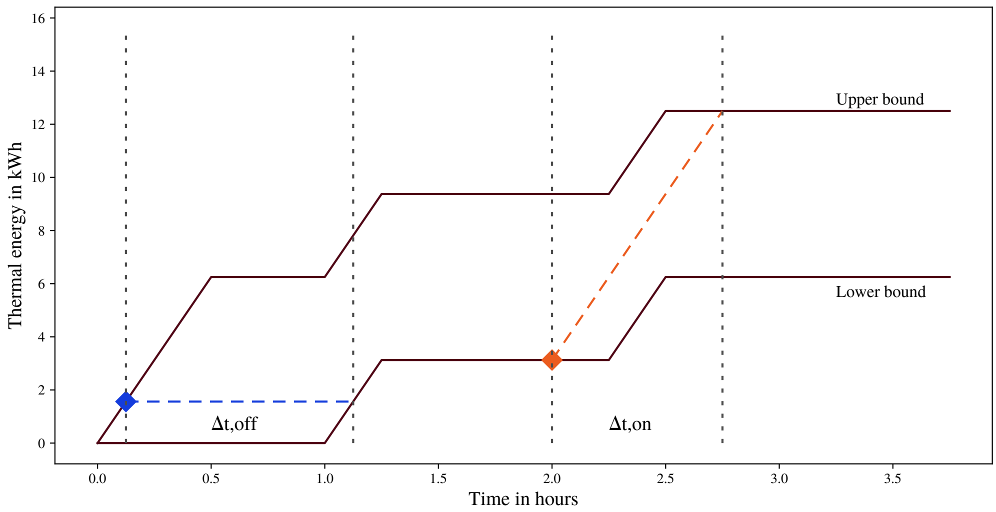

Figure 4.4: Abstracted flexibility of a single CHPP, based on [256].

An example is depicted in Figure 4.4. It shows a corridor for a CHPP and a heat storage. The CHPP is able to provide a thermal output of up to 12.5 kW and around 6 kWh of thermal energy can be stored in the heat storage. The lower and upper bounds are derived by forecasting the local heat demand and determining the points in time the storage is empty or full. The operational zone of the DER lies between the upper and lower energy bounds. Essentially, the corridor is a series of time-dependent intervals for the allowable total production of thermal energy.

[3] `https://csells.net/`

The first proposed metric measures the ability to delay the operation of DERs. It is determined by calculating the amount of time until the lower boundary of the corridor is reached, when no further thermal energy is produced. Starting from the blue diamond on the left, the horizontal line shows the CHPP being shut down and idling as long as possible. The second metric measures the opposite, activating the DER with maximum power. In this case, the time until the upper bound is reached is computed. This is depicted by the orange line starting from the orange diamond.

The approach is not limited to describing single DERs, an aggregation of multiple DERs is also presented. In the aggregated case, different units may be unavailable during some time steps. This is dealt with by determining the actually enforceable power in addition [256]. Since generators and TCLs are considered, the resulting sign of the changes in electrical power may be positive or negative. Additionally, as not only multiple types of DERs are considered, but also the aggregation of DERs, the proposed quantification is a perfect example for the exploitation of abstracted flexibility.

In **[313]** we propose the utilization of ANN-based surrogate models for modeling flexibility and thereby enabling its exploitation, which is also a central topic of this thesis. As already mentioned in Section 4.3.2, the paper presents multiple patterns for identifying or generating feasible load schedules. These patterns are further investigated in **[312]**, coming to the conclusion that it is indeed possible to identify or generate feasible load schedules with the help of ANNs. Combining the two best performing patterns of **[312]**, we suggest an iterative approach to generate feasible load profiles step by step in **[308]**. An in-depth presentation and discussion of this ANN-based modeling is provided in Chapters 5 and 6.

Market-based exploitation: Markets are a prominent coordination mechanism in the project. A total of three local market platforms has been proposed and implemented. Each platform has its own unique characteristics. The individual markets are named *ALF*, *Comax*, as well as *ReFlex*, and are outlined by [257].

ALF, the “Altdorfer Flexmarkt”, is a day-ahead market described more closely in [258] and [259]. It has no limitations regarding potential sellers, but considers the DSO as the sole buyer. The platform provides a registry for enlisting offers and the grid operator’s requests. Market clearing is executed once daily. Sellers may offer two different products, a “schedule product” and a “long-term contract” [258]. The latter is a contract for participating in a direct exploitation scheme for integrating small DERs from residential buildings. Even though this particular contract is no market-based approach, it is listed here, since it is integrated in the market clearing process [259]. The schedule product is outlined as a “time-series in 15 min steps including available power, price and potential constraints” [258]. Unfortunately, a thorough and conclusive presentation of technical details or models is provided by neither of these sources.

The descriptions of Comax and ReFlex are based on [257]. Comax does not only target DSOs, but also TSOs. Hence, market participants require enough capacity to have an impact on the ultra-high voltage level. Offers define a load schedule and possible deviations for the considered time slots. Resulting requests are allowed to call the offered power partially, i.e., to select arbitrary values from the offered interval.

Finally, ReFlex is again designed to cater to the needs of DSOs. On ReFlex two products can be traded, one resulting in an exact request and one imposing constraints in the form of a quota. Remember that a quota in the context of DSM is a constraint on the maximum consumption or feed-in, generally expressed as a percentage. In contrast to the exact request, quotas are more suitable for DERs highly affected by stochastics. On ReFlex, as [257] states, the quotas are intended to control the behavior of EVs and the variable production from RES. Influencing devices by setting additional restrictions and thereby narrowing the set of feasible schedules is rarely considered, as the review presented in this thesis indicates. A special feature of employing this method is that the flexibility providers largely remain in control of their DERs. More conceptual information on all three markets can be found in [257]. However, it is not sufficient for a more detailed analysis of the respective models.

Aside from the three market platforms, there are more concepts for markets and their associated products. A home energy management system managing DERs and participating in a flexibility market is investigated in [260]. In particular, generation units, storages, as well as the two types of loads, EVs and TCLs, are controlled by the EMS. The EMS places offers to deviate from its planned schedule in positive or negative direction. Offers for a given time step comprise power, energy, and a price. While the power is fixed, a call does not require to exhaust all the offered energy [260], it can be understood as an interval instead. Using this model, once flexibility has been requested, new offers reflecting the resulting situation need to be generated.

Indirect exploitation: Incentives are the basis for the "Cell cluster Karlsruhe East". The cell cluster demonstrates how smart buildings can collaborate to solve congestions. It builds upon the regional EMS presented in [261] and comprises the FZI House of Living Labs, the Energy Smart Home Lab at the Karlsruhe Institute for Technology, and additional 54 simulated smart buildings. Figure 4.5 depicts the cluster and the simulated grid. The cluster itself constitutes another cell on higher hierarchical level. It is managed by a cluster EMS that observes the electrical grid and operating resources, and controls the smart building if necessary.

Requests are sent according to the smart grid traffic light concept. There are different ways of implementing this concept. Here, the traffic light phase is determined by the observed and estimated states, including transformer temperature, voltages, and currents of the lines. Forecasts are not considered in the process. Instead, there are upper and lower boundaries, for each traffic light phase, respectively. The boundaries for the yellow phase are narrower than those of the red phase, meaning a red phase is always preceded by a yellow phase. Once a boundary is violated, the associated traffic light phase is proclaimed. If a yellow phase is detected, the cluster EMS begins to request reactive power from smart buildings with inverters, and determines suitable candidates for sending incentives. These candidates receive a set of multiple alternative tariffs. Each of the candidates computes the load schedules resulting from each individual tariff and sends the resulting set to the cluster EMS. Then, the cluster EMS chooses the most beneficial load schedule and grants the respective tariff to the building. The building, in turn, must now follow the reported schedule. Should these measures not suffice and a red traffic light phase occur, buildings can be forced to react.

The central mechanism in the cell cluster is the indirect exploitation pattern, because incentives are the central mechanism. Nevertheless, the concept involves elements of direct exploitation in regard to the requests for reactive power and the red traffic light signal. The implementation of the indirect pattern is rather unusual, but adds some predictability found in the abstracted flexibility pattern. By generating a bundle of different tariff options and asking the building's EMSs for their aggregated responses to each tariff, a limited form of control based on abstracted flexibility is established, as the external entity, that is, the cluster EMS, can choose from a small set of feasible schedules. A more detailed discussion of this concept and how tariff options can be derived is presented in [261]. In contrast to the original mechanism, in the cell cluster Karlsruhe East, there are no further price incentives during the red traffic light phase.

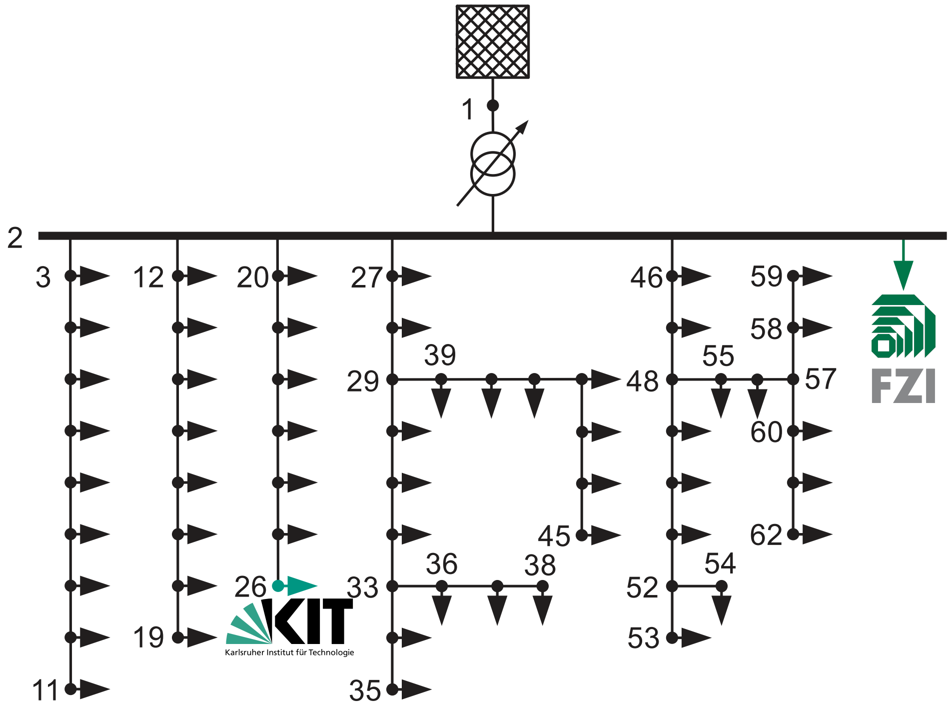

Figure 4.5: Cell cluster Karlsruhe east, including the Energy Smart Home Lab, the FZI House of Living Labs, and 54 simulated smart buildings. The 62 depicted nodes comprise all buildings, power grid nodes, and the transformer.

Overview: Finally, an overview of the modeling approaches discussed in this section is presented in Figure 4.6. The illustration makes use of the classifications developed in this chapter.

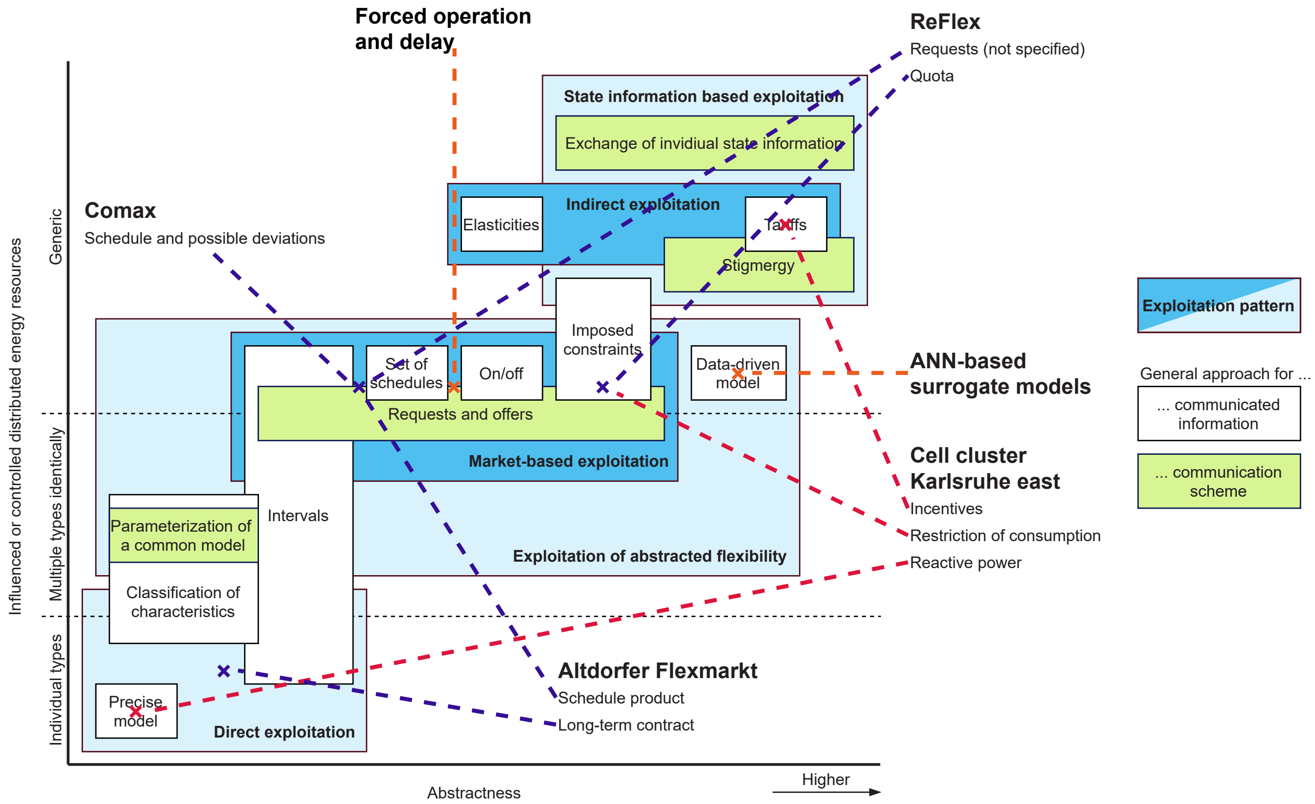

Figure 4.6: Exploitation of flexibility in C/sells. Overview of the approaches presented in this section.

4.6 Summary

The literature proposes and makes use of many different exploitation and coordination mechanisms. With the help of the exploitation patterns and identified general modeling approaches, it is possible to make comparisons of different solutions and point out their distinct features, as has been demonstrated for selected approaches from the research project *C/sells*. However, the presented framework is not only suitable for the analysis of existing solutions, but may also be used to plan and select mechanisms in future DSM applications, since it provides guidance which modeling approaches are especially suited for which exploitation pattern. In the following, the framework is used to derive approaches for the utilization of ANN-based surrogates.

5 Artificial Neural Networks as Surrogate Models for Flexibility

From a general perspective, surrogate models are very flexible in their application. This is not only shown by the manifold of examples named in Section 4.4.3, but also our previous publications, were we presented and evaluated multiple ANN-based approaches for modeling flexibility using quite diverse data as inputs and outputs **[312, 308]**. As surrogates are generated from data samples, that is, pairs of input and output data of the function to be approximated, surrogate modeling primarily falls under the data-driven category in the classification of modeling approaches presented in Section 4.4. Furthermore, surrogate modeling offers the possibility to train models in an automated process, therefore, given a suitable modeling approach and the availability of the required data, eliminating the need to formulate and implement models manually. A trained model can not only be used in the local operation of DERs, but also communicated. Nevertheless, the exploitation of flexibility based on the communication of surrogate models is a very uncommon concept, as the literature review showed. ANNs being a very powerful and versatile tool is the main motivation for investigating ANN-based surrogate modeling in this thesis. But even though the focus lies on ANNs, the presented concepts are not limited to the utilization of ANNs. The systematic for classifying modeling approaches introduced in the previous chapter is the basis for identifying, developing, and testing approaches to use surrogate models for the flexibility of single or multiple DERs. This chapter presents the core of this thesis and aims to answer the research question **RQ2** "What is the quality of approximation ANN-based surrogate models can achieve?", which is again subdivided into more particular questions.

RQ2.1 What are the advantages and disadvantages of using surrogate models?
In the context of this first question the utilization of surrogates as models for flexibility is motivated. To do so, Section 5.1 presents a literature based overview of benefits and challenges associated with surrogate modeling, and draws conclusions regarding the modeling and communication of flexibility.

RQ2.2 How can ANNs serve as surrogates for the flexibility of DERs?
Section 5.2 outlines how the communication of flexibility with ANN-based surrogates could be implemented in a real-world application and, for this purpose, identifies several approaches to encode the flexibility of DERs into ANNs. The general concepts presented by us in **[313]**, **[312]** and **[308]** are all incorporated in this section. A selection of the most promising approaches is presented in Section 5.2.5.

RQ2.3 What is the quality of the trained ANN-based surrogate models?
Finally, Section 5.3 presents a detailed evaluation of the most versatile approach, establishing a baseline for the quality of approximation achieved by ANN-based surrogates.

A summary of this chapter is given in Section 5.4.

5.1 Benefits and Challenges of Surrogate Modeling

This section provides a concise overview of arguments for and against the utilization of surrogate models for modeling flexibility. Firstly, a literature based overview of benefits and challenges is presented. Secondly, possible ways to overcome some of these challenges are illustrated. Lastly, everything is put into the context of modeling flexibility. The insights gained in this section are the basis for the subsequent considerations and, therefore, the results of this thesis.

General benefits: Benefits mainly arise from the replacement of costly computations or real world experiments with much simpler surrogate models. Even more so, when there is no known algebraic model of the system [24], or a black-box needs to be modeled [25]. Solving optimization problems and simulation are examples for computationally expensive tasks that can benefit from using surrogates [262, p. 409]. This is especially true, when derivatives of the involved functions are complicated or absent [24] and, therefore, gradient based methods are unsuitable. Overall, surrogate models can help to reduce the computational burden by simplifying the object of study [25]. To which extent a simplification is achieved is, however, depending on the exact problem and also the employed type of surrogate. A deep neural network with many hidden layers, for instance, is obviously much more resource demanding than a simple linear regression model. Given there is a reduction of complexity, surrogates can help by improving the efficiency of algorithms, such as evolutionary algorithms [70], and by speeding up processes like parameter fitting, parameter analysis, and sensitivity analysis [25]. Aside from the potential reduction in complexity and the accompanied speedup, a major benefit of surrogate modeling is its versatility, i.e., the ability to be used in various different tasks, while being accurate [24].

General challenges: Challenges are also strongly depending on the problem that is to be solved, and the model to be utilized as a surrogate. On the one hand, the different types of models offer dissimilar complexity, such as a varying number of parameters that have to be estimated, nonlinear and nonconvex terms, or a lack of suitability for global optimization [24]. Hence, there are model specific drawbacks, like the huge amounts of data needed to train ANNs [25]. Overall, the resulting surrogate should be cheap, smooth, and easy to optimize, while, at the same time, being sufficiently accurate for the individual use case [262, p. 409]. Aside from the type of model, there are many factors to consider in this regard. The accuracy and, therefore, the quality of the model, for instance, strongly depends on the sample data used to generate the surrogate [25]. On the other hand, model choice is influenced by the nature of the problem [263]. Selecting a model can be difficult and may be influenced by common practices and the experience of the designer [263]. From a general perspective, a selection criterion is required [26], specifying what the surrogate needs to achieve in order to be considered suitable. In order to define such a criterion, again, further implications need to be considered. Different criterions can vary in their interpretability, accuracy, bias, or computational efficiency [26]. Therefore, a middle ground needs to be found.

Overcoming these challenges: Solutions to these challenges are actively researched, and there are many approaches, as these issues are often not limited to the field of surrogate modeling. Take subset selection for regression, for instance. As the name indicates, it is a methodology tailored for regression models. It is used to select the input variables of the model from a larger feature set, with the goal of achieving better predictions. Since it is a general methodology for regression models, it is also one option to improve the accuracy of surrogates [24], if applicable in the specific case. Likewise, any approach to improve a model of a certain type can be considered in order to generate more precise or more compact descriptions in the form of surrogates. As there is a multitude of possible model types, the remainder of this discussion focuses on general approaches. A major requisite for many applications of surrogate models is, as mentioned before, efficiency. One way to improve the efficiency can be the combination of surrogate models of varying fidelity [264]. In an optimization problem, for example, a low-fidelity surrogate can be employed to search on a global scale, followed by a local search with a high-fidelity model. Since, in this case, it is not necessary for a single model to capture all relations between inputs and outputs, simpler and thus more efficient models may be used. Another key factor for all surrogates is the sample data, since a surrogate can only pick up relationships that are present in the data used to generate the model. Employing a suitable sampling strategy is hence important. Given the use case allows the generation of more than the initial samples, adaptive methods can help to improve the model accuracy [25, 24]. In contrast to a one-shot design, which involves the generation of just one model, the adaptive design evaluates the resulting model, and, if needed, repeats the model generation with additionally and newly created samples [25]. Using this adaptive method, it is possible to identify points of interests, e.g., the locations of extreme values predicted by the surrogate model, and generate the actual response to the input to obtain a new sample. This methodology is especially relevant for surrogate based optimization, to build a more precise representation step by step (compare [24]). The definition of a criterion for the model generation is another issue named above. There are usually multiple options, not only for measuring the performance of the resulting model, but also for model selection itself. For example, there are plentiful measures for assessing prediction errors, including the Mean Squared Error, Average Euclidean Error, Harmonic Average Error, and many more (e.g., [26]). Likewise, there exist multiple criteria for model selection, including metrics like the Akaike Information Criterion, hold-out error, and cross validation error [26]. One possibility to deal with the trade-offs between individual objectives is to make use of a multi-objective surrogate generation approach. In [26], three ways of implementing such an approach are listed. Firstly, the combination of multiple objectives into a single one. Secondly, the sequential enforcement of objectives, and lastly the simultaneous enforcement. All of these, however, come with new challenges, including an exponentially growing solution space with an increasing number of criteria [26].

Modeling flexibility: Modeling flexibility with the help of surrogate models offers even more benefits, but also poses new, additional challenges. As surrogates approximate outputs from given input data, any data could be used for trying to generate surrogates, i.e., train them. Therefore, in theory, surrogates open up many possible ways of modeling flexibility. Furthermore, due to this versatility, they can be utilized as generic models for any DERs to be represented, potentially reducing the need to build hand-tailored models. Whether a surrogate is suitable for the intended task or not, and if there is really no need to build models from hand, is, however, use case specific. For a surrogate model to pose as a model of flexibility, the central requirement is the ability to derive how DERs are operated from the given input and output data. In Section 2.1.3 it is established that this can be achieved with the help of feasible load schedules. Hence, if the surrogate allows the deduction of feasible load schedules, it can be seen as a model for flexibility. There is, of course, no limitation to load schedules. Any other control signal which can be used to deduce distinct instructions is feasible, too. However, there are many parameters to consider and which determine whether approaches are practical or not. These include the intended input and output data, as well as the choice of model, for instance, regression models, SVMs, or ANNs.

One of the core features of surrogates named above is their efficiency. The need for small, efficient models is mainly motivated by the intention to use the much faster surrogate instead of the computationally expensive original. Efficiency is, furthermore, a desired quality for speeding up processes like optimization. If the objective is, however, to approximate the behavior of the original as closely as possible, more complex models may be required. There is again a problem specific trade-off, that is, whether the algorithms for solving the problem benefit more from efficiency or accuracy. A more efficient surrogate allows the computation of more iterations in the same time span, but comes at the cost of a decreased precision. If the actual feasibility of a load schedule is the main criterion, such a surrogate may be too imprecise. In this case, more sophisticated surrogate models should be employed, even at the cost of a decreased efficiency. Two possible ways of alleviating this issue have already been named in this section. Both involve generating and using multiple surrogates for describing an identical flexibility. Firstly, a low-fidelity model could be used for all computations requiring efficiency, followed by corrections made with the help of a high-fidelity surrogate. Secondly, the solution space, i.e., the flexibility itself, may be split into smaller parts represented by smaller and more efficient surrogates.

Another relevant challenge is the availability of samples for generating the surrogate. When only observed data is to be used, i.e., the model should be learned from a physical system, there may be too little data, or the data may not reflect the true flexibility, as the surrogate can only learn about alternative operational choices if the data presents them. Take a building with an EVSE, for example. If connected BEVs are always charged at the highest power, the surrogate can only learn to charge BEVs as fast as possible, even though the EVSE may support slower charging rates. This issue can be tackled by willfully deviating from the planned operation in order to explore new states and collect novel samples. Additional information, such as admissible set points, may also be provided by DERs. One method to accelerate the training of ANNs, and thereby getting by with fewer data, is to start with an ANN trained on a similar task rather than a completely new one. This approach is called transfer learning (see [27]). Additionally, given the modeling approach is compatible, dataset augmentation could be used to obtain more training data (see [27]).

When few data is available, updating or newly generating models gains importance, since more data can be collected as time passes. Updates need not necessarily be provided by the flexibility provider. Given the external entities can record the required data, they can compute updates themselves.

5.2 Encoding the Flexibility of Distributed Energy Resources in Surrogate Models

In order to answer how surrogate models can be automatically generated, it is first necessary to specify what exactly they should replace. It has already been established that surrogate models approximate functions $f(x)$ and are generated from input and output samples $(x, f(x))$. Therefore, it is necessary to define what these inputs and outputs are. Several possible combinations have already been named in the previous chapter, and also been proposed in previously mentioned work, including [63, 65, 12, 312, 308]. They are not simply reproduced in this section, but instead considered in the systematic deduction of possible surrogate modeling approaches. Please note that even with this structured procedure, it is not possible to identify every sensible approach. Depending on the use case and how the problem to be solved is tackled, completely different possibilities for creating such models may emerge. Therefore, it is not the goal to provide a complete list of all possible surrogate models for flexibility. Building on top of the classification introduced in Section 4.4, which can be seen as a general framework for the analysis and development of communication related flexibility modeling approaches, this section derives and discusses opportunities for ANN-based surrogate modeling. But first, the general circumstances and requirements are considered in more detail, illustrating how a practical application based on such flexibility surrogates would look like.

The exploitation pattern: When utilizing surrogates, the *exploitation of abstracted flexibility* introduced in Section 4.2 is used. Here, the abstracted flexibility transmitted to the external entity is the surrogate combined with the knowledge of how to derive choices. Using this information, the external entity is able to derive the expected consequences of different choices, select fitting options, and send them back to the flexibility providers for their implementation. With the utilization of ANNs, all presented modeling approaches fall under the *data-driven model* category identified in the previous chapter and depicted in Figure 4.3. The exploitation of abstracted flexibility, as depicted in Figure 4.2 in Section 4.2, comprises multiple steps, namely, the generation, transmission, and storage of the model, as well as the exploitation itself.

External entities: An external entity may belong to any role existing in the energy system. In other words, any external entity, no matter their role, could make use of this process to exploit flexibility. Exemplary roles include, but are not limited to, aggregators who manage large collections of DERs and participate on energy markets, distribution system operators, transport system operators, and regional EMSs as found in the research projects *grid-control* **[310]** or the *C/sells* cluster Karlsruhe presented in Section 4.5.

Flexibility providers: From a technical perspective, flexibility providers range from single DERs to large ensembles of interconnected DERs. Generally, an EMS manages the devices. In case of a single DER, the necessary functionality for generating the model and communicating with an external entity can be built directly into the device. If multiple DERs are aggregated, they are operated collectively, or all available information is simply merged together, a dedicated EMS is required. Owners of DERs are either obliged to participate with their devices or are incentivized, for instance by receiving monetary compensations.

Surrogates: Surrogates are not restricted to ANNs, even though ANN-based surrogates are the focus of this thesis. Other learnable models that are able to provide the same information, may perform equally well or bad. Whether the proposed surrogate modeling approaches allow a sufficiently good approximation, is depending on the specific use case. Some proposed combinations may not be suitable for all types of applications.

Generation of a model: Generating a model means learning the model, in case of ANN-based surrogates. For learning any type of model, it is of course a necessity that there is a relation between the intended input and output, that is, a sensible output can only be produced if the necessary information is present in the input.

In regard to the flexibility of DERs, there is usually a strong dependence on the current state of the individual DERs. Charging and discharging a BESS, for instance, is limited by its SOC. A surrogate, if technically able to do so, can take the state as an input in order to solve this dependence. If the surrogate is not able to handle such an input, it needs to be replaced with a newly generated surrogate every time the state changes. The process is exactly as depicted in Figure 4.2. The state of one or multiple DERs is essentially a vector of selected values which need to be monitored. In general, the state comprises all scheduling relevant information. In its simplest form, it is a concatenation of current measurements and state variables the EMS receives from meters, sensors, user interfaces, and DERs. These values can either be selected by hand, for instance, based on a list of usually relevant information, or by an automated process, e.g., utilizing methods like the already named subset selection for regression if applicable. In this thesis, state variables are selected by hand. Scheduling constraints arising from user requests also need to be included in the state vector, as they limit the feasible region and are thus scheduling relevant.

More sophisticated approaches may process these observed state variables to derive new ones, and use the newly derived data as an input instead. Such a preprocessing step could be used for different reasons, including the generation of more accurate surrogates, e.g., by filtering faulty measurements, obscuring of state data for privacy reasons, or the aggregation of multiple DERs. The latter may, for example, be achieved with autoencoders (e.g., [113], where an ensemble of TCLs is aggregated), which open up the potential for larger scale applications by aggregating a multitude of devices.

From a more general perspective, the availability of data itself is a prerequisite. Hence, in order to make use of an ANN-based surrogate modeling approach and automatically learn a flexibility model, some unit, such as an EMS, must collect all relevant data over a period of time via some interface. Aside from the already named state data, some approaches require additional information, which either has to be derived by an EMS itself, or must be provided by the DERs in addition to their state. Given an approach, it is usually obvious which device should provide or derive the data. If the data describes a single DER, the DER should provide it. If it describes the joint operation of multiple DERs, the EMS should compute it. For instance, the interval of possible charging powers a BEV can currently be charged with is data the DER itself should provide. Whether a building with multiple DERs can hold its total electricity consumption below a given threshold or not, should be determined by the EMS. The challenge of training surrogates has already been discussed in the previous section, pointing out transfer learning and data augmentation as possible remedies when there is too little data. Moreover, the need for the data to reflect the true flexibility has already been explained and is not repeated here.

Transmission and storage: Right after a model has been generated by the flexibility provider it is transmitted to the external entity, where it is also stored. Depending on the ANN topology, the surrogates can vary considerably in their size, usually ranging from kilobytes to many megabytes. Hence, at least initially it is necessary to transmit a larger amount of data from each flexibility provider to the external entity. In practice, the external entity would offer some interface for the EMSs to connect to and send their data. The external entity saves the collected surrogate models in some kind of database.

Exploitation: For the exploitation step, the flexibility providers send their current states to the external entity, who then derives the flexibility choice, for instance, a schedule. As surrogates only generate approximations, the flexibility providers may not be able to exactly reproduce the intended schedules or implement other kinds of choices. Hence, either some deviation from the predicted behavior has to be expected and considered during the selection process, or measures have to be taken to guarantee a high likelihood of feasibility (compare. **[308]**). If high deviations must be expected, for instance, due to poor model quality, or if communication between flexibility provider and external entity is inexpensive and reliable, implementing a feedback loop is sensible. With such a feedback mechanism, the flexibility provider tells the external entity whether the choice is feasible or not, and may also hint feasible options. Depending on the data provided by the flexibility provider and the type of surrogate model, this may also enable the external entity to compute model updates and thereby improve the surrogate. Take, for instance, a surrogate that assigns a price to a load schedule, i.e., a schedule is mapped to a price. When a feedback mechanism is used, the flexibility provider responds to the schedule requested by the external entity with the actual price. The combination of selected schedule as input, and the price as output, can then be used by the external entity to update the surrogate and thereby improve the model's accuracy.

Deviations may also be considered in the form of uncertainty. Some types of models, such as kriging models or mixture density networks, do not only predict an output from a given input, but also provide information for estimating probabilities. Given such a model, this information could be incorporated in the flexibility exploitation process, e.g., by performing stochastic optimizations or for giving guarantees. Another possibility to help the external entity in deriving a choice is the provision of the planned load schedules by the individual flexibility providers. The load schedule can then be used as a starting point in order to derive possible deviations **[312]**, possibly simplifying the search. Whether planned schedules are available or not is more relevant for deriving choices, which is the focus of Chapter 6. Also, since planned schedules are optionally provided alongside the data needed as inputs for the surrogate models, they are not listed in the discussion below.

Overall, any combinations of these features could be sensible, depending on the use case and the intended functionality of the surrogate. Hence, such features are seen as possible additions to be considered when selecting a surrogate modeling approach, and not separately named for each individual approach described below. Please note that sometimes, it is possible to swap input and output data, and still have a theoretically sensible surrogate model. However, even though examples are outlined below, their feasibility is not guaranteed.

Keeping all this in mind, subsequently surrogate modeling approaches are derived, based on the presented patterns and the categorization developed in the previous chapter. Figure 4.3 is used as a reference to structure the procedure. Beginning from the least abstract pattern, that is, the direct exploitation, the individual patterns are analyzed until the state information based exploitation is reached. For ease of reference, approaches relevant for the remainder of the thesis are marked as such by providing their name in the paragraph header. Surrogates, as explained before, are built from input and output data. For this reason, the starting point to keep in mind for each category is the question of which data is available or can easily be derived. This question is closely related to how models belonging to each category are implemented and how the associated problems are solved. Please note that in the process of deriving and incorporating a surrogate model, generally, the associated exploitation pattern changes. After analyzing all patterns, a summary of the findings is given in Section 5.2.5, providing an overview of the most promising approaches.

5.2.1 Direct Exploitation

Three basic modeling approaches have been identified for the direct exploitation pattern in the previous chapter, namely, formulating (precise) models for each individual DER, shared models for DERs with similar characteristics, and models mainly based on intervals. Given the identified literature falling under this category, it can be stated that the employed algorithms for solving the investigated optimization problems are rather diverse. There are many examples, of which only some are named here, implementing heuristics [100, 103, 15, 115], linear programs [102, 120, 122, 124], quadratic programs [98, 111, 119, 132], or even nonlinear programs [85, 110, 136, 138]. They are often implemented as mixed integer models and, furthermore, the utilization of model predictive control is popular.

Precise Models

At a first glance surrogates and precisely formulated models seem to be mutually exclusive. While precise models are built to be detailed representations of reality, surrogates simply approximate some function, accepting imprecision. Depending on the exact circumstances, especially the relations between decision variables and the objective, different opportunities for deriving surrogates may arise. Simply approximating a target function $f(x)$ taken from an optimization model is outlined first in this section. Then, identified opportunities for employing surrogates for EAs and Markov models are outlined, followed by the already mentioned schedule repair found in the literature. For the remaining types of optimization models, no further opportunities have been recognized.

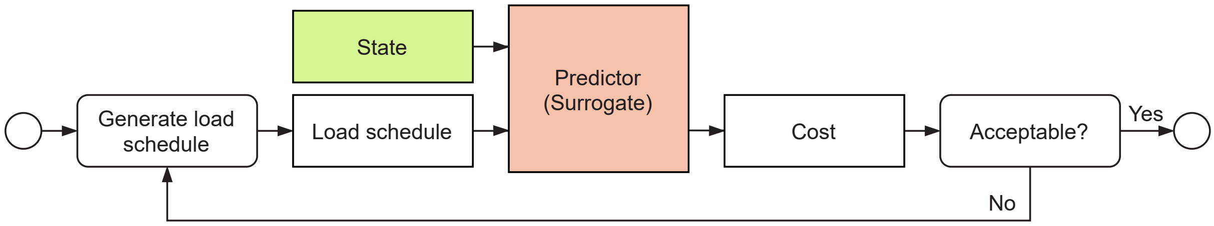

Figure 5.1: Cost based search and evaluation of load schedules.

Cost evaluation: Consider a model that has been tailored to precisely describe the processes in a building for the purpose of minimizing the total cost of operation, which is a task often found in the literature. Any other objective influenced by the DERs would work equally well, but since usually cost is the primary target, only cost is considered here. In order to minimize the total cost $f(x)$, the model must be able to compute this value based on some decisions x defining the operation of the available DERs. Given that x directly corresponds to the DER schedules, let x contain the schedule of each DER. Both, the schedules x and the target function value $f(x)$, are easily available for an EMS that optimizes the operation of DERs. For the purpose of modeling flexibility, it is sufficient if the surrogate simply approximates $f(x)$ for the external entity, for whom it serves two purposes. Firstly, it can be an indicator whether a load schedule is feasible or not. Similar to the penalty method from the field of mathematical optimization, invalid load schedules should result in a very large $f(x)$ value, signifying infeasibility in order to avoid it. Secondly, the cost provided by the surrogate is an estimate for the actual cost caused for the flexibility provider implementing the given load schedule. The external entity can make use of this information to search for a cheaper option that still satisfies all requirements. Figure 5.1 depicts the process of using this cost evaluation surrogate. The state in the green colored box and the surrogate in the orange colored box are supplied by each flexibility provider individually. It is stopped once an acceptable schedule has been found. Evaluating the cost in the stopping criterion is optional. Since the cost information may be leveraged in order to find valid schedules, which is outlined in more detail in Section 6.2, the surrogate can pose as a flexibility model.

Generation of load schedules inspired by evolutionary algorithms: Evolutionary algorithms offer a possible way to heuristically deal with computationally complex optimization problems. Please see Section 2.5.4 for a brief introduction. The EA framework, which is based around mapping abstract representations (genotypes) to possible solutions (phenotypes), is very flexible in its application and, therefore, suitable for many use cases. Given such a mapping, any phenotype, including DER schedules of arbitrary length, can be generated from a genotype. The utilization of surrogates in the context of EAs is not new. Several ways to integrate surrogate models found in the literature are summarized in [70]. All of these replace different steps of the algorithm, such as the initialization, mutation, crossover, and evaluation. As the mapping from genotype to phenotype is usually hand tailored, it is not considered for replacement with a surrogate model by any of the works referenced in [70]. However, in the context of modeling and communicating flexibility, this exact step is especially significant. As the DERs determine the objective value in an energy optimization problem, scheduling decisions for the DERs must be encoded in the genotype. In other words, given an evolutionary algorithm that optimizes a DER related objective, a mapping from genotypes to feasible load schedules is known. This mapping itself can be seen as a model for flexibility. By varying the input, that is, the genome, different feasible load schedules can be generated. Communicating this mapping may, however, pose some problems, as it is often a specific algorithm rather than a closed-form expression. As a consequence, the external entity would need to execute the various algorithms provided by different flexibility

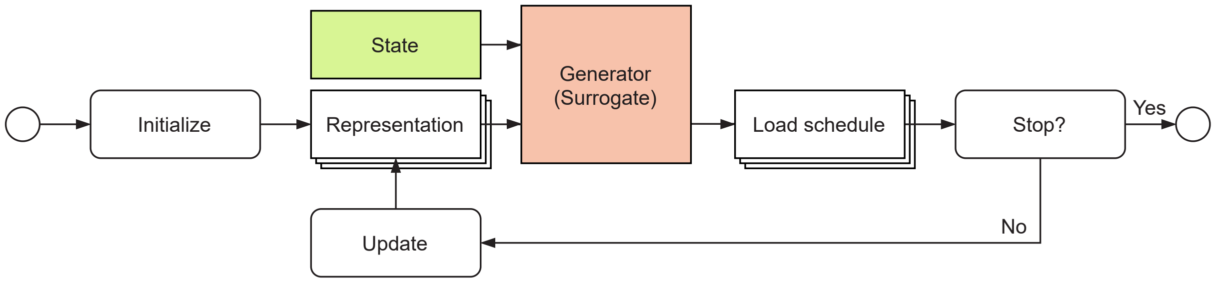

Figure 5.2: Generating load schedules from a representation.

providers. In such a case, surrogate models may be a suitable replacement. Using surrogates, there only needs to be a common understanding of the data structure and how the surrogate works. This rational is the core of the generator pattern we introduced in **[313]**.

It is, however, important to point out that there may be some restrictions to this approach, when starting from an existing EA. Even though the load schedule is a required output needed to compute a target function value, such as the overall cost for energy, the phenotype associated to a genotype is not necessarily a load schedule. Müller et al. [265], for instance, use an EA to select boundaries for a closed-loop controller which operates a BESS. Hence, an additional simulation of the closed-loop control is required in order to determine the load schedule. A more advanced and comprehensive implementation of this combined EA and simulation approach is presented by Mauser in [5], considering devices of all kinds. Even though the phenotype is not a load schedule, the result of the simulation is. Therefore, in this case, it is possible to map from a genotype to a load schedule and use this data to train a surrogate. Further limitations can arise if the EA has only a restricted ability to control DERs. If, for example, the EA is not allowed to charge a BESS with electricity from the grid, a learned surrogate will also deny it, even when the BESS is technically able to. In conclusion, an existing EA is a good opportunity to implement this surrogate modeling approach, as training data can easily be generated, but it is important to assess possible limitations beforehand.

Figure 5.2 depicts the process for generating load schedules from abstract representations. It is executed by the external entity, who makes use of the surrogate. In the first step, a set of starting points, i.e., the initial population of representations, is created. Then, with the help of the surrogate, each representation is mapped to a load schedule. If the surrogate works as intended, the schedule is likely to be feasible for the given state. The resulting load schedules are assessed by the external entity, and if one of them suffices for the intended use, the algorithm is stopped. If there is no schedule meeting the requirements, new representations need to be generated. There are different options to do this, including mutation and crossover. The process is then repeated with the updated population. After the algorithm has stopped, the best identified schedule, or alternatively the associated representation, is communicated to the flexibility provider as flexibility choice. Using this schedule generation process, the external entity is essentially implementing and executing an EA. Which methods for generating and updating the population, which may as well be a single individual, work best, is generally problem specific. Hence, knowledge from the domain of EAs is crucial to implement and tweak each step involved in this approach. From a general perspective, the external entity must be aware of the type of representation, i.e., if it is a binary string, a tuple of integers, or similar, in order to generate a population and updates. Knowledge on how this data can be interpreted is not required.

Overall, in order to use this EA inspired generation approach, three components are required. Firstly, the surrogates provided by the flexibility providers. The surrogate is specific to each flexibility provider, but may be based on a common specification, defining how certain elements of the abstract representations are interpreted. Such a common understanding of the representation may help in the development of the remaining two components. Secondly, an algorithm for sampling the initial population of representations and, lastly, algorithms for updating or combining individual representations. These algorithms may be generic, fitted for a specified genotype structure, or even supplied by the flexibility provider, e.g., in the form of surrogates as outlined in [70]. If the flexibility providers are already making use of EAs in their local optimization, all required data for generating the surrogates is easily available. Else, the underlying mapping must be implemented first. Fortunately, the field of machine learning provides tools that could be employed to automate this step (see [266]). Two basic ideas are outlined in the appendix in Section A.1. It is important to note that using these approaches comes with many new challenges. Furthermore, there is no guarantee that the learned representations are in any way meaningful.

State-based simulation inspired by Markov decision processes: Markov decision processes are another powerful modeling tool, which can be used for describing flexibility and optimizing its exploitation. A basic introduction can be found in Section 2.5.8. Since the MDP framework is highly generic and hence very versatile, it is one option for implementing direct exploitation approaches. In an MDP the state s_t of the environment at time t plays a crucial role. The environment state, from the perspective of a flexibility provider or external entity, comprises all variables relevant for optimizing schedules, as has been explained in the beginning of this section. For utilizing an MDP the Markov property must be fulfilled, i.e., the next state must only depend on the current state s_t and the selected action a_t. An exemplary state and action of a BESS could be $s_t = 50\%$ SOC and $a_t = 1000$ W of power provided by the device. If s_t does not suffice to explain s_{t+1} with a_t, it is missing information, which invalidates the model. Hence, in this approach even more so, it is important to make sure all relevant information is contained in the state vector.

The state s_t influences not only which actions are available, that is, $A(s_t)$, but also the probability distribution $p(s_{t+1}|s_t, a_t)$ of the subsequent state s_{t+1}, and the reward $r(s_t, a_t)$ for executing action $a_t \in A(s_t)$. Therefore, surrogates derived from an MDP need to be able to process the state s_t as an input. Figure 5.3 depicts the general process of determining a feasible sequence of actions. Potential surrogates are again highlighted in orange color. Starting from an initial state s_0, the set of feasible actions is $A(s_0)$. Given s_0 and $A(s_0)$, an action $a \in A(s_0)$ can be selected according to some criterion, for instance, with the goal of minimizing the distance to a desired load profile. Then, with the selected action, the subsequent state s_1 can be estimated using $p(s_{t+1}|s_t, a_t)$. For an external entity to be able to execute this process and determine feasible sequences of actions, they must know $A(s_t)$ and $p(s_{t+1}|s_t, a_t)$. Both could be supplied in the form of a surrogate. If the stochastic information provided by $p(s_{t+1}|s_t, a_t)$ is not required, a predictor function for the subsequent state $s_{t+1} = f(s_t, a_t)$ can be provided as a surrogate instead.

In case of a deterministic process, the function $f(s_t, a_t)$ provides perfect predictions. After s_1 has been determined, the procedure can be repeated, using the latest estimate of the state s_t as an input for each subsequent iteration. Once the sequence of actions is sufficiently long, the process is finished. As the selected actions determine how the flexibility provider's DERs are operated, the sequence of actions defines a load schedule. Hence, given the external entity knows $A(s_t)$, $f(s_t, a_t)$, and which action results in which load, they can use this process to determine feasible load schedules. We proposed and tested this concept for modeling and communicating flexibility with surrogates in **[308]**.

In general, the required data for creating both surrogates can either be observed from the system that is to be represented, or be collected from another existing model, such as a simulation model. A combination of both, collecting data and generating artificial samples, is also possible. Overall, it is necessary to collect at least

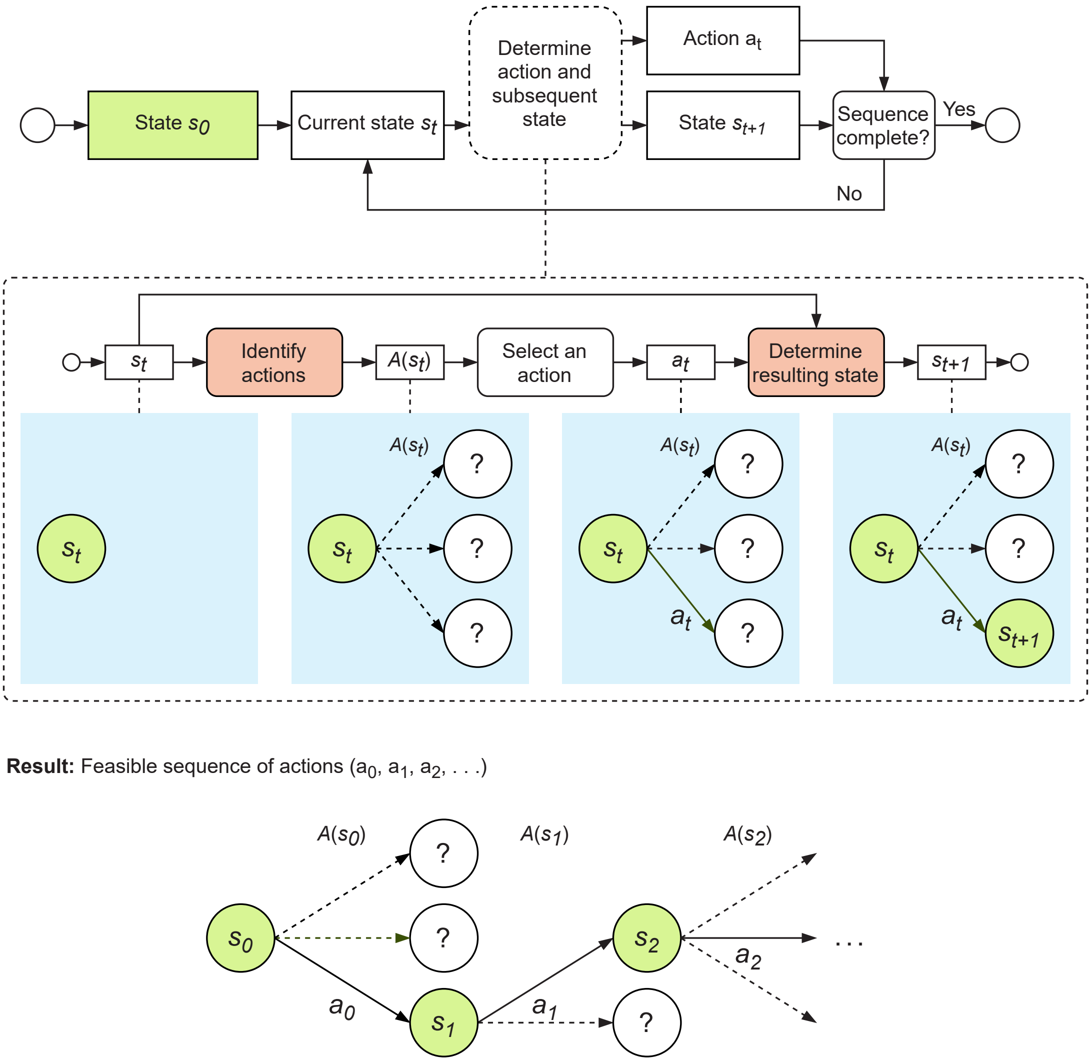

Figure 5.3: Determining a sequence of actions.

consecutive states s_t and s_{t+1}, as well as the action a_t executed in time step t. A surrogate $s_{t+1} = f(s_t, a_t)$ can directly be created from this data. $A(s_t)$, on the other hand, has to be reconstructed using the observed s_t and a_t combinations, derived from the state itself, or collected directly from the devices via some interface. If feasible, the second and third options are to be preferred, as the reconstruction based on finding multiple time steps with identical or similar state is only viable for sufficiently large data sets. Furthermore, even if comparable states are included in the data set, it is likely that the actions executed in similar states are similar as well. Consequently, it is likely that large partitions of the set $A(s_t)$ cannot be reconstructed via this method and the true flexibility cannot be captured by the learned model.

One issue that arises with this MDP surrogate modeling approach is the definition of the actions. There are two general possibilities. Firstly, the actions can directly reproduce the operational choices offered by the individual DERs. Take a BESS which provides control over the (dis-)charging power in steps of 100 W, for instance. Then, the actions 0 ("idling"), 100 ("charge with 100 W"), 200 ("charge with 200 W"), etc. could be defined by the flexibility provider. This works flawlessly, as long as only a single DER is represented. Once a second DER is added, $A(s_t)$ becomes the Cartesian product of the individual devices' actions. Hence, there are multiple actions resulting in the same total load, e.g., "BESS 1 idle and BESS 2 charge with 100 W" and "BESS 1 discharge with 100 W and BESS 2 charge with 200W" produce both a total load of 100 W. One possible solution is to create and communicate an additional mapping from individual action combinations to the associated total load (see **[308]**). Another issue is that the external entity is confronted with more possible choices, making the search for a certain load schedule even more difficult. Since not every combination is equally efficient, for instance, the action with BESS 1 idling is more efficient than the action charging BESS 2 from BESS 1, this could be dealt with by removing inefficient options. The second option for defining actions is to simply split the interval of achievable load levels into n discrete actions. In case the DERs are not able to exactly achieve the defined power value, the closest achievable power level is scheduled. A set of n actions from the power interval $[l, u]$ is given by $\{((u-l) \cdot k/(n-1)) + l | k = 0, 1, \ldots, n-1\}$. Action k results in load $((u-l) \cdot k/(n-1)) + l$, $k = 0, 1, \ldots, n-1$. With this option, there is no need to create an additional mapping for identifying the resulting total power, even when multiple DERs are present. Whenever multiple options are available for achieving a given power, the most efficient one is selected. Since the DERs of the flexibility provider are managed by an EMS, the EMS should be able to decide which controls are favorable. In case there are multiple desirable solutions, the DERs should always be operated the same way in the identical state, in order to guarantee a deterministic state transition. Throughout the remainder of this thesis, it is assumed that actions are specified as real numbers, with the numbers signifying the resulting total load.

The iterative process outlined above and depicted in Figure 5.3 involves the two functions $A(s_t)$ for determining the feasible actions and $f(s_t, a_t)$ for estimating s_{t+1}. Hence, two individual surrogates are required. It is, however, possible to combine both into a single model. There are multiple ways to achieve this. For instance, both could be combined into one single surrogate which approximates $(s_t, a_t) \mapsto (f(s_t, a_t), A(f(s_t, a_t)))$. Another possibility is to generate a surrogate which outputs $(f(s_t, a_t), \tilde{a}_t)$, where $\tilde{a}_t$ is the nearest feasible action in terms of the resulting power at time step t. This output $\tilde{a}_t$ can be compared to a_t, and if the deviation is too large, a_t is infeasible and the returned s_{t+1} invalid. Moreover, as $\tilde{a}_t$ is the closest (predicted) feasible action, it provides a hint for choosing the new a_t and computing $(f(s_t, a_t), \tilde{a}_t)$ again. Only if the deviation is small enough s_{t+1} is accepted. Overall, using such a combined model, only one instead of two ANNs need to be trained and communicated. However, when two ANNs are trained, they can be trained and selected separately, allowing their individual replacement when a better model has been identified.

Repairing load schedules: From a more general perspective, the generation approach inspired by EAs and outlined above is intended for producing a feasible load schedule from some representative input. This input

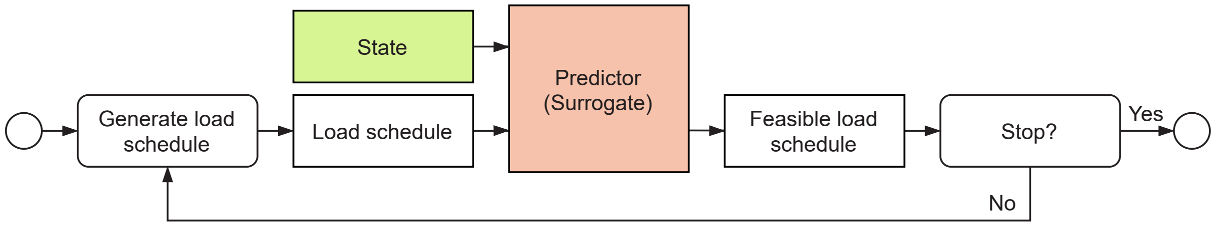

Figure 5.4: Repair of infeasible load schedules **[313]**.

encodes abstract, scheduling relevant information and is mapped to a valid schedule. On a closer look, load schedules themselves are similar in this regard, as they hold scheduling relevant information, too. The major difference is that this scheduling information can be infeasible. However, similar to the translation of instructions encoded in the genotype, a valid schedule may be generated from the potentially invalid input. As such a mechanism transforms infeasible schedules into feasible ones, it is called “repair” in this thesis.

This repair approach is not new, as it is the core of the flexibility exploitation with SVDDs [11]. In the case of the SVDD, there is not even a need to identify pairs of feasible and infeasible schedules for generating the surrogate. Instead, it is sufficient to train the model with valid samples. Using the SVDD, infeasible schedules are projected onto the set of feasible schedules in the feature space. Therefore, inside the feature space, the “repaired” schedule is close to the original one given as input. In **[312]** we tried to identify the closest feasible schedule with the help of MILPs. The infeasible schedules and the resulting feasible schedule were used to train ANNs. However, the surrogates were not able to achieve results which could compete with the other evaluated approaches. We argued that this is likely due to the lack of a clear input and output relationship, as the MILP only provides one “random” solution of many (see **[312]** for more details). Another option could be to follow the load schedule myopically step by step, as long as possible. Doing so, similar to the state-based approach, the possible actions in the current time step t are evaluated and the devices are operated in a way minimizing the deviation from the target during this single time step. Albeit future time steps are neglected and the resulting schedule may not be the solution closest to the input, it is conceivable that this approach produces better results than the use of MILPs, as with this procedure there is a more clear relationship between input and output.

After the flexibility provider has implemented a schedule repair algorithm, they can collect data and train a surrogate. Figure 5.4 illustrates the general exploitation procedure executed by the external entity. After generating a desired load schedule, the corresponding feasible schedule is determined with the help of the model. If this feasible schedule is acceptable, the process can be stopped. Otherwise, it can be repeated until better options have been identified.

Classification of Characteristics

Not only for the direct exploitation, but also the exploitation of abstracted flexibility, similar DERs are often described with a shared model where only parameters need to be updated. Typically, the specific classes are defined manually. Possibilities to automate the determination of a categorization could be derived from clustering techniques. The goal would be to group DERs, either individually or as an aggregate, according to the similarity of their achievable load schedules. This would not only yield groups of similar DERs, but also help to classify other DERs in order to select the best fitting model. The automated generation of such a categorization is not further investigated in this thesis. Given a set of DER classes, shared models can be created. This is usually done by hand. The classification and models are fixed for all entities participating in the flexibility exploitation. If shared models are given, there is no general need for surrogate modeling. Every flexibility provider may simply express their flexibility in terms of the provided classes. Possible reasons for using surrogates instead, including the potential to aggregate devices, have already been pointed out, but since no further input and output combinations can be derived, considering shared models alone does not lead to new ANN-based surrogate modeling approaches. Nonetheless, it hints a way for possibly improving the overall ANN performance with the help of transfer learning. Using this technique, base surrogates for different classes of DERs could benefit any ANN-based surrogate modeling approach.

Interval-Based Models

Modeling the flexibility of DERs by intervals for power, energy, and/or power ramping provides a simple and concise description of their abilities. Even if the intervals vary over time, the number of required parameters only grows linearly. For instance, instead of one interval for the electrical power valid for 24 hours, 96 intervals would suffice to describe a day in 15 minute steps. This is a rather small number of parameters, especially compared to deep ANNs. Hence, training and communicating a surrogate to predict these intervals from a state vector is not advisable. Instead, it is generally more efficient for the EMS to determine and communicate the intervals directly. There are, however, exceptions where surrogate models could be used in a sensible way. Firstly, if uncertainty is modeled, information on the likelihood of certain power, energy, or ramping levels being feasible could be leveraged by the external entity in order to make more robust flexibility choices. One option would be to generalize the interval-based description and distinguish multiple intervals for each value, each associated with a confidence of the values within the interval being feasible. As the number of parameters required in such a description grows rapidly with the level of detail, at a certain point, surrogates become viable again. In this scenario, a surrogate would predict all required intervals from a given state. Secondly, another possible application of surrogates arises when there are time dependent intervals with a causal dependence on past actions, that is, when the subsequent interval is depending on the exact choice within the prior interval. This is essentially a modification of the state-based simulation approach presented above, and demonstrates its versatility.

5.2.2 Exploitation of Abstracted Flexibility and Market-Based Exploitation

In the previous chapter it has been worked out that the central modeling approaches of the market-based exploitation are also valid options for the exploitation of abstracted flexibility (see Figure 4.3). The major difference lies in the existence of a market. This plays only a minor role in regard to surrogate modeling, as it does not matter whether a market platform or some superordinate EMS decides on requests or offers. Hence, both patterns are considered simultaneously in this section. Already discussed approaches and data-driven models, as they are incorporated throughout this section, are omitted here.

Parameterization of a Common Model

The parameterization approach is based on the existence of a shared, versatile model, which can be adapted entirely by setting the correct parameters. As surrogates cannot provide the shared model itself, i.e., the shared model has to be defined manually, there only is the possibility to use surrogates for the estimation of the parameters. However, estimating parameters via surrogates is only a sensible option if the parameters need regular updates or if a huge number of parameters has to be derived from the state. If this is not the case, it is more efficient and precise to simply communicate the parameters instead. Anyhow, it is important to note that in this scenario the flexibility is encoded differently compared to the other approaches. Here, the surrogate is only an instrument for keeping model parameters updated, but the space of feasible schedules is described by the model and not the surrogate. Nevertheless, since the surrogate must encode the operational restrictions and DERs' characteristics to derive the correct parameters, the surrogate must still encode the flexibility.

Requests and Offers

It has been pointed out before that in order to control the operation of DERs, the flexibility providers have to be influenced. Hence, for a surrogate model to be helpful, it must describe the behavior of the flexibility provider given a certain signal. As requests and offers may originate from flexibility providers or external entities, it is necessary to distinguish the respective target. For the sake of brevity, in the following only requests are named, but the same statements are true for offers, too. An external entity as target has the choice of accepting or declining requests. A surrogate could tell the external entity what the consequences of declining a request are. However, this information could easily be supplied as part of the request, if necessary. Nevertheless, when the opposite direction is implemented, that is, the external entity submits offers or requests to the flexibility provider, surrogates may be helpful instruments in multiple ways.

Request outcome and constraint outcome: As explained before, based on the meaning of the individual terms, offers from the external entity are optional and requests are typically compulsory. If an external entity is able to forecast how a flexibility provider reacts to certain requests or offers, they are able to compare different options and select the most promising one to send to the respective flexibility provider. Useful information includes, but is not limited to, the resulting load, the duration the change is maintained, and possible rebound effects. All the named data could be provided by surrogate models generated by each individual flexibility provider and transmitted to the external entity. Surrogates providing this information are applicable for many types of requests, including on/off control and imposing constraints. This approach is, however, not sensible in combination with load schedules, as the schedule itself already holds all this information. As imposed constraints have a less restricting character, the "constraint outcome" approach is named separately, in addition to the "request outcome". With the "response characteristics" $f(t, t_r)$ [138], which describes the change in load for a request at time t with duration t_r, a similar approach has already been presented in the literature review.

Classification: Sets of load schedules play a significant role in the development of concepts for surrogate based flexibility models. Many approaches presented in this chapter could be listed in this context, as they aim to generate feasible load schedules from some input. By simply repeating the proposed generation procedures, a set of schedules likely to be feasible is obtained. An alternative option to compile such a set is to sample random schedules, test whether they are valid or not, and collect the positively classified results. The classification approach introduced and evaluated in **[313]** and **[312]** respectively, takes a whole load schedule as an input and returns the estimated probability of feasibility. The general procedure is depicted in Figure 5.5. It starts out with the generation of one or multiple candidate load schedules. These schedules are then individually passed to the classifier, i.e., the surrogate model, which evaluates whether the schedules are feasible or not. If none of the load schedules is feasible, the process needs to be repeated with new candidates. The procedure either stops once a desired schedule

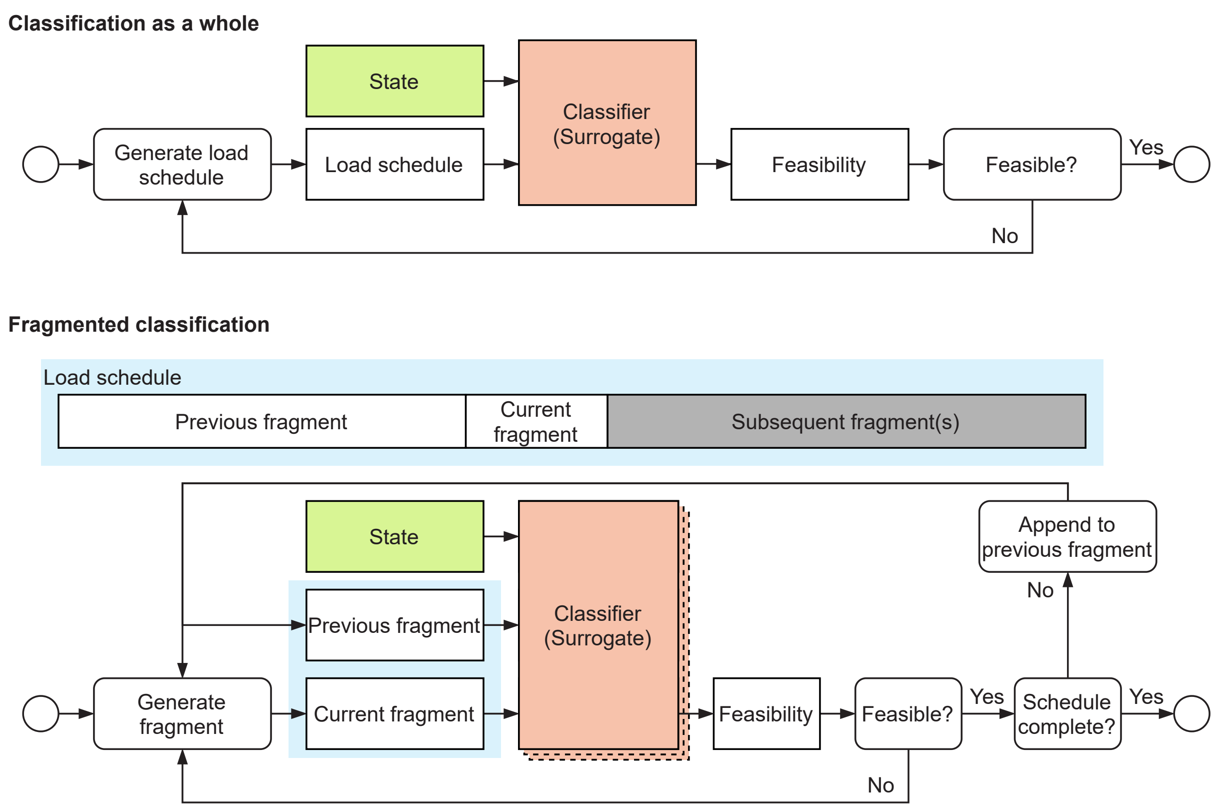

Figure 5.5: The original classification pattern (top) presented in **[313]** and a classifier based schedule generation approach (bottom).

has been classified feasible, or once a certain number of valid schedules has been identified. In the latter case, a subsequent process needs to select the most desirable option. However, simply passing random inputs to the classifier generally only yields very few positive results, if any at all. As usually only a small part of the set of potential load schedules is feasible [62], the generation of candidates is a non-trivial task. An external entity would, therefore, need additional information provided by the flexibility provider in order to confine the search space. A potential option for ANN-based classifiers could be to exploit the gradient information and perform backpropagation steps in order to generate an input vector with a better rating. This idea is picked up again in Section 6.2.

Fragmented classification: Another possible way to tackle the difficulty of finding correct schedules is the modification illustrated in the bottom half of Figure 5.5. Here, the schedule is composed by multiple fragments. With this modification, there is no need to guess entire schedules. Instead, only fragment after fragment needs to be found. Fragments may result from any arbitrary partition of load schedules, i.e., have any arbitrary length. However, single time steps are the simplest option for the external entity to deal with, as only one additional value is required in each step. For this iterative procedure to work, either the surrogate must be capable of processing only certain parts of the input, or there must be multiple surrogates for the different steps involved. Overall, the

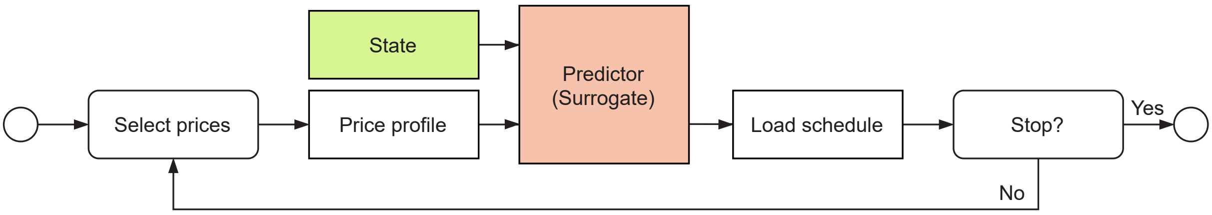

Figure 5.6: Targeted generation of dynamic tariffs **[313]**.

approach is similar to the state-based simulation in regard to the schedule being built step by step in a chronological order. Nevertheless, with this approach, the estimation of states is not required at all. Please also note the similarity to the cascade classifier [12], as it essentially is just a cascade of classifiers.

5.2.3 Indirect Exploitation

Tariff outcome: Models being optional is a feature making the indirect exploitation of flexibility appealing. In its simplest implementation, electricity tariffs reflect the production, consumption, and grid state without considering the specific flexibility providers. Nonetheless, some kind of model is needed, when a more targeted influence is desired or the consequences of pricing decisions need to be estimated. As the literature review shows, often elasticities are used to estimate the change in load resulting from a given change in price. Since elasticities depend on factors like the presence of people, they vary throughout the day. Hence, often multiple elasticities are used. Furthermore, to estimate intertemporal relations, for instance rebound effects, cross-elasticities can be added. All these are usually estimated by the external entity, who needs to exploit the flexibility. Alternatively, these could be estimated by each flexibility providers' EMS on a regular basis and sent to the external entity as a (very rough) model of the DERs. Like explained before, a potential surrogate model must be able to provide estimates of how the associated flexibility provider reacts to a given signal, in this case the prices of the tariff. Analogously to the previously discussed options for estimating model parameters, the elasticities could be estimated and updated with the help of a surrogate. However, here again, it is easier to simply transmit the elasticities instead of using a surrogate. Another option would be to estimate the changes in load directly with the surrogate, instead of using an elasticity. In contrast to the elasticities, which provide linearized estimates, surrogates can provide estimates incorporating nonlinear relations.

Figure 5.6 depicts the process of finding a suitable tariff for an individual flexibility provider **[313, 312]**. It is also applicable if only one single price for a single time step is determined. There may be rules restricting the shape of price profiles (e.g., [5]), for instance requiring a given average. In general, the prices are determined on the basis of the desired change in load. During time steps in which the load should be lowered, prices above the average are usually selected, and vice versa. Utilizing the surrogate, and the state vector provided by the respective flexibility provider, the resulting load schedule can be estimated. If the schedule satisfies the needs of the external entity, the resulting tariff can be communicated back to the flexibility provider. If it is unsuitable, the tariff can be updated until a more suitable expected load schedule is found. As mentioned before, the reversal, that is, the estimation of a tariff from a desired load schedule, may also be a meaningful approach (e.g., [267]).

5.2.4 State Information Based Exploitation

Finally, surrogate modeling for the state information based exploitation remains to be discussed. As the indirect exploitation with non-discriminating tariffs can be seen as stigmergic and, therefore, as part of this pattern, the generalization of the previously presented tariff outcome approach provides one possibility to integrate surrogate models.

Deception outcome: In the tariff approach, the idea is to find the best option by repeatedly adjusting the tariff and predicting the resulting behavior of the flexibility provider. By replacing the tariff in Figure 5.6 with the type of signal that is used in the pattern, e.g., the total load of all flexibility providers, the external entity can evaluate the consequences of sending a certain signal. Furthermore, if such an approach was to be implemented, the reversal of the input and output, that is, the surrogate receives the intended schedule and predicts which signal would result in the schedule, could also be considered. However, in both cases the external entity would need to send artificial signals, which are different for each flexibility provider and do not reflect the true state of the electrical grid. Therefore, the integration of surrogates for the purpose of communicating flexibility changes the associated exploitation pattern.

Both, tariffs and aggregated signals reflecting the state of the grid, are generally determined by some central party. Hence, the approaches outlined here are assuming that there is at least one external entity. In a distributed architecture with peer-to-peer communication, each flexibility provider is an external entity at the same time, as not only flexibility is provided, but also signals are sent to other flexibility providers. In such a case, any of the surrogate modeling approaches derived and introduced in this chapter could be utilized, given the approach fits the intended use and available data.

5.2.5 Summary

With all the general modeling approaches analyzed, it is now possible to summarize the findings. As the discussion shows, there are many opportunities to utilize surrogates for the purpose of encoding flexibility. Whether they are beneficial or the cost outweighs the benefits is use case dependent. Generally speaking, the major benefit of using surrogates is the possibility to generate them in an automated fashion and to communicate them. Therefore, if there is no need for individual models of flexibility providers, or there is already a model, surrogates may not provide much further value. Table 5.1 provides an overview of the most promising surrogate modeling approaches. Approaches with questionable benefits are not listed here. As has been stated before, it is not possible to provide a conclusive list of all imaginable approaches. Further sensible ways of encoding flexibility may exist. Nevertheless, all surrogate modeling approaches found in the literature and referenced in this thesis are covered by the table.

Table 5.1 divides the individual approaches into two distinct categories. The aim of those in the upper section is the generation of schedules. An external entity employing one of these approaches would use the surrogates to generate load schedules likely to be feasible, evaluate the generated options, and then request the flexibility provider to follow the desired schedule. Using such an approach, the external entity is able to exactly select the schedule of each individual flexibility provider. The second type of approaches influences DERs with the help of other types of signals, such as tariffs or constraints. For these, the external entity can only estimate how the flexibility providers will react. The necessary predictions are made with surrogates, which can be generated and provided by the flexibility providers themselves. By varying the inputs and evaluating the outputs of the surrogates, the external

Table 5.1: Summary of the most promising surrogate modeling approaches for encoding and communicating flexibility. Using the upper entries, the flexibility providers receive schedules. For the approaches in the bottom half, the input of the surrogate is sent to the providers.

Purpose	Surrogate	Input*	Output	See
Schedule generation	Generation	Representation	Feasible load schedule	Figure 5.2
	State-based simulation	Action	State, feasible actions	Figure 5.3
	Repair	Load schedule	Feasible load schedule	Figure 5.4
	Fragmented classification	Load schedule fragments	Feasibility	Figure 5.5
	Classification	Load schedule	Feasibility	Figure 5.5
	Cost evaluation	Load schedule	Cost	Figure 5.1
Signal selection	Tariff outcome	Tariff	Load (schedule)	Figure 5.6
	Request outcome	Requested load (change)	Duration (, energy) (, rebound)	Section 5.2.2
	Constraint outcome	Constraint	Duration (, energy) (, rebound)	Section 5.2.2
	Deception outcome	Global state	Load (schedule)	Section 5.2.4

*The current state of the system is generally required as an additional input

entity can select the most promising signal. This signal is then sent to the flexibility providers. For each approach, the table lists the respective inputs and outputs, as well as the intended purpose of the surrogate which details how the overall aim, that is, generating a schedule or identifying a signal, is achieved. As pointed out before, in general, the current state of the represented DERs is required to accurately identify the true flexibility. Therefore, an external entity usually needs to communicate with a flexibility provider before sending requests. Especially for those approaches in the bottom half of the table, it is not always sensible to pursue a surrogate modeling approach. The surrogate is beneficial in situations where a large number of potential flexibility choices needs to be generated. If only a few options are required, it is sufficient to have them generated directly by the flexibility provider instead. In order to generate surrogates according to one of the listed approaches, a sufficient amount of data is required. Which data is needed exactly, evidently varies. From a general perspective, the required data may be collected from a database, observed during operation, generated using simulation models specifically built for this purpose, or be generated and extracted from existing model implementations. Furthermore, dataset augmentation could supply more training data. For more detailed descriptions of the individual approaches, please see the referenced figures and sections.

5.3 Evaluation

The primary goal of this section is to analyze how well ANN-based surrogates are able to encode flexibility and therefore act as models for DERs. In our recent publications **[312, 308]** we already presented a first assessment for some of these surrogate modeling approaches. Taking our previous findings into account, this thesis provides a more structured and reasoned investigation of surrogate modeling, uses a newly developed and more versatile software framework for generating and evaluating models, tests more types and combinations of DERs, and presents a more thorough analysis of the individual results.

Given the previous sections, it is apparent that at least in theory there are many possible ways to control the behavior of DERs with the help of surrogates. Individual surrogate modeling approaches showed to serve one of two primary purposes, that is, the generation of load schedules or the generation of some sort of influential signal. However, due to the high level of abstractness, it is not possible to draw a clear separating line. A surrogate that can be used to explore feasible schedules, can often also be used to derive other types of signals. If, for instance, the goal is to determine feasible constraints and the duration the constraints can be met, the same information can be gathered by searching for feasible schedules within these boundaries. Nevertheless, this does not make the surrogates for signal selection obsolete, as they may need substantially less training data to achieve similar results, since the difficulty of generating load schedules is rather high.

The most versatile approach identified before is the *state-based simulation*. With this surrogate it is possible to replicate all other approaches listed in Table 5.1. Therefore, it can be seen as a baseline of what can be achieved with ANN-based surrogates. For this reason, and since it yielded very promising results in our first tests **[308]**, the *state-based simulation* is used in this evaluation. How each approach could be replicated is outlined in Table 5.2. The training of additional models or development of complementary algorithms may be required though, for this purpose. As an example of how other approaches can be emulated, please consider the *constraint outcome* approach. The *constraint outcome* surrogate predicts the duration a given imposed constraint can be satisfied when in state s_0. Starting from this state s_0 the *state-based simulation* surrogate can be utilized to determine a trajectory of states $(s_0, s_1, \ldots, s_i)$, selecting only feasible actions satisfying the boundary b given by the imposed constraint. Once state s_i is reached, where no feasible choice within the boundary is left, a lower bound for the duration is given by i times the length of a time step. If, for instance, the length of a time step is 15 minutes and there is an $a_0 \in A(s_0) \subset \mathbb{R}$ with $a_0 \leq b$, but no $a_1 \in A(f(s_0, a_0))$ with $a_1 \leq b$, then it has to be expected that the boundary of b watts can be abided by for at least 15 minutes. The number of time steps i and, hence, the bound for the duration, is of course depending on the action choices leading up to the state s_i, which suggests that a variety of trajectories should be tested, when the goal is to emulate the *constraint outcome* surrogate. The general procedure for the assessment is as follows:

1. Definition of the exact goals and evaluation criteria
2. Selection of DER combinations to be tested
3. Specification of the data source for generating the surrogates
4. Generation of the surrogates
5. Conducting tests
6. Compiling results
7. Analysis and discussion of results

As has already been pointed out multiple times throughout this thesis, the specific use case is an important factor to consider when implementing a surrogate based exploitation of flexibility. In the first step, it is therefore necessary to specify the respective goals in order to test and evaluate ANN-based flexibility models. Also, in this step, the specific criteria for assessing each model's performance is derived from these goals. Next, all DERs to be considered in the evaluation are selected. It is then decided how the necessary data for training the ANNs is gathered. Given the data sources, the ANN-based flexibility models are generated, and subsequently tested. With the obtained results it is then possible to give a general answer to which quality of approximation is achievable. The remainder of this section is structured according to this methodology.

Table 5.2: Using the state-based simulation to replicate the other surrogate modeling approaches. While some approaches may be replicated directly, others require additional models or algorithms. However, all approaches can be replicated, which demonstrates the versatility of the state-based simulation approach.

	Surrogate	Replication with a surrogate of the *state-based simulation* approach
Schedule generation	Generation	Generic function using the surrogate to simulate a state trajectory by selecting actions from the set of feasible actions based on some associated generic genotype
	State-based simulation	*
	Repair	Step by step transitions, always choosing the feasible action closest to the respective power value of the time step given by the (potentially infeasible) input schedule
	Fragmented classification	Step by step replication of the schedule by selecting the given power values. Once an infeasible actions is chosen, the classification result is “infeasible”. If all actions are admissible, the schedule is classified “feasible”
	Classification	See *fragmented classification*
	Cost evaluation	Additional cost model $c(s_t, a_t)$ specifying cost for selecting action a_t in state s_t
Signal selection	Tariff outcome	An additional RL policy $\pi(a\|s)$ trained by the flexibility provider is passed to the external entity, who then uses the policy to emulate the optimization conducted by the flexibility provider in order to determine the resulting load schedule
	Request outcome	Testing a given request by selecting the associated actions. Like in the classification, up to the first infeasible action choice, the schedule is feasible
	Constraint outcome	Testing a given constraint by selecting the respective actions, similar to the *request outcome* approach
	Deception outcome	See *tariff outcome*

5.3.1 Goals and Evaluation Criteria

Table 5.3 provides a short summary of the evaluation’s general framework. The goal is to generate a set of feasible load schedules from which the external entity can choose an option. This is the same goal that is pursued in our previous publications **[312, 308]**. Schedules are generated by randomly selecting actions in each single time step. With the state-based simulation approach, for a schedule to be feasible, it is necessary that the ANNs are correctly identifying feasible actions and provide sufficiently accurate forecasts of the resulting states.

The assessment is conducted with DERs found in households and commercial buildings. A detailed specification of the investigated configurations is provided in the next section. To learn the necessary ANNs, it is sufficient to have data listing feasible actions for given states, and state transitions for given actions and states. A control or optimization logic is not required. Therefore, the approach can easily be applied to individual DERs and aggregates of multiple DERs, such as buildings. The data for training the surrogates is acquired with the help of simulation models. It would also be possible to extract the required data from optimization models. However, the implemented optimization models are only used during the evaluation, in order to assess the results in more detail. Finally, the criteria for assessing the performance reflects the respective goal. In the past, we have either used the percentage of

Table 5.3: Evaluation procedure summary.

	State-based simulation
Goal	Generating a set of feasible load schedules of a length up to 24h
DERs	Individual devices and aggregates of DERs
Data sources	Simulation models
Evaluation	Overall feasibility, classification performance, deviation from feasible schedules

feasible load schedules **[308]** or deviation measures **[312]** in order to assess the performance of schedule generation approaches. Both showed to have their individual shortfalls. On the one hand, when generating load schedules, it is desirable to end up with truly feasible schedules, which can be reliably reproduced by the DERs. On the other hand, even minor prediction errors can make a schedule infeasible, even though in practice the DERs could have reproduced it "well enough". Therefore, in contrast to our previous publications, both criteria are used in this thesis. Please keep in mind that in a practical implementation it is possible to make use of responses to the external entity for giving feedback whether the choices are plausible or not.

5.3.2 Considered Distributed Energy Resources

In our recent papers we primarily considered residential buildings, and decided to test a BESS, a CHPP with an HWT, and the combination of both. While, in this thesis, the focus on DERs commonly found in residential buildings remains, additionally an EVSE and more complex aggregates of DERs are tested. All tested DER configurations exhibit different characteristics and operational constraints, which need to be learned by the surrogate model. Please keep in mind that the goal is to encode the flexibility in electricity consumption and production. Therefore, thermal flexibility cannot directly be exploited with the model, but still needs to be learned as it determines the behavior of heat producing DERs. Overall, the BESS is by far the most flexible device, as it is the only considered DER that can switch between consumption and production, or more precisely, charging and discharging electrical energy. The other devices are either able to consume or provide electricity. Furthermore, different DER specific characteristics are considered, which add new constraints. The EVSE, for instance has availability restrictions and varying charging requirements, depending on connected BEV's capacity and SOC. The CHPP, on the other hand, has dwell time requirements to prevent the device from repeatedly starting up and shutting down, resulting in wear and tear.

Battery energy storage system: For parameterizing the BESS model in a stand-alone test, the Tesla Powerwall 2 was chosen as a reference. All relevant parameters are given in Table 5.4. The data sheet of the Powerwall only provides the round-trip efficiency of 0.9 and does not provide information on self-discharge [268]. Splitting the 10% loss equally, charging and discharging efficiencies of 0.95 are assumed, leading to a comparable round-trip efficiency. Regarding self-discharge, in experiments with lithium-ion battery cells, fully charged cells lost around 10% of their energy over a period of 31 days [269]. This breaks down to a loss of 0.34% per day and is hence negligible.

Table 5.4: Parameters of the tested BESS.

BESS	Tesla Powerwall 2		
	Capacity	13.5 kWh	[268]
	Maximum (constant) power (charge and discharge)	5 kW	[268]
	Charging efficiency	0.95	assumption*
	Discharging efficiency	0.95	assumption*
	Standing loss	0	**

*based on round-trip efficiency of 0.9 [268]

**negligible for the investigated 24h periods

Combined heat and power plant satisfying heat demand: The reference device for parameterizing the CHPP model is a first generation Senertec Dachs HKA G 5.5. When running, it provides 5.5 kW of electrical power. In order to add more complexity to the model and make it harder to learn, ramping and minimum dwell times are considered. It is assumed that the power is ramping up and down linearly. The minimum dwell times are chosen randomly and range from 0 to 60 minutes. The CHPP is not tested in isolation, as even with dwell time constraints and ramping, the constraints are rather simple. Instead, the combination of CHPP and HWT, both satisfying a given, varying heat demand, is tested. For the sake of brevity, in the following, the HWT is omitted when referring to this system simply as "CHPP satisfying heat demand" or "CHPP configuration". Table 5.6 lists all parameters. The assumed HWT standing loss has been computed with the formula utilized for determining HWT energy labels given in the EU [270]. Overall the same parameters as in **[312]** are used, but not the same models. Heat demands have been determined with the *CREST Demand Model* [271], assuming a 4 person detached family home, on a week day. A total of 180 time series has been generated, 60 for summer, winter and intermediate seasons, respectively.

Table 5.5: Parameters of the tested CHPP.

CHPP	Senertec Dachs HKA G 5.5		
	Electrical power	5.5 kW	[272]
	Thermal power	12.5 kW	[272]
	Power ramping	linear, 15 min	assumption
	Minimum running time (soft constraint)	0, 15, 30, 45 or 60 min	assumption
	Minimum idle time (soft constraint)	0, 15, 30, 45 or 60 min	assumption

Electric vehicle supply equipment: The parameters for the EVSE are based on the Keba P30, a commercial charging station. It is connected to three phases, each providing 32 A at 230 V, that is, 22.08 kW in total. A minimum charging power of 3.6 kW has been assumed. As the charging equipment is able to supply different BEVs, the model must be able to handle varying charging requirements. In this evaluation, varying capacities, initial SOCs and target SOCs are considered. The target SOC is specified as an interval of allowable values. Minimum and maximum charging powers are assumed to be fixed, independent of the specific EV. Charging

Table 5.6: Parameters of the tested combination of CHPP, HWT, and heat demand.

CHPP satisfying heat demand			
CHPP	Senertec Dachs HKA G 5.5		
HWT	Volume	750 l	assumption
	Temperature range (soft constraint)	60 to 80 °C	assumption
	Standing loss	0.00245 1/h	assumption*
Heat demand	Four person household on a week day		[271]

*energy efficiency class B to C [270] measured at a 45 K temperature difference [273]

requests have been derived randomly based on the battery capacities of a variety of different BEV models. The arrival rate, as well as the allowable ranges for the target SOC, and standing times are assumptions. An arrival rate of 1/48 means that 2 BEVs are expected to arrive during 96 time steps.

Table 5.7: Parameters of the tested EVSE and BEVs.

EVSE	Keba P30		
	Minimum charging power	3.6 kW	assumption
	Maximum charging power	22 kW	32 A on 3 phases
	BEV arrival rate	1/48	assumption
	BEV capacities		
	smart EQ fortwo	17.6 kWh	[274]
	BMW i3	27.2 or 37.9 kWh	[275, 276]
	Seat Mii Electric	36.8 kWh	[277]
	Renault ZOE (R110 and R135)	52 kWh	[278]
	Tesla Model S	70 or 85 kWh	[279]
	Charging parameters		
	Arrival SOC	$\mathcal{U}(0, 0.9)$	assumption
	Target min	$\mathcal{U}(0.3, 0.99)$	assumption
	Target max	$\mathcal{U}(0.7, 1)$	assumption
	Staying time	15 min to 1 day	assumption

Detached family home: The first aggregate of DERs is a combination of a BESS, CHPP, HWT, and heat demand, as listed in Table 5.8. It simply joins together the BESS and the CHPP satisfying the heat demand configurations.

Table 5.8: Parameters of the tested aggregate of BESS, CHPP, HWT, and heat demand.

Detached family home	
BESS	Tesla Powerwall 2
CHPP, HWT, heat demand	Identical to *CHPP satisfying heat demand* (see Table 5.6)

Table 5.9: Parameters of the HoLL model.

FZI House of Living Labs			
3× BESS, each	BAE SECURA PVV Block 6V6 PVV 420		
	Capacity	7.8 kWh	[280]
	Maximum (constant) power	780 W	[280]
	Charging efficiency	0.78	assumption*
	Discharging efficiency	1	assumption*
	Standing loss	0	**
EVSE	Keba P30		
CHPP	Senertec Dachs HKA G 5.5		
GCB	Elco Thision L 100		
	Thermal power	40 kW	***
	Ramp on start up	40 kW/ 450 s	***
	Ramp on shut down	20 kW/ 60 s	***
HWT	Volume	3300 l	
	Temperature range (soft constraint)	40 to 60 °C	
	Standing loss	0.005655 1/h	***
Demand	Historic time series		***

*efficiency of lead-acid batteries reported to be around 0.78 to 0.8 in [281]

**2 % per month [282], negligible for the investigated 24 h periods

***based on measurements in the FZI HoLL

FZI House of Living Labs: The fifth configuration tested in this thesis is a simplified model of the FZI House of Living Labs (HoLL). It is a combination of the above models with adapted parameters and constraints. An overview of the parameters can be found in Table 5.9. The BESS is made up of three identical lead-acid battery systems with a total usable capacity of 23.4 kWh. The maximum total charge and discharge power is 2,340 W. According to the manufacturer the BESS only looses 2% of its charge over the course of a month [282]. Self-discharge is hence negligible. In contrast to the other configurations, the HWT is larger and additionally connected to a GCB. The GCB parameters, as well as the HWT heat loss are derived from measurements collected during normal operation of the devices. Like in the other configurations, inflexible electricity consumption is not considered as it cannot be influenced.

Aggregated battery energy storage system: When moving to higher voltage levels in the electrical grid, the need for aggregation grows. To test the aggregation on a larger scale, a configuration consisting of 100 independent BESSs has also been tested. The simplified HoLL configuration, in contrast, combines different types of DERs and helps to test the ability of ANN-based surrogates to aggregate heterogenous DERs. Half of the 100 BESSs are small storages suitable for single family homes, the other half has enough capacity to act as a shared buffer for a larger number of buildings in a distribution grid. The exact parameters are listed in Table 5.10.

Table 5.10: Parameters for the aggregated BESS model combining a total of 100 individual storages.

Aggregated BESS			
50× BESS, each	Tesla Powerwall 2		
50× BESS, each	Capacity	120 kWh	**[310]**
	Maximum (constant) power (charge and discharge)	120 kW	**[310]**
	Charging efficiency	0.95	assumption
	Discharging efficiency	0.95	assumption
	Standing loss	0	assumption

5.3.3 Data Sources

The versatility of ANNs comes at the cost of needing large amounts of data for training, especially for deep ANNs. As a rule of thumb, for supervised deep learning working with categorical data, 5,000 samples per category suffice to train a model with acceptable performance, and 10 million samples to reach or exceed human performance [27, p. 20]. Given the $4 \cdot 24 \cdot 365 = 35,040$ quarter hours of a year, training surrogate models using observed data samples of 15 minute length seems viable. If, however, time series covering a whole day are required, only 365 non-overlapping samples remain. For those approaches falling under the category *signal selection*, the situation is even more severe when the signal is only sent sporadically. Then only a few samples may remain.

Another potential pitfall of using solely observed data is insufficient variability. As explained before, a trained model can only reflect the flexibility found in the training data. If the training data does not exhibit variability, e.g., due to following a similar schedule each day, the generated surrogate does not encode the true potential of the DERs. While dataset augmentation may alleviate some of these issues, it cannot generally solve them. Therefore, surrogate modeling approaches for encoding flexibility can greatly benefit from existing optimization and simulation models, as they can be used to generate data for training surrogates. For instance, EAs can provide all the data required for surrogates following the generation approach, MDPs are suitable for a state-based simulation surrogate, and any cost based optimization model allows the generation of feasible load schedules needed in the classification and cost evaluation approaches. If such models are not already present, the potential lack of data or data variability can also be tackled with simulation models. By utilizing any of these options, it is possible to generate synthetic data for situations not encountered in the past. This can greatly help to improve the quality of the trained surrogate models, as more regions of the set of possible inputs are covered during the training process. Since simulation models provide the required data more efficiently, as there is no need to interact with a solver software, training data is acquired by means of simulation. Nevertheless, additional optimization models are implemented for the purpose of analyzing the results. Simulation and optimization models for each configuration of DERs are presented in the following. The models are based on our previous publications **[312]** and **[308]**. All of them have been implemented in Python as part of a newly developed framework for generating load schedules and training ANN-based surrogates.

Simulation Models

In the developed framework, all simulation models are derived from a common, abstract class named *Model*, which defines the basic functionality all models must provide. A depiction of this base class can be found in Figure 5.7. For the sake of brevity, only the core functionality of this base class is illustrated. It is designed to provide plentiful information in order to be a suitable data source for most surrogate modeling approaches. The attribute *dt* holds the length of a single time step Δt in seconds, which is needed to compute the energy flows during a single time step. The set of all considered actions is stored in an array named *actions*. Each action is an integer value and equals the resulting power in watts. In the framework, power flowing towards a flexibility provider has a positive sign, while power flowing away has a negative sign. Therefore, the action -1000 translates to "provide 1 kW of power", as the power flows away from the DERs providing it. The set of all considered actions may be a proper superset of the set of actually achievable actions, i.e., it can contain elements which are unachievable for the modeled DER.

In general, an action determines how a single DER or a combination of DERs should operate. If two or more DERs are aggregated, there can be multiple actions associated with the identical total load, depending on the encoding of actions. In order to generate models that are easy to handle for the external entities, such "duplicates" are avoided by encoding actions as the resulting electrical power. Consequently, there can be ambiguous actions that may be achieved in multiple different ways. This is solved by choosing the most profitable option if there is more than one way of providing or consuming the requested power level. In a state where more than one option is optimal, always the same specific option should be selected in order to act less randomly and more predictably. The current state of the simulated DER is stored in the *state* array. As in reality the true state of a device is generally only partially observable, a *hidden_state* array can hold further information unavailable to the learning algorithm generating the surrogate.

Based on the state and hidden state the set of feasible actions can be retrieved with the *feasible_actions()* method, which is implemented as a class property. The set is created by collecting the feasible elements from the actions set, which results in a subset. Finally, the *transition* method allows advancing the simulation to the subsequent time step. Given a feasible action, for instance selected randomly from the set of feasible actions, it computes the subsequent state, hidden state, and the resulting *interaction* with the environment of the DER. The method returns the new state and interaction.

Model
- dt: int - actions: array<int> + state: array<float> + hidden_state: array<object>
+ feasible_actions(): array<int> + transition(action: int, interaction: pair<int, int>): pair<int, int>

Figure 5.7: Base class of all models. The simulation model works exactly like the state-based simulation surrogate.

Figure 5.8 depicts the interplay of actions and interactions for an exemplary ensemble comprising a CHPP, BESS, HWT, and a heat consumer. The interaction tuple initially holds only zeroes. Each model, one by one, receives the interaction tuple from the previous model and manipulates the values according to its behavior. In regard to the sign of the power flow, the resulting interaction is specified from the perspective of the next model in line. While the CHPP and BESS both take an action as an input, the (inflexible) heat consumer and HWT do not. The individual actions for the CHPP and BESS are determined by the local EMS. After calling each transition function, the interaction tuple holds the resulting aggregated electrical and thermal power flows. Using this mechanism, aggregates can be formed by combining multiple simulation models within another simulation model.

In comparison to reality, the interaction of the components is highly simplified. Overall, all implemented simulation models are rather simple and abstract when compared to the actual DER. Nevertheless, compared to most DSM related models found in the literature review, the simulation models utilized in this thesis are still very detailed. In order to generate a load schedule, the selection of an action and the transition to a new state are repeated until the desired amount of steps has been reached. Depending on the surrogate modeling approach and machine learning algorithms it is beneficial to include infeasible samples in the training data. An infeasible load schedule is generated by simply selecting an infeasible action in any of the time steps.

In the following the individual models for the implemented DERs are introduced in terms of the required parameters, as well as the equations describing the state transition. The time step is given by the index t. It assumes values from $t = 0$ up to the considered time horizon T. The length of a simulated time step in seconds is Δt and a fixed value. Amounts of energy are calculated in Ws. All parameters have been set according to the configurations presented in the previous section. The set of feasible actions is derived by inserting the state transition equation into the constraints and checking which actions lead to an acceptable state. As an example, consider the state transition law $s_{t+1} := s_t + a_t$ which depends on action a_t. Furthermore, let the state be restricted to the interval $[0, 100]$. Then the set of feasible actions is $A(s_t) = \{a_t | s_{t+1} \in [0, 100]\} = \{a_t | -s_t \leq a_t \leq 100 - s_t\}$.

Battery energy storage system: The model for the BESS includes the capacity, a maximum charging and discharging power, charging and discharging efficiencies, as well as standing losses. Additionally, artificial minimum and maximum constraints for the SOC are incorporated. These artificial constraints can serve multiple purposes. Firstly, they can act as a buffer for increasing the share of feasible load schedules, which is evaluated in further detail later in this section. Secondly, they could be used by the EMS to restrict the flexibility provided to the external entity. Aside from the restriction posed by the capacity and the minimum and maximum SOCs, it is assumed that there is no dependence between the current amount of stored energy and the different parameters. In

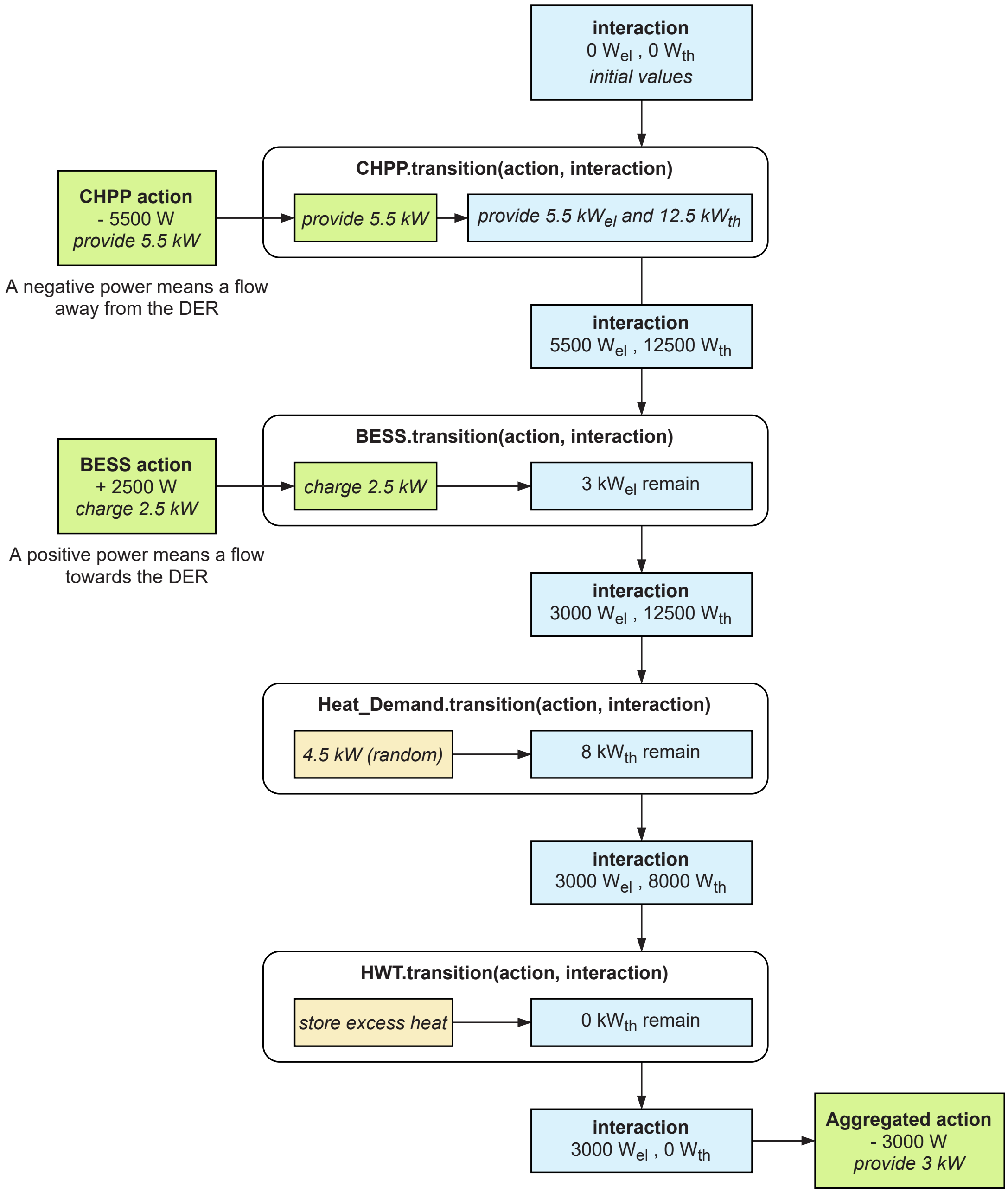

Figure 5.8: Connecting multiple DER simulation models with the help of the interaction tuple provided by the transition function.

other words, the SOC does not influence the maximum and minimum power. Let Q_t^{BESS} be the amount of stored energy at the beginning of time step t. In order to compute the losses from charging and discharging, the direction of the energy flow needs to be considered. When neglecting the standing losses, the change in stored energy is given by

$$\Delta Q_t^{\text{BESS}} := \begin{cases} P_t^{\text{BESS}} \cdot \eta_C^{\text{BESS}} \Delta t & , P_t^{\text{BESS}} \geq 0 \quad \text{(charging)} \\ P_t^{\text{BESS}} \cdot \frac{1}{\eta_D^{\text{BESS}}} \Delta t & , P_t^{\text{BESS}} < 0 \quad \text{(discharging)} \end{cases},$$

with the average power P_t^{BESS}, as well as charging and discharging efficiencies η_C^{BESS} and η_D^{BESS}. In any time step, the BESS may either charge or discharge. The power P_t^{BESS} is limited by the (constant) maximum charging and discharging powers $\overline{P^{\text{BESS}}}$ and $\underline{P^{\text{BESS}}}$, that is, $\underline{P^{\text{BESS}}} \leq P_t^{\text{BESS}} \leq \overline{P^{\text{BESS}}}$. The standing loss is computed with respect to the average amount of stored energy during the considered time step, and using the relative self-discharge per hour η_S^{BESS}. Since the average is computed, the resulting state Q_{t+1}^{BESS} appears on both sides of the equation. Therefore, some minor rearrangements are necessary.

$$\begin{aligned} Q_{t+1}^{\text{BESS}} &= Q_t^{\text{BESS}} + \Delta Q_t^{\text{BESS}} - \frac{Q_t^{\text{BESS}} + Q_{t+1}^{\text{BESS}}}{2} \cdot \frac{\Delta t}{3600} \eta_S^{\text{BESS}} \\ Q_{t+1}^{\text{BESS}} &= Q_t^{\text{BESS}} \cdot \frac{1 - \eta_S^{\text{BESS}} \frac{\Delta t}{2 \cdot 3600}}{1 + \eta_S^{\text{BESS}} \frac{\Delta t}{2 \cdot 3600}} + \Delta Q_t^{\text{BESS}} \cdot \frac{1}{1 + \eta_S^{\text{BESS}} \frac{\Delta t}{2 \cdot 3600}}. \end{aligned}$$

At any time the amount of stored energy should satisfy

$$0 \leq \underline{SOC^{\text{BESS}}} \leq \frac{Q_t^{\text{BESS}}}{\overline{Q^{\text{BESS}}}} \leq \overline{SOC^{\text{BESS}}},$$

where $\overline{Q^{\text{BESS}}}$ is the capacity of the BESS, and $\underline{SOC^{\text{BESS}}}$ and $\overline{SOC^{\text{BESS}}}$ are the minimum and maximum SOC, respectively. If for some reason the SOC is exceeding the minimum or maximum, for instance in the random initial state, the power is restricted in order to push the SOC back inside the bounds. As each action directly corresponds to some power P_t^{BESS}, a feasible action must comply with all restrictions for P_t^{BESS} and Q_t^{BESS}. The interaction, which is determined by the transition method, is simply the average electrical power provided or consumed by the BESS.

Electric vehicle supply equipment: Since the EVSE can only operate in conjunction with a BEV, it is built around a generic BEV model. The BEVs are simplified BESSs of varying capacity that can only be charged. Discharging is neglected here for two reasons. Firstly, there is currently no widespread use of vehicle to grid technology. Secondly, to make the BEV more distinct from the BESS, in order to test more diverse models. In the EVSE model, each BEV is characterized by its capacity and SOC. Given a BEV, a charging process is specified by adding a minimum and maximum target value for the SOC and the remaining standing time. Therefore, at time step t the EVSE state comprises the capacity of the currently connected BEV Q_t^{EVSE}, the associated SOC SOC_t^{EVSE}, the minimum and maximum target SOCs $\underline{SOC_t^{\text{EVSE}}}$ and $\overline{SOC_t^{\text{EVSE}}}$, and the remaining standing time τ_t^{EVSE}. If vehicle specific minimum and maximum charging powers were to be considered, additional state variables would be required. While the capacity and target SOCs remain untouched when transitioning to the next time step, the SOC is updated according to the selected action, using the BESS model with a relative self-discharge of 0.

Self-discharge is not considered, since the BEVs are only connected for several hours each time. The remaining standing time is updated by simply subtracting the length of a time step.

$$\tau_{t+1}^{\mathrm{EVSE}} := \max\{0, \tau_t^{\mathrm{EVSE}} - \Delta t\}.$$

Once the remaining standing time reaches the value 0, the BEV leaves and the EVSE has to wait until a new BEV arrives. Upon the arrival of another BEV, or whenever the parameters need to be adjusted, the state variables are updated, defining the details of the new charging process. The feasibility of actions is determined by the limitations of the vehicle's battery, the remaining standing time, and the constraints for the final SOC. Both target SOC constraints are soft constraints, since there is no plausibility check if it is actually attainable or not. Should the minimum SOC be unreachable during the remaining standing time, charging at maximum power is the only feasible action. If the maximum SOC is exceeded, the EVSE can only idle. An additional minimum charging power, aside from idling with 0 W consumption, can be enforced in the model by restricting the set of possible actions passed to the simulation model.

Arrivals of BEVs are modeled by means of queuing theory, assuming exponentially distributed time intervals between two individual arrivals. The capacity, SOC, as well as the minimum and maximum target SOCs are all chosen randomly. The parameters used in the evaluation can be found in Table 5.7. If a BEV arrives while another one is connected to the EVSE, it is placed in a queue. Once a vehicle leaves, that is, when $\tau_t^{\mathrm{EVSE}} = 0$, the next BEV in the queue starts charging in $t + 1$ and all previous state variables are replaced.

Hot water tank: The model of the HWT is derived from the BESS model, but uses the tank and ambient temperatures instead of the amount of stored energy to describe its state. Given the HWT temperature θ_t^{HWT} and the ambient temperature $\theta_t^{\mathrm{ambient}}$, the stored thermal energy $Q_t^{\mathrm{HWT,H}}$ in Ws at time t is

$$Q_t^{\mathrm{HWT,H}} := (\theta_t^{\mathrm{HWT}} - \theta_t^{\mathrm{ambient}}) V^{\mathrm{HWT}} \rho^{\mathrm{water}} c^{\mathrm{water}},$$

where V^{HWT}, ρ^{water} and c^{water} are the volume of the tank, density of water, and specific heat capacity of water. The state transition equations of the HWT model are identical to those of the BESS, that is,

$$Q_{t+1}^{\mathrm{HWT,H}} = Q_t^{\mathrm{HWT,H}} \cdot \frac{1 - \eta_S^{\mathrm{HWT}} \frac{\Delta t}{2 \cdot 3600}}{1 + \eta_S^{\mathrm{HWT}} \frac{\Delta t}{2 \cdot 3600}} + \Delta Q_t^{\mathrm{HWT,H}} \cdot \frac{1}{1 + \eta_S^{\mathrm{HWT}} \frac{\Delta t}{2 \cdot 3600}},$$

with the change in stored energy while neglecting standing losses

$$\Delta Q_t^{\mathrm{HWT,H}} := \begin{cases} P_t^{\mathrm{HWT,H}} \cdot \eta_C^{\mathrm{HWT}} \Delta t & , P_t^{\mathrm{HWT,H}} \geq 0 \quad \text{(charging)} \\ P_t^{\mathrm{HWT,H}} \cdot \frac{1}{\eta_D^{\mathrm{HWT}}} \Delta t & , P_t^{\mathrm{HWT,H}} < 0 \quad \text{(discharging)} \end{cases}$$

and the total power flow

$$P_t^{\mathrm{HWT,H}} := P_t^{\mathrm{CHPP,H}} + P_t^{\mathrm{GCB,H}} - P_t^{\mathrm{demand,H}},$$

which is depending on the thermal powers $P_t^{\text{CHPP,H}}$ and $P_t^{\text{GCB,H}}$ produced by the CHPP and the GCB, as well as the heat demand $P_t^{\text{demand,H}}$. Since the HWT itself cannot be controlled, the model always returns an empty array of feasible actions and the power $P_t^{\text{HWT,H}}$ is passed to the transition method in the form of the interaction argument. The HWT model does not directly make use of the minimum and maximum SOC constraints. Instead, there are minimum and maximum temperature constraints. While the minimum temperature is the ambient temperature, which is reached once the tank lost all of its stored energy, the maximum temperature may vary. These restrictions are hard constraints, once the HWT reaches one of them, it either cannot provide any more energy or store any more energy. Energy demand or supply exceeding these constraints is passed on to other DERs via the interaction vector. This means, as long as the HWT is able to store or provide energy, the resulting thermal interaction is zero. Therefore, if multiple DERs are combined, the HWT is usually the last model in the chain of DERs to transition to the next time step. Additional soft constraints specifying a corridor of desirable temperatures are introduced in the aggregated models that combine the HWT with other DERs.

Combined heat and power plant: A CHPP produces varying amounts of electricity and heat, depending on its mode of operation. Typically, its control is influenced by the local heat demand. However, in the context of demand side flexibility, the focus lies on the provided electrical power. Therefore, in this thesis "operation mode" refers to a specific set point and amount of produced electrical power. In regard to set points, it is possible to distinguish non-modulating and modulating CHPPs. A non-modulating CHPP is either running or idling, i.e., has $n = 2$ operation modes. Modulating CHPPs can attain various power levels and provide a larger number $n > 2$ of operation modes. In the implemented model, the achieved output power is defined in the form of square matrices, similar to how transition matrices describe Markov chains. Given the index of the current operation mode i, the CHPP provides on average $m_{i,j}$ watts of power during the transition to the mode j. As in general heat and electricity outputs are not identical, they are given by two separate matrices $(m_{i,j}^{\text{CHPP,H}}) \in \mathbb{R}^{n \times n}$ and $(m_{i,j}^{\text{CHPP,E}}) \in \mathbb{R}^{n \times n}$. Take for instance the following two matrices of a non-modulating CHPP, which are used in the experiments:

$$(m_{i,j}^{\text{CHPP,E}}) := \begin{pmatrix} 0 & \frac{-5500}{2} \\ \frac{-5500}{2} & -5500 \end{pmatrix}, \text{ and } (m_{i,j}^{\text{CHPP,H}}) := \begin{pmatrix} 0 & \frac{-12500}{2} \\ \frac{-12500}{2} & -12500 \end{pmatrix}.$$

Please note that the negative signs in both matrices are by modeling convention, as energy flowing towards a device has a positive sign and energy flowing away has a negative sign. While in mode $i = 1$ (first row), the CHPP is not running. It has the options to either remain in this mode throughout the next time step, which results in 0 W output (first column), or to start up and provide $\frac{5500}{2}$ W of electrical power (second column). For the produced heat it is 0 W and $\frac{12500}{2}$ W, respectively. As the example shows, ramping can easily be modeled with the help of these matrices. The implementation assumes that ramping always finishes within a single time step. Even though no modulating CHPPs are evaluated in this thesis, the following description is given for a total of n operation modes.

Aside from different operation modes, the CHPP model considers minimum dwell times. The dwell time is the time the device has been staying in its current operation mode. Here, the CHPP model only incorporates dwell time restrictions for being turned off and on, i.e., producing no power or any power. Assuming the first row $i = 1$ reflects the "off" mode, and given the current dwell time τ_t^{CHPP} and minimum running time $\underline{\tau^{\text{CHPP,on}}}$ in seconds, a CHPP can perform the following actions:

$$A^{\text{CHPP}}(i, \tau^{\text{CHPP}}) := \begin{cases} \text{Actions leading to “off” mode} & \text{, if “off” and minimum “off” time not reached yet} \\ \text{Actions leading to “on” modes} & \text{, if “on” and minimum “on” time not reached yet} \\ \text{Actions leading to any mode} & \text{, else} \end{cases}$$

$$= \begin{cases} \{m_{i,j}^{\text{CHPP,E}} \mid j = 1\} & \text{,if } i = 1 \text{ and } \tau^{\text{CHPP}} < \underline{\tau^{\text{CHPP,off}}} \\ \{m_{i,j}^{\text{CHPP,E}} \mid j = 2, 3, \ldots, n\} & \text{,if } i > 1 \text{ and } \tau^{\text{CHPP}} < \underline{\tau^{\text{CHPP,on}}} \\ \{m_{i,j}^{\text{CHPP,E}} \mid j = 1, 2, \ldots, n\} & \text{, else} \end{cases} .$$

Both dwell time constraints are soft constraints intended to reduce wear and tear from frequent mode switching. In order to ensure that there is always at least one feasible action in any situation, these constraints can be ignored by setting a flag. This flag is used in the aggregated models. Each time the CHPP transitions from a running to the stopped state or vice versa, the dwell time τ^{CHPP} is reset to Δt, since at the beginning of the next simulation step Δt seconds will have passed. With the current mode i and the subsequent mode j, it is therefore

$$\tau_{t+1}^{\text{CHPP}}(i, j) := \begin{cases} \tau_t^{\text{CHPP}} + \Delta t & \text{, if remaining “off” or “on”} \\ \Delta t & \text{, if switching between “off” and “on”} \end{cases}$$

$$= \begin{cases} \tau_t^{\text{CHPP}} + \Delta t & \text{, if } (i = 1 \wedge j = 1) \vee (i > 1 \wedge j > 1) \\ \Delta t & \text{, if } (i = 1 \wedge j > 1) \vee (i > 1 \wedge j = 1) \end{cases}$$

The state of the CHPP is given by its current mode i, dwell time τ_t^{CHPP}, and minimum dwell times $\underline{\tau^{\text{CHPP,on}}}$ and $\underline{\tau^{\text{CHPP,off}}}$. By making the minimum dwell time part of the state vector, it is possible to adapt the constraints without changing the model and therefore without relearning the surrogate.

Gas condensing boiler: In contrast to the other DERs, which are implemented as stand-alone models, the GCB has directly been integrated into an HWT model, since the GCB only provides thermal flexibility. The GCB is modeled the same way as the CHPP, but without constraints for the dwell time. Its state is therefore completely described by its current mode of operation i, which is needed to select the correct output power from the given $(n^{\text{GCB}} \times n^{\text{GCB}})$-matrix $(m_{i,j}^{\text{GCB}})$. The associated HWT uses exactly the model described above. Therefore, the overall state is given by the mode of the GCB, the temperature of the HWT and the ambient temperature. Since the GCB does not provide electrical flexibility, the set of feasible actions is always empty for this device. Instead, it operates on its own, following a hysteresis control. Every time the simulation reaches the lower temperature constraint, the GCB turns on, and every time the upper temperature constraint is reached it shuts down.

Electricity and heat demand: The implemented demand model uses a set of reference time series to determine the consumed power. In this model, the state variable is the power consumed during the next time period. Therefore, whenever a state transition occurs, this value is simply replaced with the help of the reference time series. All reference time series must have the same length and temporal resolution. Whenever a simulation is started, one reference time series is chosen randomly. The identifier of this series is stored as a hidden state variable, for the

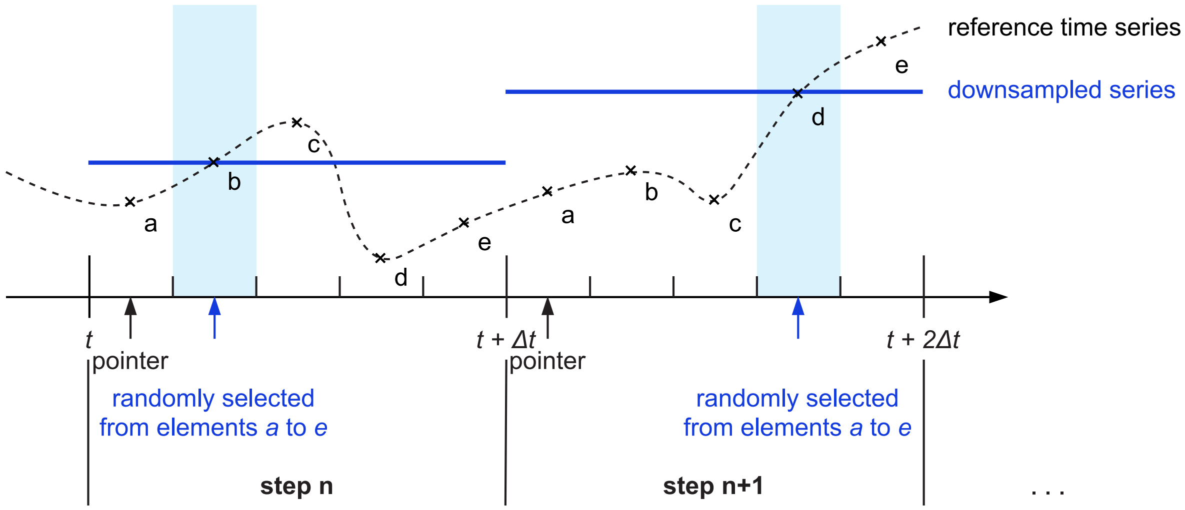

Figure 5.9: Downsampling of the reference time series by randomly selecting values.

purpose of selecting only values from this specific series. Furthermore, the hidden state includes a position marker, pointing to the next time series element to consider. This pointer is shifted with every state transition to reflect the passage of Δt seconds. In case the time series' temporal resolution is higher than Δt, that is, less than Δt seconds lie between two entries of the time series, the pointer moves more than one element in each time step and a downsampling mechanism is applied.

The implemented downsampling process is depicted in Figure 5.9. In the illustrated example, the black, dashed reference time series holds a value for each minute and the time step length Δt is 5 minutes. This means that the pointer to the current element in the reference time series (black arrow) is shifted forward 5 values with each time step. The blue lines are the downsampled time series. It is constructed by repeatedly drawing one of the elements between the current and the subsequent pointer position, and then moving the pointer. Here, the potential values in each step are labeled *a* to *e*. This downsampling method is used with the goal of generating more diverse demand series, compared to simply computing the average.

The demand model aggregates all inflexible demand. Therefore, there is no possibility to influence the demand with any action. The interaction returned by the transition method is equal to the negative state variable, as the direction is inverted from the perspective of other DERs.

Combined heat and power plant satisfying heat demand: This model combines the CHPP, HWT, and heat demand models presented above into a new one and adds further operational constraints. At any time the heat demand must be satisfied, either directly by the CHPP or by using the thermal energy stored in the HWT. As mentioned before, there are additional temperature soft constraints added to keep the temperature in a desirable region. In this combined model, the CHPP is forced to start or stop its production of heat (and electricity) once the minimum or maximum temperature is reached, regardless of its current dwell time. If for instance, the minimum temperature has been reached, the only feasible action for the CHPP is to switch to the mode producing the most energy. In such a case, the dwell time constraint is ignored by setting the respective temporary flag provided by the CHPP model. Nonetheless, as long as the temperature is acceptable, the CHPP's dwell time constraints must not be violated. By operating the CHPP in this manner, the heat demand can always be satisfied, given the CHPP and HWT are reasonably sized. Moreover, there is at least one feasible action in any state. Hence, the "dead-end" issue discussed in **[308]** is completely avoided and the model is more realistic. The power flow towards or away from the HWT is given by

$$P_t^{\text{HWT,H}} := P_t^{\text{CHPP,H}} - P_t^{\text{demand,H}},$$

which is passed to the HWT's transition method as an interaction.

Detached family home: The detached family home is one of the three investigated systems combining multiple DERs that provide electrical flexibility. It joins the BESS, CHPP, HWT, and heat demand models into a single aggregated model. The CHPP, HWT, and demand models are integrated the exact same way as described above, using soft constraints for the HWT temperature and allowing to force the CHPP into certain modes when the temperature bounds are violated. Thereby, they form one of two sub-systems. The second sub-system is the BESS, which can operate independently of the other DERs. The set of feasible actions is determined by computing the feasible actions of the two individual sub-systems and determining each possible combination of actions. As actions are encoded in terms of the associated total power, there can be combinations of CHPP and BESS actions resulting in the same total value. Ambiguous actions are resolved by considering the temperature of the thermal storage. Assuming standard German consumer prices for electricity and gas, it is generally only beneficial to use the CHPP when the produced heat is utilized. If the HWT temperature is already high, it is cheaper to simply draw electricity from the grid and use the HWT to satisfy the heat demand. Therefore, when there are multiple ways of achieving the requested total power, the CHPP is turned on when the temperature is in the lower half of the intended temperature range, and it is turned off if the temperature is in the upper half. The remaining difference is covered by the BESS.

FZI House of Living Labs: With the combination of a BESS, EVSE, CHPP, HWT and a GCB, the HoLL model is the most diverse one investigated in this thesis. While the BESS and EVSE can operate independently, the CHPP and GCB are both sharing the same HWT which stores excess energy for later consumption. In contrast to the detached family home, in this configuration, the CHPP is never forced to start, since the GCB is able to handle the heat demand on its own. Nevertheless, the CHPP is still forced to shut down once the upper temperature boundary is exceeded. In the real HoLL the minimum dwell time of the CHPP is 10 minutes for both modes. While it would be possible to simply set 15 minutes in the simulation model, the minimum dwell times are varied throughout the experiments, in order to make the task more challenging. If there are multiple ways to achieve a requested action, the model makes a short-sighted choice based on an economic rational. Similar to the detached family home, it is only beneficial to use the CHPP when the generated heat is utilized. Therefore, it is activated or stopped if the HWT temperature is in the lower or upper half of the intended temperature range. After this choice, another degree of freedom may still remain. If the BEV can be charged without discharging the BESS, the ambiguity is eliminated by prioritizing the charging of the BEV over the charging of the BESS. In case the BESS must be discharged to achieve the requested action, the BEV is charged as slowly as possible, if at all. This simple strategy tries to avoid losses from transferring energy between batteries.

Aggregated battery energy storage system: The aggregated BESS combines 100 individual battery storages, each using the BESS model outlined before. Feasible actions are determined by simply adding the admissible power intervals. If one BESS could provide $[-5, 5]$ kW and a second one could provide $[-5, 3]$ kW, the admissible power for their aggregate is $[-10, 8]$ kW. Every predefined discrete possible action lying within this interval is feasible. Each of these BESSs has its own SOC, which means the aggregated maximum and minimum power varies over time, as some storages may be nearly empty while others are almost full. When an action is requested, the specified power needs to be disaggregated and distributed to the individual storages. It is assumed that the aggregated power is split proportionately to the power each storage is able to provide on its own. If the maximum charging power of a single storage i is $\overline{P_t^i}$ at time t and the requested, aggregated power is $P_{total} > 0$, then the resulting charging power of this BESS is $P_t^i = \overline{P_t^i} \cdot \frac{P_{total}}{\sum_j \overline{P_t^j}}$. Discharging is handled analogously.

Artificially Relaxed and Tightened Constraints

In our most recent publication on the topic of surrogate modeling **[308]** we recognized that it is important to consider inaccurate state estimations when formulating or selecting constraints. Since, in general, ANNs do not provide perfectly accurate predictions, the approach may fail to produce feasible load schedules, even with well performing ANN-based surrogates. This issue is further amplified with each single computed time step, as the prediction errors add up. Please consider the CHPP satisfying a heat demand as it is described above, for instance. The joint model keeps the HWT temperature within an acceptable temperature range by overriding the CHPP's dwell time constraints if necessary. Now, let the actual HWT temperature be 80.1° C, while the forecast of the ANN is 79.9° C. In this situation, the ANN-based model may decide to keep the CHPP running, even though it must turn off in reality. Conversely, the surrogate would try to stop the CHPP if its prediction was 80.1° C, even though the real temperature is 79.9 °C and it is not allowed to do so yet. This example shows that even the smallest error can result in the infeasibility of generated load schedules. The issue is not limited to ANNs, but also applies for all other types of models that can only approximate and not exactly reproduce the reference model or real world. In preliminary trials the same state-based simulation approach has been tested with finite state machines instead of ANNs, where the exact same problem arose from rounding errors. Therefore, it is necessary to incorporate buffers **[308]** or select less hard constraints.

In this thesis, this challenge has been tackled by artificially tightening and relaxing some selected constraints during the training and evaluation. It is important to point out that it may not always be possible to manipulate constraints in such a way. If not, the state variables associated with the respective constraint must be forecasted with very high accuracy. By using tighter constraints, it is possible to lead the surrogate to act before the actual constraint is reached, or to stay away from the boundary at all. In the example above, the surrogate could be forced to deactivate the CHPP when only approaching 80° C. Therefore, even when the predicted temperature is too low, the CHPP will correctly be deactivated. Then again, this may cause the CHPP to be deactivated too early, even though the minimum running time has not yet been reached. As a consequence, it is necessary to relax the dwell time constraint within this newly created border region, allowing to force the CHPP to turn off, but not enforcing turning it off. However, this relaxation must not be taught to the surrogate model, as this would simply shift the original problem to the newly created boundary of the relaxed constraint. Combining both, the tightened constraints for training the surrogate and the relaxed constraints for evaluating with the reference model, overlapping regions are created, in which the effects of prediction errors are absorbed.

There are two possible ways to add manipulated constraints to the surrogate itself. Firstly, they can be directly taught to the surrogate. Secondly, they may be passed as an additional state variable. The latter approach has the advantage of being more flexible, as the constraints, and thereby the size of the buffer, can be adapted dynamically. However, this comes at the cost of adding more variables that need to be estimated. Since the goal of this chapter is

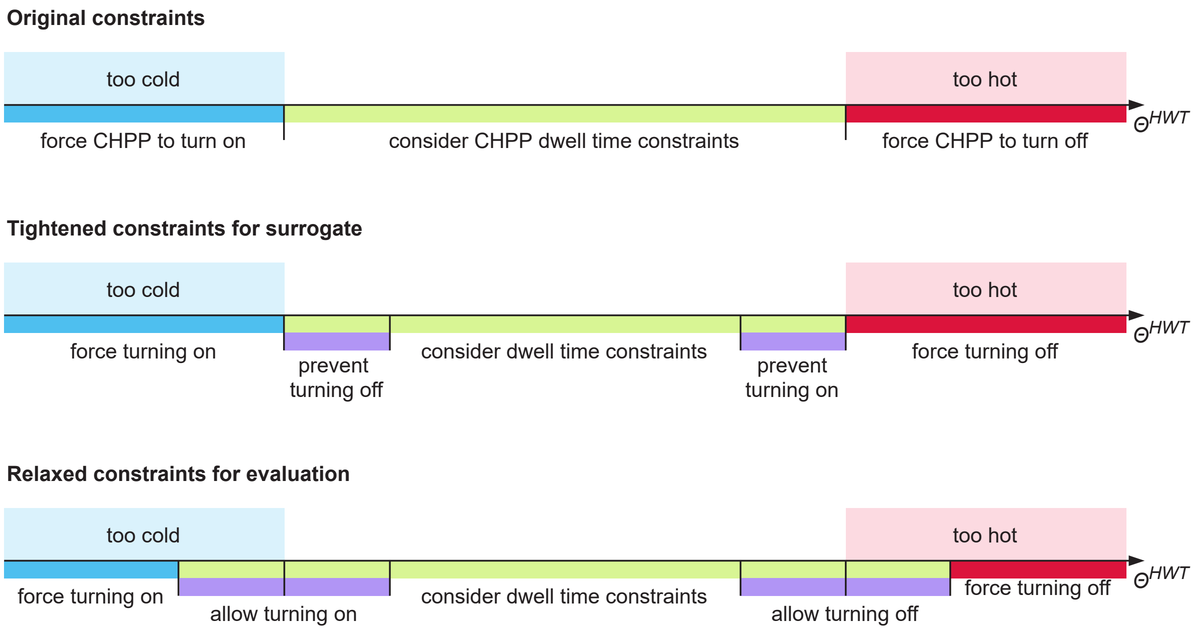

Figure 5.10: Adaptations made to the constraints of the CHPP satisfying heat demand configuration.

to evaluate which quality of approximation is achievable with ANN-based surrogates, mostly the latter approach is used, to be able to vary the buffer size as needed. This is also the reason why the BESS model's state has a minimum and maximum SOC variable. For the CHPP satisfying a heat demand, the constraints have been adapted as depicted in Figure 5.10. In order to understand the meaning of each adaptation, it is necessary to consider two cases. Firstly, the (soft) temperature limit being approached from outside the interval of acceptable temperatures and, secondly, the limit being approached from inside. As the temperature may be over- or underestimated, a violation of the limit is either detected too late or too early. During training, the adapted constraints teach the surrogate to not activate or deactivate the CHPP if it is close to the respective temperature boundary, even though the dwell time would allow it. This adaptation is relevant for the first case, when the temperature approaches the limit from outside. By adding this constraint, resuming the normal operation too early is prevented. Consider a temperature exceeding the upper (soft) constraint, for instance. Due to the violation, the CHPP must shut down and eventually the temperature begins to fall again. Without the additional constraint, if the temperature is underestimated, the CHPP may be started too early. Starting too late, on the other hand, is acceptable, since the temperature of the HWT is high. The considerations for the lower boundary are analogous. During the evaluation, two relaxations are applied to the constraints in order to deal with prediction errors while approaching the limit from inside the boundaries. Firstly, if the surrogate underestimates the progression of the temperature towards one of the temperature constraints, the CHPP mode may be adapted too late, as the violation occurs earlier. To deal with this issue, the CHPP is only forced to change its operation mode when the original boundary plus a given margin is exceeded. This effectively shifts the original boundary outwards, widening the range of acceptable temperatures. Secondly, to deal with an overestimated progression and too early actions, mode changes within an area close to the boundary are permitted. In the case of the HoLL model, only the adaptations for the upper temperature boundary are applied, since there is no forced activation on low temperatures.

Optimization Models

For each simulated device, the framework also includes a mixed integer linear model that can be used in MILPs. A MILP is formed by adding an objective function. Using different objective functions, it is possible to determine schedules for various goals, e.g., minimizing cost or following a target schedule. The parameters for the individual device models can be extracted directly from the simulation models. With this setup it is possible to run a simulation and then directly generate a MILP from the simulation model to validate the simulation output. For evaluating the feasibility of a load schedule $(P_t)_{t=1,2,\ldots,T}$, the objective is to minimize the total deviation from the target schedule $(P_t^{\text{target}})_{t=1,2,\ldots,T}$, that is,

$$\min \sum_{t=1,\ldots,T} dP_t^{\text{abs}}$$

s.t.

$$|P_t^{\text{target}} - P_t| \leq dP_t^{\text{abs}}, t = 1, \ldots, T.$$

The aggregated power P_t at time t is the sum of the consumed or provided powers of the individual DERs. For instance, when only the BESS is evaluated, the constraint $P_t = P_t^{\text{BESS}}$ is added alongside the BESS model. In the following the mixed integer linear models of the individual simulation models are presented.

Battery energy storage system: The optimization model of the BESS is mostly identical to the simulation model. In order to distinguish charging from discharging, which is necessary for computing charging and discharging losses separately, the power flow P_t^{BESS} is split into two parts, P_t^{BESS+} for charging and P_t^{BESS-} for discharging. This modification adds a degree of freedom to the model that would allow the optimizer to dissipate electricity by charging and discharging at the same time. To prevent this from happening, the binary variables $b_t^{\text{BESS,charge}}$ limit the BESS to either charge or discharge during each time step $t = 1, 2, \ldots, T$. All remaining constraints and equations are directly taken from the simulation model. The initial amount of stored energy is passed to the model via Q_0^{BESS}.

$$\begin{array}{lclcll}
P_t^{\text{BESS}} & = & P_t^{BESS+} - P_t^{BESS-} & & & ,t = 1, \ldots, T \\
0 & \leq & P_t^{BESS+} & \leq & \overline{P^{\text{BESS}}} \cdot b_t^{\text{BESS,charge}} & ,t = 1, \ldots, T \\
0 & \leq & P_t^{BESS-} & \leq & |\underline{P^{\text{BESS}}}| \cdot (1 - b_t^{\text{BESS,charge}}) & ,t = 1, \ldots, T \\
\Delta Q_t^{\text{BESS}} & = & P_t^{BESS+} \cdot \eta_C^{\text{BESS}} \Delta t - P_t^{BESS-} \cdot \frac{1}{\eta_D^{\text{BESS}}} \Delta t & & & ,t = 1, \ldots, T \\
Q_t^{\text{BESS}} & = & Q_{t-1}^{\text{BESS}} \cdot \frac{1-\eta_S^{\text{BESS}\,\frac{\Delta t}{2\cdot 3600}}}{1+\eta_S^{\text{BESS}\,\frac{\Delta t}{2\cdot 3600}}} + \Delta Q_t^{\text{BESS}} \cdot \frac{1}{1+\eta_S^{\text{BESS}\,\frac{\Delta t}{2\cdot 3600}}} & & & ,t = 1, \ldots, T \\
\underline{SOC^{\text{BESS}}} & \leq & \frac{Q_t^{\text{BESS}}}{\overline{Q^{\text{BESS}}}} & \leq & \overline{SOC^{\text{BESS}}} & ,t = 1, \ldots, T \\
b_t^{\text{BESS,charge}} & \in & \{0, 1\} & & & ,t = 1, \ldots, T
\end{array}$$

Electric vehicle supply equipment: In contrast to the EVSE simulation model, which makes use of SOCs to describe its state, the optimization model is built around amounts of stored energy. Let Q_t^{EVSE} be the current amount of stored energy in the BEV connected to the EVSE and P_t^{EVSE} the power used by the EVSE. Since the EVSE can be configured with a minimum charging power $\underline{P_t^{\mathrm{EVSE}}}$, binary variables $b_t^{\mathrm{EVSE,charge}}$ are required to identify the time steps a BEV is charged. The maximum charging power is $\overline{P^{\mathrm{EVSE}}}$. Assuming perfect predictions, all charging parameters generated by the simulation model in the given time horizon are passed to the optimization model. Aside from the initial amounts of stored energy and the time varying target range $[\underline{Q_t^{\mathrm{EVSE}}}, \overline{Q_t^{\mathrm{EVSE}}}]$ of each charging process, the planned remaining standing time at each time step $\hat{\tau}_t^{\mathrm{EVSE}}$ is passed to the optimization model. Given $\hat{\tau}_t^{\mathrm{EVSE}}$ and the maximum charging power $\overline{P^{\mathrm{EVSE}}}$, $\hat{\tau}_t^{\mathrm{EVSE}} \cdot \overline{P^{\mathrm{EVSE}}}$ is the maximum amount of energy the EVSE can provide until the BEV leaves. Therefore, at any time step t of a charging process, $\max\{0, \underline{Q_t^{\mathrm{EVSE}}} - \hat{\tau}_t^{\mathrm{EVSE}} \cdot \overline{P^{\mathrm{EVSE}}}\}$ is the minimum amount of energy the BEV must have stored in order to reach $\underline{Q_t^{\mathrm{EVSE}}}$ in time and, hence,

$$\max\{0, \underline{Q_t^{\mathrm{EVSE}}} - \hat{\tau}_t^{\mathrm{EVSE}} \cdot \overline{P^{\mathrm{EVSE}}}\} \quad \leq \quad Q_t^{\mathrm{EVSE}} \quad \leq \quad \overline{Q_t^{\mathrm{EVSE}}} \qquad , t = 1, \ldots, T \ .$$

With the help of $\hat{\tau}_t^{\mathrm{EVSE}}$ it is further possible to check if the current charging process has finished or a new one is starting. Once the car leaves $\hat{\tau}_t^{\mathrm{EVSE}}$ equals zero and, hence, $\mathbb{1}_{\{0\}}(\hat{\tau}_t^{\mathrm{EVSE}}) = 1$. If instead $\mathbb{1}_{\mathbb{R}_{>0}}(\hat{\tau}_t^{\mathrm{EVSE}}) = 1$, the charging process is still in progress. It is therefore

$$\mathbb{1}_{\mathbb{R}_{>0}}(\hat{\tau}_t^{\mathrm{EVSE}}) \cdot \underline{P_t^{\mathrm{EVSE}}} b_t^{\mathrm{EVSE,charge}} \quad \leq \quad P_t^{\mathrm{EVSE}} \quad \leq \quad \mathbb{1}_{\mathbb{R}_{>0}}(\hat{\tau}_t^{\mathrm{EVSE}}) \cdot \overline{P^{\mathrm{EVSE}}} b_t^{\mathrm{EVSE,charge}} \qquad , t = 1, \ldots, T \ .$$

As long as $\mathbb{1}_{\mathbb{R}_{>0}}(\hat{\tau}_{t-1}^{\mathrm{EVSE}} - \hat{\tau}_t^{\mathrm{EVSE}}) = 1$ the standing time still decreases and the current charging process is in progress. Once $\hat{\tau}_t^{\mathrm{EVSE}}$ reaches 0 or is set to a higher value in order to start a new charging process, we have $\mathbb{1}_{\mathbb{R}_{>0}}(\hat{\tau}_{t-1}^{\mathrm{EVSE}} - \hat{\tau}_t^{\mathrm{EVSE}}) = 0$. Using this, it is possible to reset the amount of stored energy Q_t^{EVSE} for each new charging process. Let $\hat{Q}_{t-1}^{\mathrm{EVSE}}$ be the expected amount of energy at the beginning of each planned charging process in step $t-1$. Then we have

$$\begin{aligned} Q_t^{\mathrm{EVSE}} \quad = \quad & Q_{t-1}^{\mathrm{EVSE}} \cdot \mathbb{1}_{\mathbb{R}_{>0}}(\hat{\tau}_{t-1}^{\mathrm{EVSE}} - \hat{\tau}_t^{\mathrm{EVSE}}) \\ + \quad & \hat{Q}_{t-1}^{\mathrm{EVSE}} \cdot (1 - \mathbb{1}_{\mathbb{R}_{>0}}(\hat{\tau}_{t-1}^{\mathrm{EVSE}} - \hat{\tau}_t^{\mathrm{EVSE}})) \\ + \quad & P_t^{\mathrm{EVSE}} \cdot \eta_C^{\mathrm{EVSE}} \Delta t \qquad , t = 1, \ldots, T. \end{aligned}$$

All indicator functions $\mathbb{1}_{\mathbb{A}}(x)$ are resolved during the initialization of the model, which is possible since all forecasts $\hat{\tau}_t^{\mathrm{EVSE}}$ are known in advance.

Hot water tank: This model is similar to the BESS. The only differences are the added water temperature θ_t^{HWT} and the temperature bounds in place of the SOC bounds.

$$
\begin{array}{llllll}
\theta_t^{\text{HWT}} & = & \frac{Q_t^{\text{HWT,H}}}{V^{\text{HWT}}\rho^{\text{water}}c^{\text{water}}} + \theta^{\text{ambient}} & & & ,t=1,\dots,T \\
P_t^{\text{HWT,H}} & = & P_t^{\text{HWT},H+} - P_t^{\text{HWT},H-} & & & ,t=1,\dots,T \\
0 & \leq & P_t^{\text{HWT},H+} & \leq & \overline{P^{\text{HWT}}} \cdot b_t^{\text{HWT,charge}} & ,t=1,\dots,T \\
0 & \leq & P_t^{\text{HWT},H-} & \leq & |\underline{P^{\text{HWT}}}| \cdot (1 - b_t^{\text{HWT,charge}}) & ,t=1,\dots,T \\
\Delta Q_t^{\text{HWT,H}} & = & P_t^{\text{HWT},H+} \cdot \eta_C^{\text{HWT}} \Delta t - P_t^{\text{HWT},H-} \cdot \frac{1}{\eta_D^{\text{HWT}}} \Delta t & & & ,t=1,\dots,T \\
Q_t^{\text{HWT,H}} & = & Q_{t-1}^{\text{HWT,H}} \cdot \frac{1-\eta_S^{\text{HWT}} \frac{\Delta t}{2\cdot 3600}}{1+\eta_S^{\text{HWT}} \frac{\Delta t}{2\cdot 3600}} + \Delta Q_t^{\text{HWT,H}} \cdot \frac{1}{1+\eta_S^{\text{HWT}} \frac{\Delta t}{2\cdot 3600}} & & & ,t=1,\dots,T \\
\theta^{\text{ambient}} & \leq & \theta_t^{\text{HWT}} & \leq & \overline{\theta^{\text{HWT}}} & ,t=1,\dots,T \\
b_t^{\text{HWT,charge}} & \in & \{0,1\} & & & ,t=1,\dots,T
\end{array}
$$

Since, in all conducted experiments, the largest possible flow of thermal energy is well below any physical limitations of the parameterized heat storages, the maximum and minimum power $\overline{P_{\text{HWT}}}$ and $\underline{P_{\text{HWT}}}$ have both been set to a very large number $\mathbf{M}$. In this and all subsequent models $\mathbf{M}$ is a placeholder for any number large enough not to limit the model in its capabilities.

Combined heat and power plant: The linear mixed integer model of the CHPP makes use of binary variables for tracking its operation mode. At the beginning of each time step t, the current operation mode i may be changed to mode j, given none of the constraints is violated. The state transition is encoded with the help of the variables $b_{t,i,j}^{\text{CHPP}} \in \{0,1\}$. If, for instance, at the start of the first time step $t = 1$ the CHPP is in operation mode $i = 1$ and switches to mode $j = 0$, we have $b_{1,1,0}^{\text{CHPP}} = 1$. As for any time step only a single transition must occur, the sum of all $b_{t,i,j}^{\text{CHPP}}$ for any given t must equal 1. Additionally, operation mode i can only be left for mode j, if mode i has been selected in the previous time step. During time step t the CHPP provides P_t^{CHPP} and $P_t^{\text{CHPP,H}}$ watts of electrical and thermal power. Like in the simulation model, the power produced in each mode is given by the matrices $(m_{i,j}^{\text{CHPP,E}}) \in \mathbb{R}^{n\times n}$ and $(m_{i,j}^{\text{CHPP,H}}) \in \mathbb{R}^{n\times n}$. This leads to the constraints

$$
\begin{array}{llll}
\sum_i \sum_j b_{t,i,j}^{\text{CHPP}} & = & 1 & ,t=1,\dots,T \\
\sum_i b_{t-1,i,j}^{\text{CHPP}} & = & \sum_k b_{t,j,k}^{\text{CHPP}} & ,t=1,\dots,T \text{ and } j=1,\dots,n \\
P_t^{\text{CHPP}} & = & \sum_i \sum_j b_{t,i,j}^{\text{CHPP}} \cdot m_{i,j}^{\text{CHPP,E}} & ,t=1,\dots,T \\
P_t^{\text{CHPP,H}} & = & \sum_i \sum_j b_{t,i,j}^{\text{CHPP}} \cdot m_{i,j}^{\text{CHPP,H}} & ,t=1,\dots,T
\end{array} .
$$

In order to simplify the tracking of the minimum dwell times for running and idling, the binary variables $b_t^{\text{CHPP,off}}$ are introduced. It is assumed that the mode $i = 1$ equals the “off” state. Any mode $i > 1$ is therefore an “on” state. Whether the CHPP is turned on or off during time step t, can therefore be determined with

$$
b_t^{\text{CHPP,off}} = \sum_k b_{t,k,1}^{\text{CHPP}} \quad ,t=1,\dots,T.
$$

The initial values $b_{0,i,j}^{\text{CHPP}}$, $b_0^{\text{CHPP,off}}$, the initial dwell time τ^{CHPP}, and the minimum dwell times in seconds are all parameterized according to the simulation model. For this purpose, all dwell times are converted from seconds to the respective number of time steps, that is,

$$\begin{array}{lll} \underline{\tilde{\tau}}^{\text{CHPP,off}} & := & \left\lceil \frac{\underline{\tau}^{\text{CHPP,off}}}{\Delta t} \right\rceil \\ \tilde{\tau}^{\text{CHPP}} & := & \left\lfloor \frac{\tau_0^{\text{CHPP}}}{\Delta t} \right\rfloor \end{array} .$$

As the dwell time constraints are soft constraints, the two binary variables $b_t^{\text{CHPP,forceOn}}$ and $b_t^{\text{CHPP,forceOff}}$ are introduced for forcing the CHPP into starting or stopping its operation, completely ignoring the dwell time.

$$\begin{array}{lll} b_t^{\text{CHPP,off}} & \leq & (1 - b_t^{\text{CHPP,forceOn}}) \quad , t = 1, \ldots, T \\ b_t^{\text{CHPP,off}} & \geq & b_t^{\text{CHPP,forceOff}} \qquad\qquad , t = 1, \ldots, T \end{array}$$

The previously discussed relaxation of constraints is achieved with the help of the binary variables $b_t^{\text{CHPP,allowOn}}$ and $b_t^{\text{CHPP,allowOff}}$. These variables allow the CHPP to ignore the dwell time constraints for the "off" and "on" state, respectively, but do not force it to change its operation mode. In the following, only the minimum "off" time constraint is presented, as both constraints, "on" and "off", are analogous. For each time step

$$\Delta\tau_t^{\text{CHPP,off}} := (b_t^{\text{CHPP,allowOn}} + b_t^{\text{CHPP,forceOn}}) \cdot \mathbf{M} \quad , t = 1, \ldots, T$$

holds either the value 0 or a very large number exceeding the minimum dwell time. By adding $\Delta\tau_t^{\text{CHPP,off}}$ to the current dwell time, the minimum dwell time constraint is automatically satisfied whenever $b_t^{\text{CHPP,allowOn}}$ or $b_t^{\text{CHPP,forceOn}}$ equals 1. In general the minimum dwell time is enforced by ensuring that the CHPP has been idling during the previous $\underline{\tilde{\tau}}^{\text{CHPP,off}}$ time steps. Whether this is the case or not can be detected with the help of $b_t^{\text{CHPP,off}}$. The device has been out of operation sufficiently long if $b_i^{\text{CHPP,off}} = 1$ for all $i = t - \underline{\tilde{\tau}}^{\text{CHPP,off}}, \ldots, t-1$. Mode changes are detected by computing $(-b_t^{\text{CHPP,off}} + b_{t-1}^{\text{CHPP,off}})$. If the mode remains unchanged, this expression assumes the value 0, if the CHPP is turned off it equals -1, and if it is turned on the value is 1. Therefore, whenever $(-b_t^{\text{CHPP,off}} + b_{t-1}^{\text{CHPP,off}}) = 1$, i.e., the CHPP is starting up in t, the sum of the associated $b_t^{\text{CHPP,off}}$ must equal $\underline{\tilde{\tau}}^{\text{CHPP,off}}$. Incorporating the initial dwell time $\tilde{\tau}^{\text{CHPP}}$ leads to

$$\begin{aligned} &\forall t \geq \max\{2, \underline{\tilde{\tau}}^{\text{CHPP,off}} - \tilde{\tau}^{\text{CHPP}}\} : \\ &\qquad \Delta t^{\text{CHPP,off}} + \sum_{i=\max\{1, t-\underline{\tilde{\tau}}^{\text{CHPP,off}}\}}^{t-1} b_i^{\text{CHPP,off}} \geq \min\{t-1, \underline{\tilde{\tau}}^{\text{CHPP,off}}\} \cdot (-b_t^{\text{CHPP,off}} + b_{t-1}^{\text{CHPP,off}}). \end{aligned}$$

The constraint only applies for $t \geq 2$, since at $t = 1$ there are only two possible situations of which neither requires this constraint. Firstly, in case $\underline{\tilde{\tau}}^{\text{CHPP,off}} - \tilde{\tau}^{\text{CHPP}} = 1$, the CHPP must idle one more time step, which is checked at $t = 2$. Secondly, if $\underline{\tilde{\tau}}^{\text{CHPP,off}} - \tilde{\tau}^{\text{CHPP}} < 1$, the CHPP has already been idling long enough, making this constraint superfluous. As mentioned before, $\Delta t^{\text{CHPP,off}}$ is added so that the soft constraint can be ignored if necessary. For $t < \underline{\tilde{\tau}^{\text{CHPP,off}}} - \tilde{\tau}^{\text{CHPP}}$, the CHPP must remain "off", as long as it is not forced to start up. Therefore,

$$b_t^{\text{CHPP,off}} \geq b_{t-1}^{\text{CHPP,off}} - \Delta t^{\text{CHPP,off}} \quad , t = 1, \ldots, \underline{\tilde{\tau}^{\text{CHPP,off}}} - \tilde{\tau}^{\text{CHPP}} - 1.$$

Please keep in mind that $t \in \mathbb{N}$.

Gas condensing boiler: The GCB optimization model uses the same state transition formulation as the CHPP model, but is limited to producing thermal power. Hence, we have

$$\begin{array}{lcll} \sum\limits_i \sum\limits_j b_{t,i,j}^{\text{GCB}} & = & 1 & , t = 1, \ldots, T \\ \sum\limits_i b_{t-1,i,j}^{\text{GCB}} & = & \sum\limits_k b_{t,j,k}^{\text{GCB}} & , t = 1, \ldots, T \text{ and } j = 1, \ldots, n^{\text{GCB}} \\ P_t^{\text{GCB,H}} & = & \sum\limits_i \sum\limits_j b_{t,i,j}^{\text{GCB}} \cdot m_{i,j}^{\text{GCB}} & , t = 1, \ldots, T \end{array} .$$

Further constraints are required in order to model the hysteresis control which is used to keep the HWT temperature in an acceptable range. Whether the device is running or not is detected by computing

$$b_t^{\text{GCB,off}} = \sum_k b_{t,k,1}^{\text{GCB}} \quad , t = 1, \ldots, T.$$

Additionally, the binary variables $b_t^{\text{GCB,start}}$ and $b_t^{\text{GCB,stop}}$ are used to signify the need to start up or shut down the device. The GCB must start to produce heat when the temperature of the HWT falls below the lower bound $\theta^{\text{HWT},min}$, and is only allowed to stop once the temperature exceeds the upper bound $\theta^{\text{HWT},max}$. It must then remain deactivated until the cycle starts again. The two constraints

$$\begin{array}{lcll} \theta^{\text{HWT},min} - \theta_{t-1}^{\text{HWT}} & \leq & \mathbf{M} \cdot b_t^{\text{GCB,start}} & , t = 1, \ldots, T \\ -\theta^{\text{HWT},min} + \theta_{t-1}^{\text{HWT}} & \leq & \mathbf{M} \cdot (1 - b_t^{\text{GCB,start}}) & , t = 1, \ldots, T \end{array}$$

force $b_t^{\text{GCB,start}}$ to attain the value 1 if and only if $\theta^{\text{HWT},min} > \theta_{t-1}^{\text{HWT}}$. The upper bound is modeled correspondingly. Given that $b_t^{\text{GCB,start}}$ and $b_t^{\text{GCB,stop}}$ only equal 1 if the respective boundary is exceeded, the hysteresis control is achieved with the constraints

$$\begin{array}{lcll} b_t^{\text{GCB,off}} & \geq & b_t^{\text{GCB,stop}} & , t = 1, \ldots, T \\ (1 - b_t^{\text{GCB,off}}) & \geq & b_t^{\text{GCB,start}} & , t = 1, \ldots, T \\ b_t^{\text{GCB,off}} & \geq & b_{t-1}^{\text{GCB,off}} - b_t^{\text{GCB,start}} & , t = 1, \ldots, T \\ (1 - b_t^{\text{GCB,off}}) & \geq & (1 - b_{t-1}^{\text{GCB,off}}) - b_t^{\text{GCB,stop}} & , t = 1, \ldots, T \end{array} .$$

While the first two constraints force the GCB to start or stop, the second pair of restrictions forces it to remain on or off.

Demand: Time series for electricity and heat demand are created by the simulation model described earlier in this section. The generated series $(P_t^{\text{demand,H}})_{t=1,2,\dots}$ is simply passed to the MILP as a parameter.

Combined heat and power plant satisfying heat demand: This model combines the CHPP, HWT and heat demand models presented above. The power received or provided by the HWT is

$$P_t^{\text{HWT,H}} = -(P_t^{\text{CHPP,H}} + P_t^{\text{demand,H}}) \quad , t = 1, \dots, T$$

Additional constraints are introduced to make use of the CHPP's ability to ignore the dwell time constraints in order to keep the HWT temperature within an acceptable region. This region is bounded by soft constraints. In the following, only the constraints for the lower bound are presented, as the complementary constraints for the upper bound are analogous. Let $\theta^{\text{HWT},min}$ be the lowest acceptable temperature, then

$$\theta^{\text{HWT},min} - \theta_{t-1}^{\text{HWT}} \leq \mathbf{M} \cdot b_t^{\text{CHPP,forceOn}} \quad , t = 1, \dots, T$$

forces the CHPP to start whenever the temperature of the HWT falls below $\theta^{\text{HWT},min}$. The constraint can be relaxed by incorporating a small buffer. Let this buffer be $\Delta\theta$ kelvin. Then we have

$$(\theta^{\text{HWT},min} - \Delta\theta) - \theta_{t-1}^{\text{HWT}} \leq \mathbf{M} \cdot b_t^{\text{CHPP,forceOn}} \quad , t = 1, \dots, T.$$

When using relaxed constraints, close to the temperature boundary, the CHPP is allowed to be activated regardless of its dwell time. This is achieved with

$$(\theta^{\text{HWT},min} + \Delta\theta) - \theta_{t-1}^{\text{HWT}} \leq \mathbf{M} \cdot b_t^{\text{CHPP,allowOn}} \quad , t = 1, \dots, T.$$

However, as long as the temperate is above $\theta^{\text{HWT},min} + \Delta\theta$, the binary variable $b_t^{\text{CHPP,allowOn}}$ must be equal to 0, which requires

$$-(\theta^{\text{HWT},min} + \Delta\theta) + \theta_{t-1}^{\text{HWT}} \leq \mathbf{M} \cdot (1 - b_t^{\text{CHPP,allowOn}}) \quad , t = 1, \dots, T.$$

Finally, the constraint

$$b_t^{\text{CHPP,forceOn}} \leq b_t^{\text{CHPP,allowOn}} \quad , t = 1, \dots, T$$

prevents the CHPP from being forced to change its mode, when it is not allowed to.

Detached family home: This configuration simply combines the model of the BESS with the "CHPP satisfying heat demand" model described in the previous paragraph. The total power is

$$P_t = P_t^{\mathrm{BESS}} + P_t^{\mathrm{CHPP}}.$$

FZI House of Living Labs: The mixed integer linear model of the HoLL combines the BESS, EVSE, CHPP, HWT, and GCB model. As pointed out before, the CHPP is only forced to shut down, but never forced to start up. Hence, the associated constraints on the lower energy bound found in the configurations combining a CHPP and HWT are not applied here. A heat demand series is again generated by the simulation model and passed to this model as a parameter. The total power drawn from or fed into the grid is

$$P_t = P_t^{\mathrm{BESS}} + P_t^{\mathrm{EVSE}} + P_t^{\mathrm{CHPP}}.$$

Aggregated battery energy storage system: The aggregated BESS MILP is formed by combining 100 individual BESS models to one large optimization problem. The total power is

$$P_t = \sum_{i=1}^{100} P_t^{\mathrm{BESS},i},$$

where $P_t^{\mathrm{BESS},i}$ is the power contributed by the i-th BESS. In contrast to the simulation, where a requested aggregated power level is split proportionally, the MILP is free to choose any combination of P_t^i values adding up to the correct sum.

5.3.4 Surrogate Generation

Surrogate models are trained by the framework in an automated process, using datasets generated with the help of the different simulation models. For each individual surrogate modeling approach different sets of data are needed. It is therefore necessary to implement specific sample generators, collecting and structuring the required data. However, this only needs to be done once, as all relevant variables and methods are standardized in the framework, using class-based inheritance. For this thesis a sample generator for the state-based simulation approach has been implemented. Tensors of a fixed size are utilized to store the training and validation datasets. Before the training process begins, the datasets are initialized, filling them with newly generated samples. If desired, the samples can be replaced during training, allowing repeated renewals of the dataset. All data is generated in dedicated processes, leveraging the processing power of multi-core processors. Training and validation batches are drawn randomly. The datasets are shared between processes, allowing the parallelized training of ANNs, either on CPU or GPU using PyTorch and CUDA.

While in our previous work, the ANN topology and hyperparameters were specified manually **[312, 308]**, in this thesis, they are all randomly selected. Motivated by neural architecture search outperforming manual architecture engineering for different tasks [283], the ANN topology is generated randomly. Alongside other hyperparameters, for instance the learning rate, learning rate decay, and regularization term, parameters describing the ANN structure are drawn from a predefined set of possible options. All generated ANNs are feedforward neural networks. Figure 5.11 shows the topology specific parameters and illustrates their meaning by showing exemplary values and

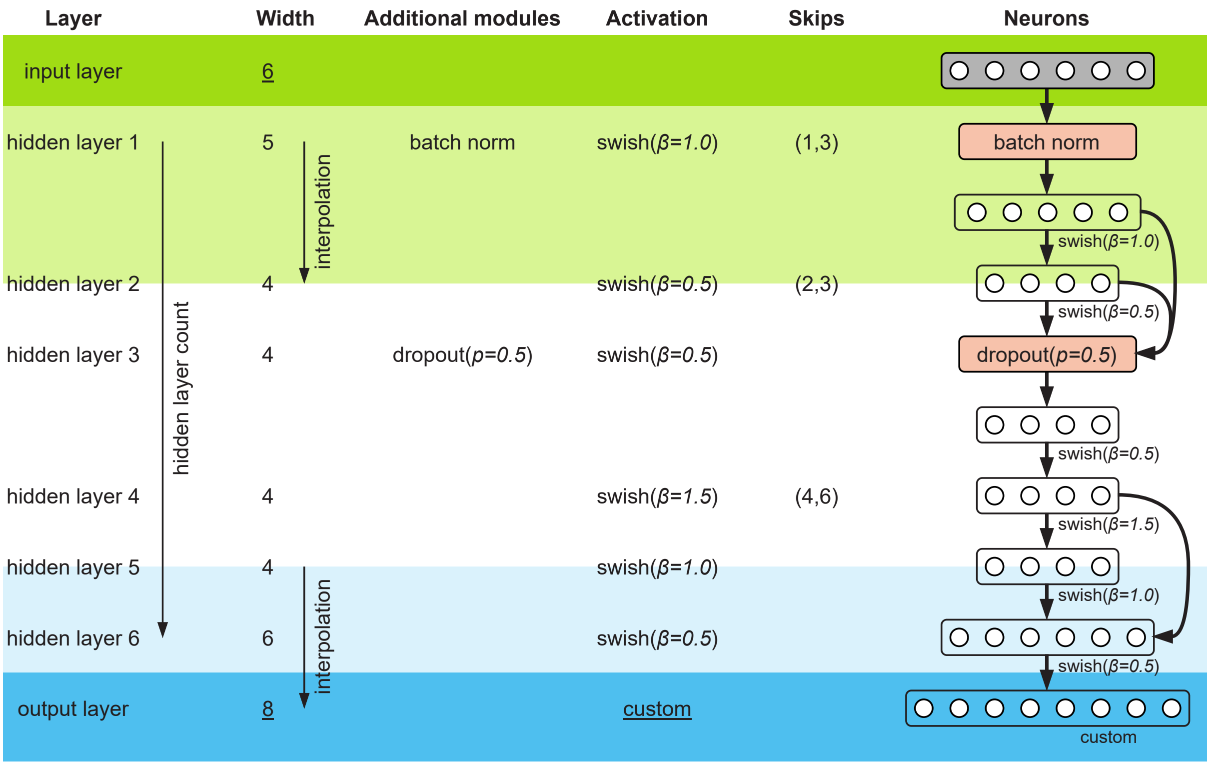

Figure 5.11: Exemplary parameters for generating an ANN alongside the resulting fully connected ANN. Additional modules like batch norm or dropout layers are placed in front of the linear layers. A swish activation function is added after every affine linear layer $y = xW^\intercal + b$, except for the output layer, where a custom activation layer can be applied. The underlined parameters are fixed.

the resulting ANN. The depth of the ANN is determined by the number of hidden layers. A width parameter specifies the width of the hidden layers. Widths for input and output layers are fixed parameters. Two additional parameters specify the number of layers the width should be interpolated. They allow the linear interpolation of widths between the input layer, hidden layers, and output layer. In Figure 5.11, it is one layer on either end. Hidden layers, generated according to the hidden layer count, are affine linear layers, i.e., compute $y = xW^\intercal + b$ (see also Figure 2.3), but may be preceded by additional modules. The framework implements the possibility to randomly add batch normalization and dropout layers, which can both help to train better performing ANNs [27]. Each hidden linear layer is activated by a swish activation function $f(x) = x \cdot \sigma(\beta x)$ [30] with a random beta. A custom activation function, including a mixture of different activations for different neurons, can be specified for the output layer. Furthermore, random skip connections are added to the ANN. Skip connections skip one or multiple ANN layers, creating shorter paths through the ANN [27]. With their help, layers receive larger gradients [27, 284], singularities of the loss function's Hessian matrix are eliminated [284], and the loss is smoothed around the minima [285], overall facilitating the generation of better results. Skips are constructed by creating a random, binary, upper triangular matrix. The rows and columns of all elements with value 1 specify a skip connection. In Figure 5.11 only the resulting skips are depicted.

With all these sources of randomness in addition to the randomly initialized ANN weights, it is apparent that chance plays a non-negligible role in finding well performing models. Therefore, multiple different ANNs are trained, each with its own random topology and hyperparameters. After each training epoch, the ANNs' performance is evaluated using the validation dataset and a performance score is computed from the validation losses. In **[308]** we rated the learned models based on the highest validation loss observed during an epoch. The idea was to select the model exhibiting the best worst-case performance. However, more recent experiments showed that the distribution of losses can vary widely, and therefore the maximum loss is only a weak indicator for how well the model performs. Exemplary distributions of losses can be found in the results section. To consider the whole distribution of losses, the 0-, 0.25-, 0.5-, 0.75- and 1.0-quantiles are determined for each epoch. The epochs are then rated by computing the average of these quantiles. Given the averages, the best ANN and epoch are selected by searching the lowest score, completing the training process.

The state-based simulation surrogates require two separate ANNs, a classifier and one predicting state transitions. All state-based surrogate models are trained from samples generated with the respective simulation models by repeatedly drawing a random state, determining the feasible actions, selecting an action randomly, and computing the subsequent state. In order to train the classifier ANN it is sufficient to repeat only the first two steps. The classifiers are all trained using the binary cross entropy loss function. For the state transition estimators, the loss function varies. While it is of course possible to compute the losses of all state variables with a single loss function, it is often desirable to adapt the variable encoding and loss function in order to achieve better results. Take for example the MSE loss, which is utilized for predicting continuous variables. Using this loss function to predict one of multiple discrete options requires the manual rounding of the results, as we did in **[312]**. In such a case, it is usually beneficial to use a one-hot encoding and apply a cross entropy loss instead. Since the state vector in general comprises different types of variables, some of them discrete, some of them continuous, it is sensible to specify the loss function as a mixture of multiple different losses. With this mixed loss function, a suitable loss function for each element of the state vector can be selected. Additional weights can be used to prioritize individual elements. When applying a higher weight to a single element, the associated prediction error makes up a larger portion of the overall loss. Therefore, it is more important to focus on a precise prediction of this element. To simplify the handling of element specific encodings and loss functions, the framework implements easily configurable pre- and post-processing functionality. This includes the normalization, discretization, and one-hot encoding of single elements, as well as the reversal. Furthermore, it is possible to process not only individual vectors, but also entire batches at once.

Even though it is not required, the state transition model is additionally trained to predict the interaction resulting from the state transition. This information was used for additional testing and further analysis during the development process. As long as only feasible actions are passed to the simulation model, the returned electrical interaction should always be the additive inverse of the action. In other words, if the sum of action and interaction does not equal zero, the system is not able to draw or provide the exact amount of power specified by the action. When an infeasible action is requested, the simulation models execute the closest feasible action. Hence, given the ANN is able to predict the correct interaction, the interaction output can be used as an indicator of feasibility, or to identify which action was performed during the previous time step. Utilizing this information, it is theoretically possible to do completely without the classifier and only use a single ANN. However, in experiments the separated classifier helped to improve the results, and it was therefore not made use of the interaction prediction.

5.3.5 Evaluation Process

The quality of approximation provided by the state-based surrogate models is evaluated in up to three different steps:

In the first step, the performance of the classifier is assessed. For this purpose, a random test dataset is generated, utilizing the same sample generators employed during the training process. With the dataset, the shares of true and false positives, as well as true and false negatives are determined. Here, using the state-based simulation surrogate, positive refers to actions being classified as feasible. This isolated test is significant for two reasons. Firstly, the shares of false positives and false negatives are rough indicators for the expected model performance. Assuming the algorithm generating the load schedules does not systematically favor actions that tend to be false positives, $\frac{FP}{FP+TP}$ can be seen as a rough estimate of the chance of selecting an infeasible action. However, since during load schedule generation imprecise state forecasts are used, this estimate is more akin to a lower bound for the chance of selecting an infeasible action. The false negative rate, on the other hand, indicates how well the surrogate is able to describe the true flexibility. A higher false negative rate means less possible choices, even though in reality more options are viable. Secondly, when only load schedules with a length of one time step are required, that is, when requesting an immediate and short-sighted change in behavior, only the classifier's ability to identify feasible actions is relevant. The classifier ANN returns values from the interval $[0, 1]$. In the experiments, any action with a rating of 0.5 or higher is considered feasible. While it would be possible to change this threshold in order to tune the false positive and false negative rates, the threshold remains fixed.

In the second step, sets of load schedules are generated and synchronously evaluated with the help of the associated simulation model. The load schedule generation is implemented as proposed earlier in Figure 5.3. In each time step a set of potentially feasible actions is identified with the help of the ratings returned by the classifier. Actions are chosen randomly from this set, employing two different strategies. The first strategy chooses any action completely randomly, which is how we performed our first evaluation of state-based surrogates for flexibility **[308]**. Doing so, the state is the only influence of past actions on the next choice, as it determines which actions are valid. For DERs with many feasible actions, like the BESS, this produces load schedules alternating between positive and negative, and high and low power flows. In a practical use case, presumably less volatile schedules are desired by the external entity, maintaining a reduced or increased demand for a longer period of time. The second strategy is intended to produce such load schedules. It divides the overall set of possible actions, which is not to be confused with the set of feasible actions in a given state, into multiple smaller sets by dividing the range between the maximum and minimum achievable power into intervals. Take for instance a BESS able to provide -5 kW to 5 kW. Dividing this range into 4 equally sized pieces yields the intervals $[-5, -2.5]$, $[-2.5, 0]$, $[0, 2.5]$, and $[2.5, 5]$. These intervals are used to limit the action selection step during load schedule generation. A series of intervals is created by drawing them randomly with replacement and repeating each element a predefined number of times, in order to create longer periods with comparable power flows. For example, the drawing procedure could lead to the sequence $([-5, -2.5], [-2.5, 0])$. Repeating each element 4 times results in $([-5, -2.5], [-5, -2.5], [-5, -2.5], [-5, -2.5], [-2.5, 0], [-2.5, 0], [-2.5, 0], [-2.5, 0])$. Now, during schedule generation, the selection algorithm tries to choose feasible actions from within the interval associated with the current time step, that is, in the first 4 steps a power in $[-5, -2.5]$. Should there be no action rated feasible within the desired interval, the interval is expanded including the direct neighbors until all options are depleted. In case no feasible actions can be identified, both strategies resort to selecting the action with the highest ratings, as there should always be at least one valid action. While generating each load schedule, a simulation is performed synchronously. By tracking the outputs of the surrogate and simulation model, a dataset allowing very detailed analyses is created. Furthermore, the feasibility can directly be detected with the simulation model.

In the third step, an additional evaluation with the help of a MILP is conducted. If aggregates of multiple DERs are evaluated, the simulation model alone may not be able to identify all feasible schedules. Please remember that in this thesis the set of actions does contain each possible aggregated power only once, since duplicates are eliminated by always operating the individual DERs the most beneficial way. Due to errors in the predicted state, the surrogate may expect the DERs to operate the wrong way, while still achieving the correct power. Even though the power in this time step is correct, this can lead to the states of the surrogate and simulation model diverging. Take for instance a CHPP and a BESS. Running the CHPP, when the produced heat is not required may be disadvantageous depending on the gas and electricity prices. An erroneous prediction of the amount of stored heat could lead to the CHPP starting up, instead of providing the same electrical power with the help of the BESS. This may prevent the simulation model from following the surrogate's schedule at a later time. To deal with this issue, each schedule deemed unachievable by the simulation model is evaluated a second time by solving a MILP that searches the schedule with the minimum MAE (see Section 5.3.3). If the schedule is feasible, the target function value is 0. In the strict sense, these schedules are infeasible, as the reference model, i.e., the simulation model, is not able to reproduce them. However, even though the schedule is not optimal in the sense of operating devices in the most beneficial way, it is, after all, still feasible and could therefore be requested by an external entity. Moreover, the only way to incorporate all possible state trajectories in the simulation model and thereby encode them into the ANNs, is by defining an individual action for any combination of DER actions, as we do in **[308]**. As explained before, this would lead to an increased complexity for the external entity.

5.3.6 Results

Surrogate models, i.e., combinations of classifiers and state estimators as required by the state-based simulation approach, have been trained for each DER configuration utilizing the models and methodology outlined above. The set of possible actions for each configuration has been defined by splitting the range between the minimum and maximum attainable power into equidistant parts. For the single DER configurations, each step in positive (negative) direction equals 1% of the maximum (minimum) power. With the exception of the aggregated BESS, for which a step size of 0.5% was used, the aggregated systems' actions all lie 100 W apart from each other. It is important to note that there is a trade-off between complexity and detailedness. Generally speaking, more actions allow for a more precise description, but may lead to a worse approximation quality. In an attempt to provide rather precise models, small distances between actions have been selected, but ultimately all step sizes are arbitrary choices.

Since the ultimate goal is to assess the quality of approximation achievable by ANN-based surrogate modeling, a rather large number of samples has been used for training the ANNs. Up to 1000 training epochs with a dataset of 1000 batches, each consisting of up to 3072 samples, have been conducted. If the score of a model did not improve within 100 epochs, training was stopped early. For the aggregated BESS the batch size had to be halved because of limited video RAM. During the training, the dataset was regularly updated, replacing the oldest samples in the dataset with newly generated ones. For each required ANN, 16 attempts with random hyperparameters and ANN topology were made. The best performing ANNs were selected automatically and an evaluation was conducted.

In this section, the results gained for each individual configuration are presented. Results were generated by testing the classifiers with 100,000 randomly selected states and by creating 10,000 load schedules following the steps described in the previous section. The resulting schedules had a length of one day, split into 96 time steps of 15 minutes. In order to evaluate the approach for more short-sighted applications or applications working with a coarser temporal resolution, all statistics have also been computed using only the first 4 and 24 elements of each schedule.

Battery Energy Storage System

The first investigated configuration was a single BESS. Its range of action was split into 201 distinct actions, one "idle" action and 100 actions for charging and discharging, respectively. Minimum and maximum SOC were fixed throughout schedule generation. An overview of the results is given in Table 5.11. For the first entry in any load schedule, only the performance of the classifier is relevant. The classification results for 100,000 randomly selected states are listed in the upper portion of the table. Each individual action was rated positive or negative, depending on the numerical rating and the (arbitrary) feasibility threshold of 0.5. With 201 actions, there were 201 classifications per state, of which each is either true or false. The listed shares of true positives, true negatives, false positives, and false negatives are the averages of all $201 \cdot 100,000$ individual classifications. With only a BESS to consider, the classifier exhibited a rather high accuracy. Both, the share of false positives and false negatives, were almost negligible with less than 0.0025%. Assuming the independence of false positives (false negatives), this means a $\sim 0.5\%$ chance ($P(X \geq 1)$ with a binomially distributed $X \sim Bin(n = 201, p = 0.000025)$) to have a false positive (false negative) in a given state sampled from the training distribution.

Table 5.11: Evaluation results for the BESS. The single time step metrics relate to a total of $201 \cdot 100,000$ classifications, performed in 100,000 ANN evaluations. For each combination of strategy and buffer size 10,000 schedules have been generated.

			Single time step			
	True positive	True negative	False positive	False negative	FPR	FNR
	87.3932%	12.6038%	0.0017%	0.0013%	0.0133%	0.0014%
Multiple time steps						
		Purely random			Random reference	
	4 steps	24 steps	96 steps	4 steps	24 steps	96 steps
Feasible	100.00%	99.95%	99.61%	99.94%	98.65%	86.76%
MAE*	0.00	2.08	0.52	12.50	2.08	0.54
RMSE*	0.00	10.21	5.10	25.00	10.21	5.21
			Multiple time steps with a 1% buffer			
Feasible	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
MAE*	0.00	0.00	0.00	0.00	0.00	0.00
RMSE*	0.00	0.00	0.00	0.00	0.00	0.00

*only infeasible schedules

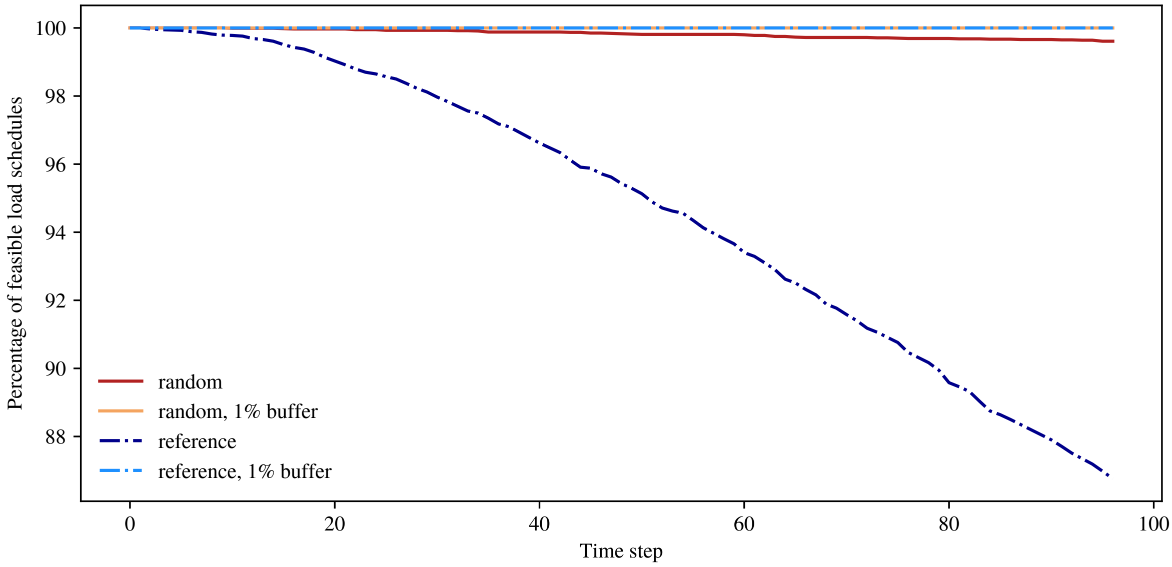

Figure 5.12: Feasibility of BESS schedules over time.

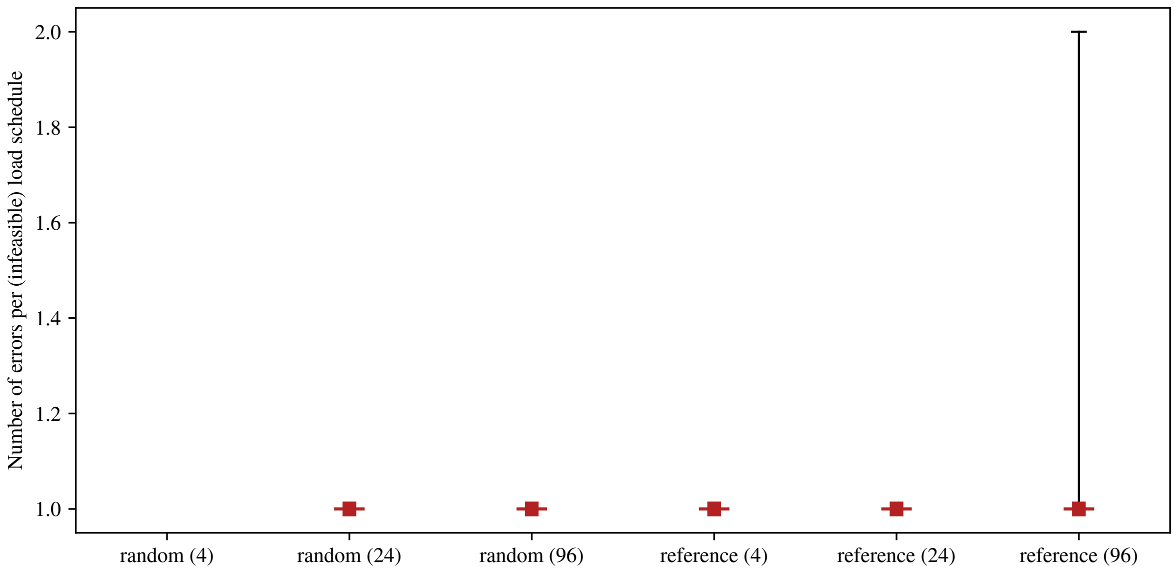

Figure 5.13: Number of erroneous actions per infeasible BESS schedule. No buffers have been applied during schedule generation.

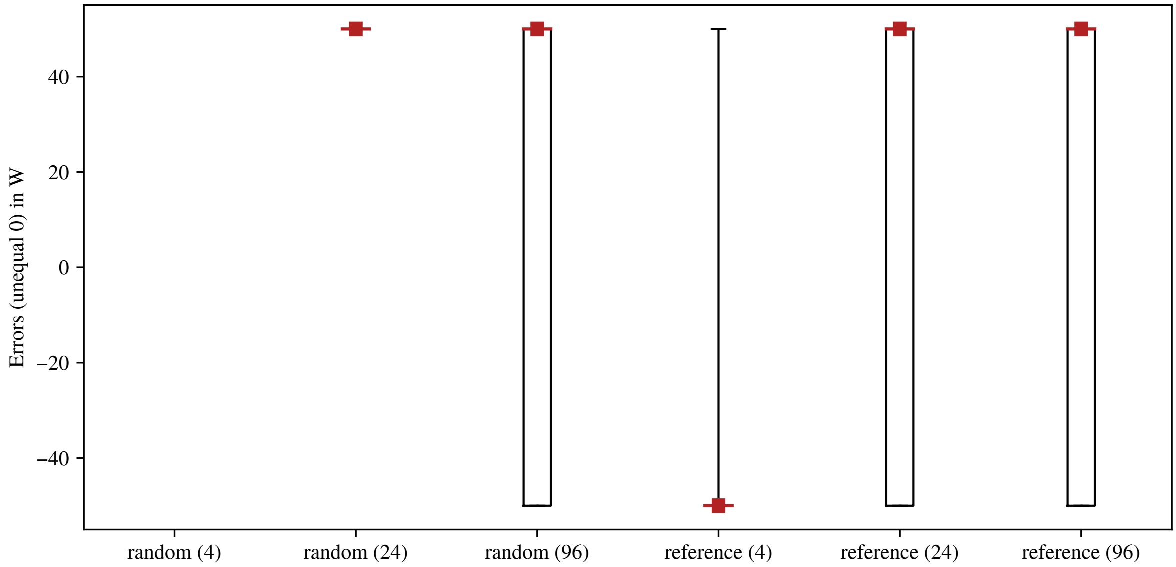

Figure 5.14: Errors (not equal to 0) of infeasible BESS schedules. Actions in the BESS configuration range from -5000 W to 5000 W. No buffers have been applied during schedule generation.

When it comes to generating entire schedules, the two employed strategies for selecting actions show large differences as long as no buffers are used. Since schedules become infeasible the time step at which an invalid action is chosen and remain infeasible from then on, the number of achievable schedules decreases with each passing time step. This process is visible in Figure 5.12, which depicts the percentages over time. With 99.61% feasible schedules even for the 96 time step horizon, the purely random selection of actions achieved very high percentages of feasibility. In contrast, selecting actions close to a given less volatile (random) reference schedule resulted in only 86.76% feasible schedules after 96 time steps. The reason for this comparatively bad result is the forced approach of the boundary regions, where any error by the classifier or state estimator can cause the selection of an invalid action. With the purely random action selection, any action deemed feasible is equally likely to be selected. Therefore, when having an SOC around 0.5, the strategy frequently switches between charging and discharging, as both are equally likely to be selected. The closer an SOC boundary is approached, the fewer actions pointing in this direction are classified feasible, resulting in a growing likelihood of selecting an action pushing the SOC in the opposite direction. The reference-based strategy, on the other hand, tries to follow a reference profile, which demands charging or discharging for multiple consecutive steps. Hence, even when a boundary is approached, the SOC may still be pushed closer and closer to the bound. With a buffer of 1% at each boundary, i.e., constraining the surrogate model to a minimum SOC of 1% and a maximum SOC of 99% by adapting the state vector, all generated schedules were valid, regardless of the employed strategy. Aside from the share of feasible schedules, the table lists the MAE and RMSE of the infeasible schedules. Both decrease with increasing schedule length, as the share of valid time steps, that is, steps with an error of zero, increases.

Figure 5.13 depicts the number of invalid actions per generated load schedule. Feasible schedules are not included in the numbers. Infeasible schedules of 96 time steps had up to two invalid actions, when using the reference-based strategy, instead of one invalid action with the random strategy. The median and 75%-quartile are at one error per schedule. The number of errors is measured with the help of the simulation model. If an invalid action is chosen, the model corrects it to the closest feasible action and the generation process continues like it would after a valid choice. The respective distribution of errors is illustrated in Figure 5.14. Each error is computed by subtracting the action performed by the simulation model from the intended action given by the schedule. Thus, a positive number means that the resulting power flow is less than the requested power, and vice versa. Only values unequal zero, that is, actual errors, are considered in the boxplots. In any case, the absolute error was at most 50 W. Please remember that the actions range from -5000 W to 5000 W for the considered BESS configuration. Therefore, even the infeasible schedules generated by using the reference-based action selection and no buffers at all, were remarkably close to feasible schedules, especially when considering that the majority contains only a single incorrect value.

Electric Vehicle Supply Equipment

The EVSE model was trained with 101 distinct actions, ranging from 0 kW to 22 kW in steps of 1%, i.e., 220 W. Any action greater than 0 kW but below 3.6 kW is always invalid, as a minimum charging power of 3.6 kW is assumed. All parameters defining the individual charging processes, that is, the capacity, initial SOC, remaining standing time, and the minimum and maximum SOC were provided as a forecast. Hence, only the SOC needed to be estimated. The results for the EVSE are summarized in Table 5.12. The shares of false positive and false negatives were comparatively small. In contrast to the BESS, the majority of actions was infeasible for the EVSE, which is visible in the shares of true positives and true negatives. This was caused by the test dataset being generated the same way the training and evaluation data sets are generated. To make sure the classifier learns the charging constraints correctly, it was mostly confronted with states in which only a few time steps remain for charging. States with low remaining standing times and states where no BEV is available are states where only few actions are valid.

For the EVSE, the purely random action sampling and the reference-based strategy performed similarly. They ended up with 58.33% and 57.10% of the schedules being feasible after 96 time steps. A surprisingly big share became infeasible early in the generation process. After 4 time steps 94.39% and 93.35% were still valid, which in turn means that around 6% already lost their validity. This initial drop is also visible in Figure 5.15. Subsequently, the slope flattens, as fewer schedules became infeasible. A major factor to keep in mind is the limited availability of the BEV, which is not only determined by the remaining standing time, but also the arrival rate. In this evaluation, initial states for each schedule are drawn from the same distribution the training states are drawn from. Therefore, almost all schedules are starting with a charging process being under way. Furthermore, this initial charging process is often limited to only a few remaining time steps, which again increases the likelihood of errors since additional constraints for the charging power are active. After finishing a charging process, and due to the expected arrival of one BEV per 48 time steps, the EVSE oftentimes idles. As time passes, more and more simulated schedules transition from the initial charging process to idling most of the time, which in turn explains the less than exponential decay of feasible schedules later on. Hence, when there is a less challenging distribution of initial states, better results than those presented here can be expected. The MAE and MSE values were considerably larger than those achieved with the BESS model, even when taking the larger range of action into account.

Table 5.12: Evaluation results for the EVSE. The single time step metrics relate to a total of $101 \cdot 100,000$ classifications, performed in 100,000 ANN evaluations. For each combination of strategy and buffer size 10,000 schedules have been generated.

Single time step						
	True positive	True negative	False positive	False negative	FPR	FNR
	25.4650%	74.5319%	0.0021%	0.0009%	0.0029%	0.0037%
Multiple time steps						
		Purely random			Random reference	
	4 steps	24 steps	96 steps	4 steps	24 steps	96 steps
Feasible	94.39%	80.22%	58.33%	93.35%	79.42%	57.10%
MAE*	282.16	92.92	35.60	243.65	73.79	30.70
RMSE*	741.88	511.82	349.27	641.66	431.63	317.50
Multiple time steps with a 1% buffer						
Feasible	99.86%	98.76%	94.54%	99.82%	98.93%	94.79%
MAE*	192.50	83.68	38.65	259.72	72.65	38.49
RMSE*	437.04	411.90	371.54	485.75	316.41	368.69
Multiple time steps with a 2% buffer						
Feasible	99.99%	99.92%	99.37%	99.99%	99.92%	99.43%
MAE*	275.00	79.06	37.36	330.00	59.58	36.63
RMSE*	550.00	415.54	370.22	491.93	300.41	368.32

*only infeasible schedules

Since the remaining standing time for each time step is provided to the external entity as part of the forecast, there are only two possible sources for errors, misclassification and imprecise SOC forecasts. In the experiment, the major cause of infeasibility was the imprecise prediction of the BEV's SOC. A closer look at the state estimator showed occasional errors of up to around 4.8 percent points in both directions. The 0.25- and 0.75-quartiles of the observed SOC errors are -0.03 percent points and 0.04 percent points, respectively. Please keep in mind that in contrast to the BESS, the EVSE must be able to handle batteries of various capacities. Imprecise prediction can be kept in check by using buffers. With increasing buffer size the share of feasible schedules quickly converged to 100%. More than three-thirds of all infeasible schedules generated without buffers contained at most 2 errors, as is visible in Figure 5.16, where the 0.75-quartile is at the value 2. Overall, none of the schedules had more than 15 errors. However, the isolated errors were comparatively large and ranged up to 7.92 kW, meaning that the requested action was to charge with 7.92 kW more than is possible for the simulation model. Regarding the remaining standing time, even though the predicted value was replaced by the provided forecast, the state transition ANN correctly determined the subsequent value for all 100,000 states in the validation set. This particular result was achieved by using a one-hot encoding and the cross entropy loss for the remaining standing time, which makes the prediction task trivial. Since the remaining standing time was predicted so well, instead of passing an individual

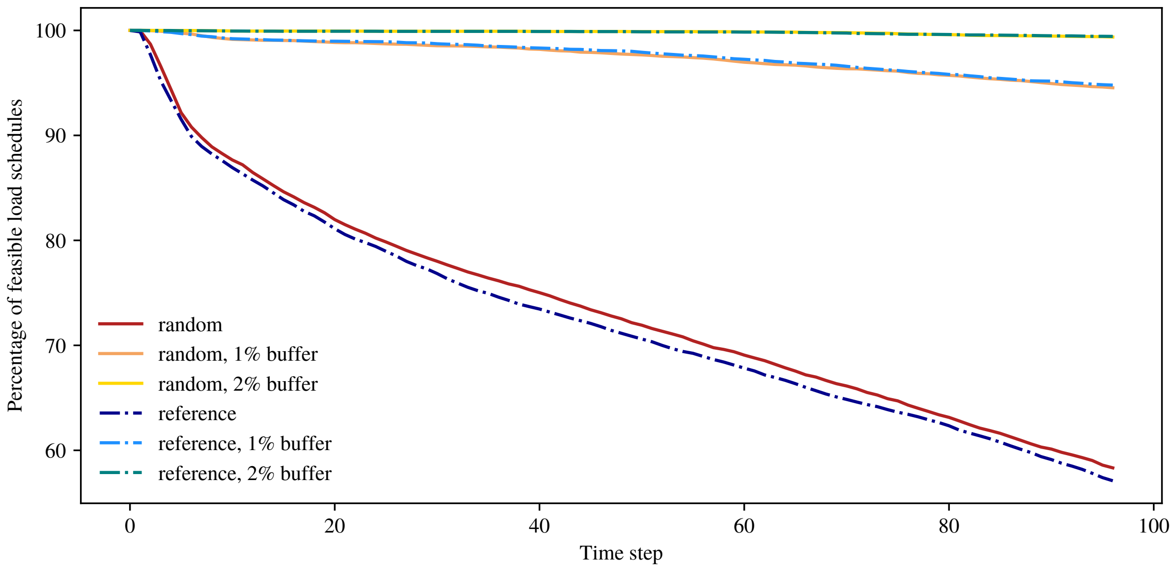

Figure 5.15: Feasibility of EVSE schedules over time.

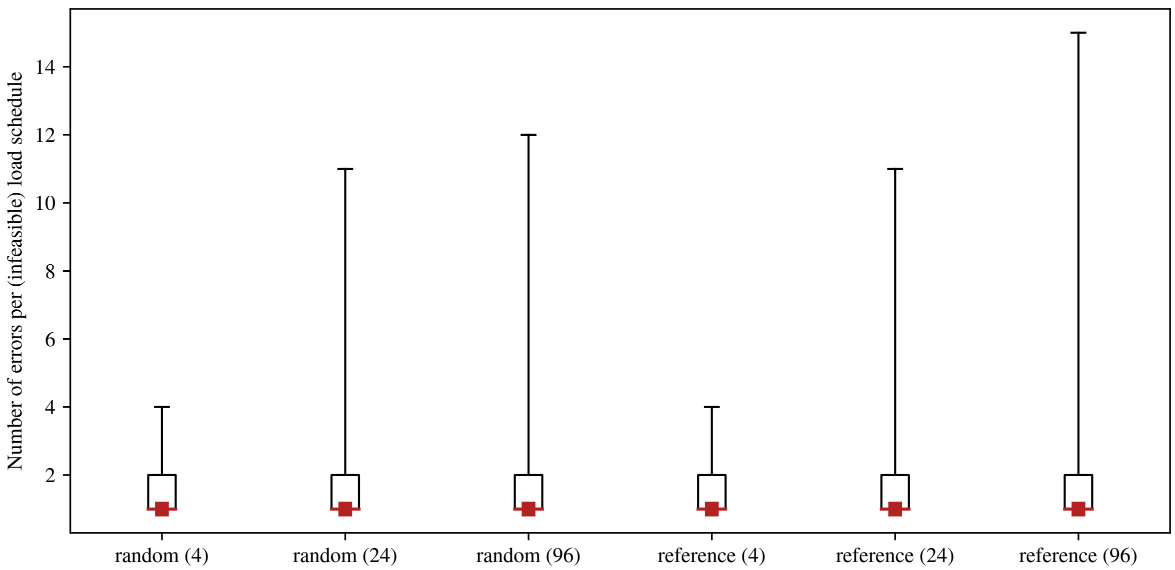

Figure 5.16: Number of erroneous actions per infeasible EVSE schedule. No buffers have been applied during schedule generation.

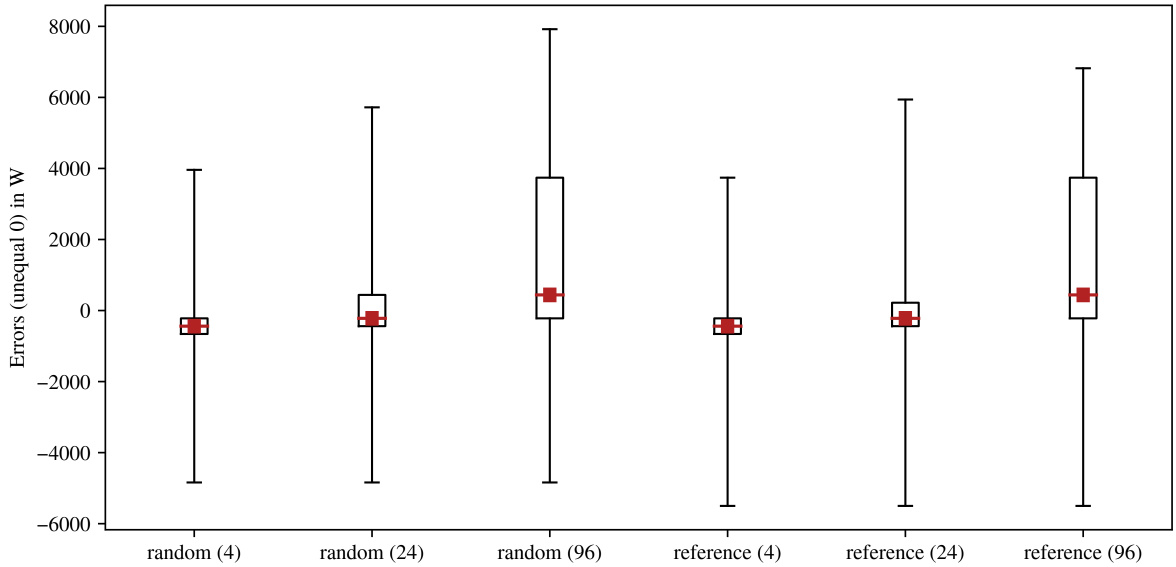

Figure 5.17: Errors (not equal to 0) of infeasible EVSE schedules. Actions in the EVSE configuration range from 0 kW to 22 kW. No buffers have been applied during schedule generation.

forecast for each time step, it would be sufficient to only include the initial time step and value for each charging process.

Combined Heat and Power Plant Satisfying Heat Demand

For the CHPP configuration, the range of action starting at -5.5 kW and ending at 0 kW, was split into 101 equidistant actions. A new, randomly created heat demand forecast was given for each individual schedule generation attempt. During training, the CHPP was additionally constrained to not start when the HWT temperature is 79 °C or higher, and not stop when it is 61 °C or below (see Section 5.3.3). Throughout the evaluation, these additional restrictions did not apply. Table 5.13 provides a summary of the achieved results. The CHPP was configured to allow two modes of operation, "on" (-5.5 kW) and "off" (0 kW), and provided -2.75 kW while transitioning between both modes. Hence, of the 101 actions, at most two actions could be feasible at any given time, which manifests itself in the high share of true negatives and low share of true positives listed in Table 5.13. Not a single false positive was observed in the evaluation. The false negative rate, on the other hand, was comparatively high.

Both strategies, the purely random choice of actions and the reference-based choice performed similarly well. This is not surprising, given that frequent changes of the operation mode are prevented by the minimum dwell times. Hence, in case of the CHPP, both strategies produce similar schedules. After 96 time steps, the share of feasible schedules was 90.08% and 89.05%, respectively. However, the MAE and RMSE were even bigger than those of the two previous configurations. Figure 5.18 shows the feasibility of the schedules over time. Similar to the BESS, the slope is becoming steeper and not flatter with a growing number of time steps. This suggests that the primary cause for choosing an invalid action was the error of the estimated state, which generally grows with every time

Table 5.13: Evaluation results for the CHPP configuration. The single time step metrics relate to a total of $101 \cdot 100,000$ classifications, performed in 100,000 ANN evaluations. For each combination of strategy and buffer size 10,000 schedules have been generated.

Single time step						
	True positive	True negative	False positive	False negative	FPR	FNR
	1.4658%	98.5102%	0.0000%	0.0240%	0.0000%	1.6096%
Multiple time steps						
	Purely random			Random reference		
	4 steps	24 steps	96 steps	4 steps	24 steps	96 steps
Feasible	99.94%	98.95%	90.08%	99.81%	98.98%	89.05%
MAE*	1604.17	288.10	119.29	1628.29	438.11	127.27
RMSE*	2381.57	923.19	607.74	2422.99	1225.31	632.05
Multiple time steps with a 1°C buffer						
Feasible	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
MAE*	0.00	0.00	0.00	0.00	0.00	0.00
RMSE*	0.00	0.00	0.00	0.00	0.00	0.00

*only infeasible schedules

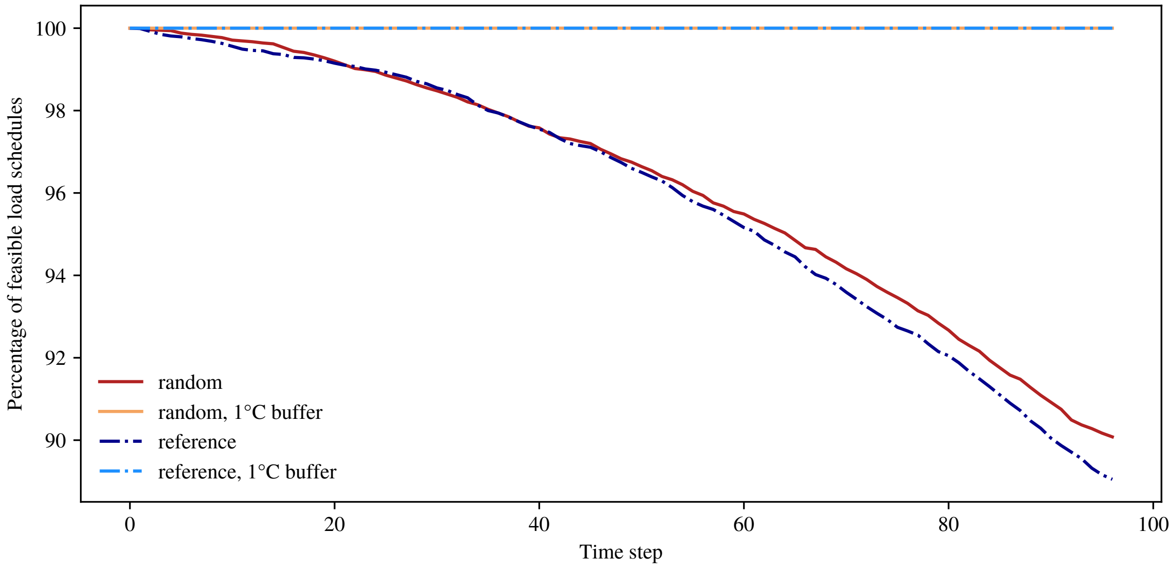

Figure 5.18: Feasibility of schedules for the CHPP configuration over time.

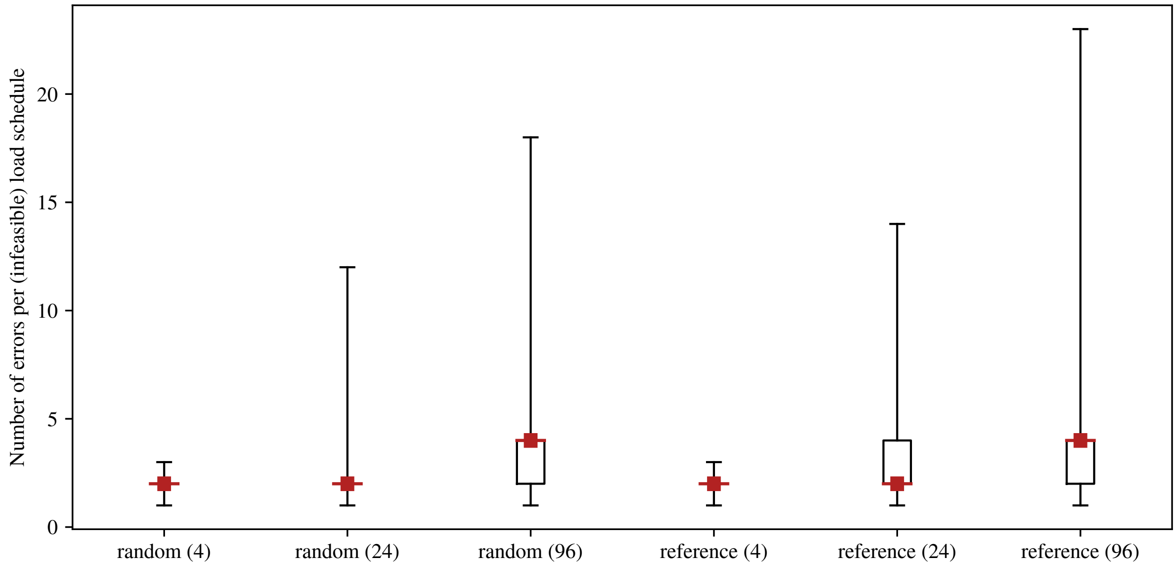

Figure 5.19: Number of erroneous actions per schedules infeasible for the CHPP configuration. No buffers have been applied during schedule generation.

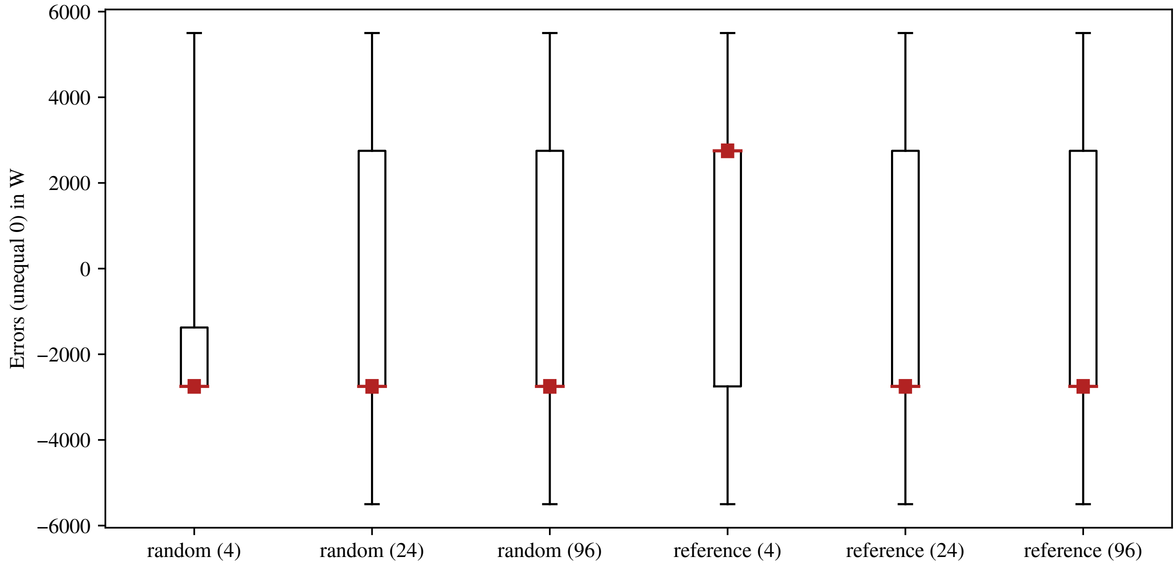

Figure 5.20: Errors (not equal to 0) of schedules infeasible for the CHPP configuration. Actions in the CHPP configuration range from -5.5 kW to 0 kW. No buffers have been applied during schedule generation.

step. The number of errors per infeasible schedule, which is depicted in Figure 5.19, was again rather small. The 0.75-quartiles are 2 and 4 errors per schedule, respectively. The errors themselves are shown in Figure 5.20. They ranged from -5.5 kW to 5.5 kW with many errors around either -2.75 kW or 2.75 kW. These two values are not surprising, as they are the only operational options apart from 0 kW. The medians are mostly at -2.75 kW. An error of -2.75 kW is recorded when the generated schedule tries to start the CHPP even though it is not possible at that moment in time. Such failed attempts to start or stop the CHPP were mostly caused by inaccurate temperature forecasts and the associated rules for forcing starts and stops. This argument is further supported by the observed lack of infeasible schedules once a buffer of 1 °C was introduced.

Detached Family Home

The detached family home combines the BESS and CHPP configurations. Therefore, in sum, the possible actions range from -10.5 kW to 5 kW. This range was split into 156 actions, yielding steps of 100 W. Analogous to the CHPP configuration, the surrogate for the detached family home has been trained with additional constraints, preventing the CHPP from starting when the HWT temperature is 79 °C or higher, and prohibiting to stop when it is 61 °C or below (see again Section 5.3.3). Table 5.14 shows the test results for the detached family home. Unlike the previous configurations, the detached family home comprises multiple DERs.

As explained earlier, for aggregated systems, there is trade-off between having multiple, possibly countless, actions for each power level and having only one after eliminating each degree of freedom. Since, in order to reduce complexity, the latter option has been chosen, the implemented simulation models may not be able to reproduce a schedule, even though it would generally be achievable by the DERs in their given states. In order to analyze the general reproducibility, even though it requires deviating from the intended operational logic, a further evaluation was conducted with the help of the optimization model. The MILP-based results are listed in the rows labeled "Feasible (MILP)". Especially the reference-based strategy for selecting loads seems to produce schedules that are infeasible for the simulation model, but achievable for the MILP. At 96 time steps 67.97% were actually reproducible, while the simulation was able to follow only 47.29% without deviation. However, strictly speaking, the additional 20.68% of schedules were still infeasible, as the simulation model defines the ground truth. In order to fulfill such a schedule, the flexibility provider needs to deviate from the logic trained into the surrogate's state transition ANN. Then again, the schedule is achievable, which means given sufficient compensation, the flexibility provider may still be inclined to implement it. Nonetheless, without an analysis in the context of a specific application, it is not possible to state which of these two rates is more meaningful. For this reason, both rates are listed.

A buffer of 1% for the BESS and 1 °C for the HWT temperature sufficed to raise the number of feasible schedules over 88%. With a 2% and 2 °C buffer, more than 89% were achievable. Please note that in this particular case the results for the two different buffer sizes are very similar. For the random strategy, the larger buffer even produced a worse result. In preliminary tests, the larger buffer usually led to slightly higher shares of feasible schedules. Most likely, due to randomness, the achieved results for the smaller buffer using the random strategy are particularly good, while those for the larger one are particularly bad. Assuming a fixed chance of generating a feasible schedule, the total number of feasible schedules after 10,000 attempts follows a binomial distribution, since each schedule is generated independently. With $p = 90\%$, in 95 of 100 runs of the experiment, the total number can be expected to lie within the interval $[8940, 9058]$. This means that fluctuations of about half a percent point are easily possible, which supports the hypothesis proposed above.

Table 5.14: Evaluation results for the detached family home. The single time step metrics relate to a total of $156 \cdot 100,000$ classifications, performed in 100,000 ANN evaluations. For each combination of strategy and buffer size 10,000 schedules have been generated.

	Single time step					
	True positive	True negative	False positive	False negative	FPR	FNR
	65.0322%	34.5162%	0.0047%	0.4469%	0.0136%	0.6825%
	Multiple time steps					
		Purely random			Random reference	
	4 steps	24 steps	96 steps	4 steps	24 steps	96 steps
Feasible	99.91%	97.90%	81.57%	99.44%	90.37%	47.29%
Feasible (MILP)	99.93%	98.88%	90.31%	99.55%	93.29%	67.97%
MAE*	91.67	94.37	36.26	97.77	32.69	21.48
RMSE*	266.15	534.01	320.05	438.49	310.77	243.34
	Multiple time steps with a 1% and 1°C buffer					
Feasible	99.93%	99.08%	90.82%	99.94%	98.84%	88.21%
Feasible (MILP)	100.00%	99.91%	99.32%	100.00%	99.90%	98.33%
MAE*	367.86	158.42	53.27	1225.00	185.20	59.70
RMSE*	837.73	716.11	390.91	2527.35	773.87	430.20
	Multiple time steps with a 2% and 2°C buffer					
Feasible	99.94%	98.95%	90.03%	99.93%	98.86%	89.26%
Feasible (MILP)	99.98%	99.93%	99.67%	100.00%	99.98%	99.67%
MAE*	954.17	119.13	49.76	564.29	157.27	53.82
RMSE*	1875.83	555.61	375.81	1346.42	713.57	402.39

*only infeasible schedules (according simulation)

The MAE and RMSE values lay in-between those of the BESS and CHPP configurations, since the additional BESS is able to compensate for some of the CHPP related errors. Interestingly, MAE and RMSE both grew with the introduction of buffers. Please keep in mind that MAE and RMSE are computed only for infeasible schedules. As the number of infeasible schedules decreases, this means that there are fewer errors in the entirety of generated schedules, but they are larger. In the previous configurations the MAE and RMSE values, if not equal to zero anyway, seemed to underlie minor, random fluctuations, but for the detached family home a clear trend was visible. The same effect could be observed in the HoLL configuration. The errors of the aggregated BESS, on the other hand, behaved as expected and decreased with increasing buffer sizes, which suggests that the effect is connected to the CHPP or its interplay with the BESS.

To further analyze the growing errors caused by buffers, a series of manual tests has been conducted. Before discussing the findings, it is important to remember that there are different kinds of buffers used in this configuration, as is explained in Section 5.3.3. In summary, the BESS's SOC buffer is set by state variables and, thus, can be changed any time by adapting the state vector. The CHPP and HWT, in contrast, have a fixed temperature buffer which is learned by the ANN during surrogate generation. Changing the buffer size in the schedule generation process does not affect this learned buffer. Instead, the HWT related constraints are relaxed. Therefore, increasing the buffer has two effects. Firstly, the BESS's range of action is narrowed, secondly, the temperature constraints may be violated more. One possible reason for the increased errors could be the ANNs performing worse with the changed input variables. However, although it does not completely rule out the ANNs as a cause, a dedicated test with randomly sampled states did not show any general difference in performance. A closer look into the state vectors produced during the schedule generation process revealed that infeasibility was mostly caused by the incorrect prediction of the CHPP's behavior. In contrast to the CHPP configuration, where the CHPP is the only DER, in this configuration, the BESS state has a strong influence on CHPP control. Whenever there is a degree of freedom, i.e., there is more than one option to implement an action, the CHPP mode is selected according to a fixed set of rules with the aim of reducing costs. The BESS only covers the difference between the desired CHPP power and the target defined by the requested action. In turn, this means that the CHPP must change modes whenever the BESS is not able to cover the difference. Therefore, decreasing the BESS's flexibility with a buffer forces the CHPP to change modes more often. Consequently, the contribution of the CHPP to the overall load schedule increases. Keep in mind that the CHPP configuration on its own exhibited very large MAEs and RMSEs. In addition to this finding, it could be observed that the state prediction surrogate occasionally neglected the SOC buffers, which again can result in incorrect CHPP modes. While setting the wrong CHPP mode does not immediately lead to infeasibility, the longer the schedule becomes, the farther the real and predicted state trajectories may diverge. In infeasible schedules, the temperature error often exceeded the 2 °C buffer. Additionally, the deviations are amplified by long minimum dwell times. If no dwell time restrictions had been considered, the results would likely be much better. In summary, the increase is caused by multiple factors combined, with reduced BESS flexibility and ANN prediction errors being major causes.

Figure 5.21 and Figure 5.22 depict the share of feasible load schedules over time. The graphs have been split into two separate figures, to allow for a better comparison. With one exception, the graphs look similar to those of the BESS and CHPP configurations, starting flat and becoming steeper over time. Only the curve associated to the reference-based strategy changes its curvature and flattens again after about half of the time horizon. The number of observed errors is depicted in Figure 5.23. With 1, 2, and 3 the values for the 0.75-quartiles are again small, although there were up to 16 and 21 errors in a single schedule. The extent of the individual errors is shown in Figure 5.24. Errors ranged from -10.1 kW to 9.7 kW, covering almost the entire range of possible errors. Similar to the CHPP configuration, large deviations were caused by the surrogate not correctly describing the CHPP. However, the BESS is able to counteract some of these errors. For the reference-based strategy, the 0.25 and 0.75-quartiles are -600 W and -100 W after 96 time steps, and for the random strategy, they are -1400 W and 600 W, respectively. Since the random strategy generally involves more mode changes of the CHPP, this result suggests that one major cause of deviations was illicit attempts to change the CHPP's operation mode.

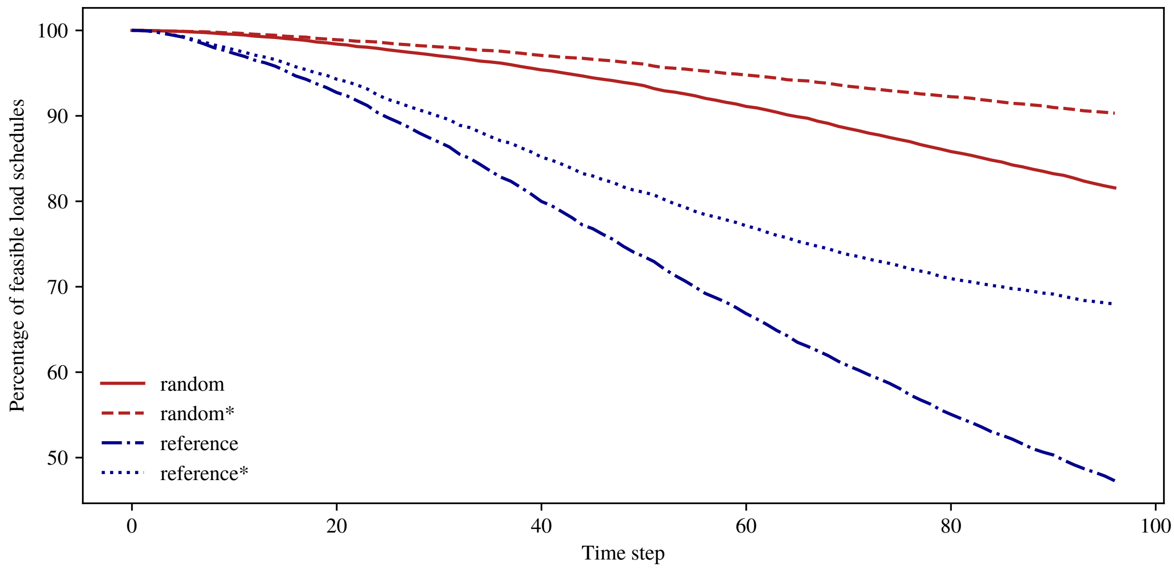

Figure 5.21: Feasibility of detached family home schedules over time.

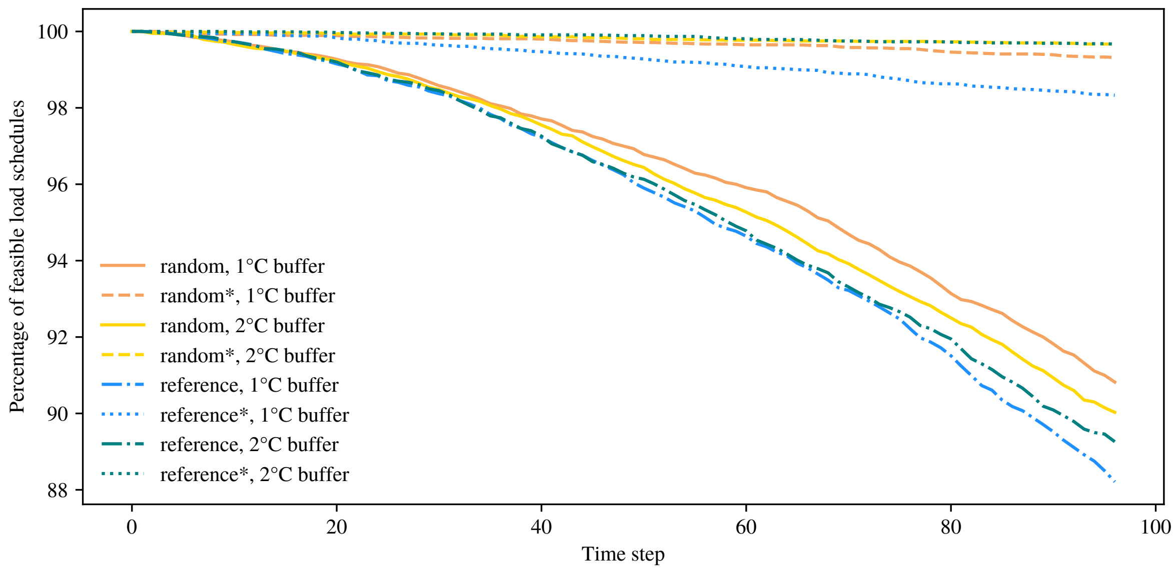

Figure 5.22: Feasibility of detached family home schedules over time.

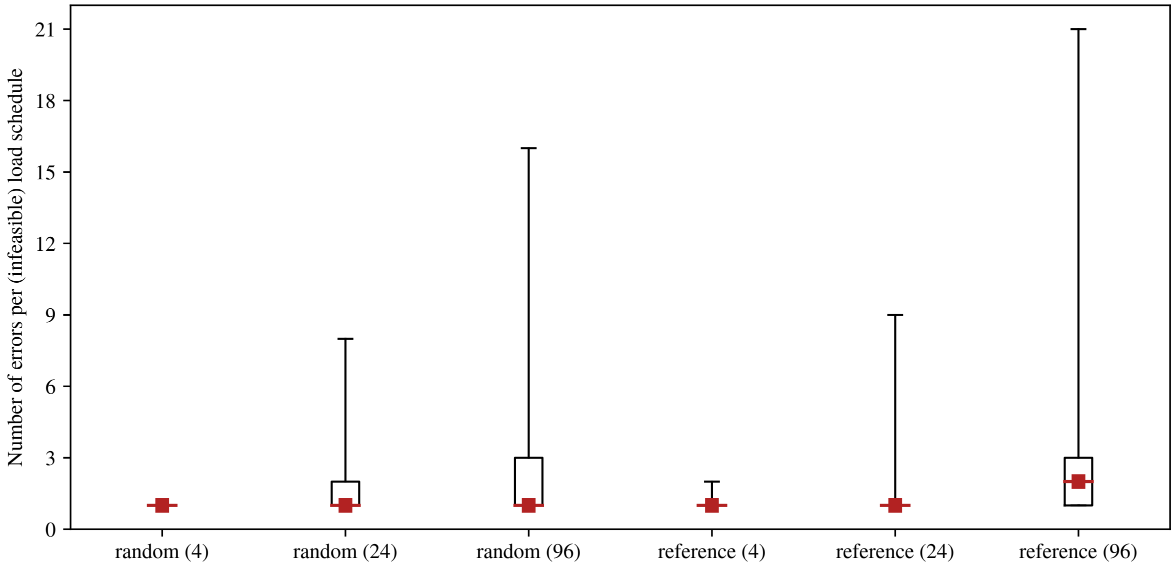

Figure 5.23: Number of erroneous actions per schedules infeasible for the detached family home. No buffers have been applied during schedule generation.

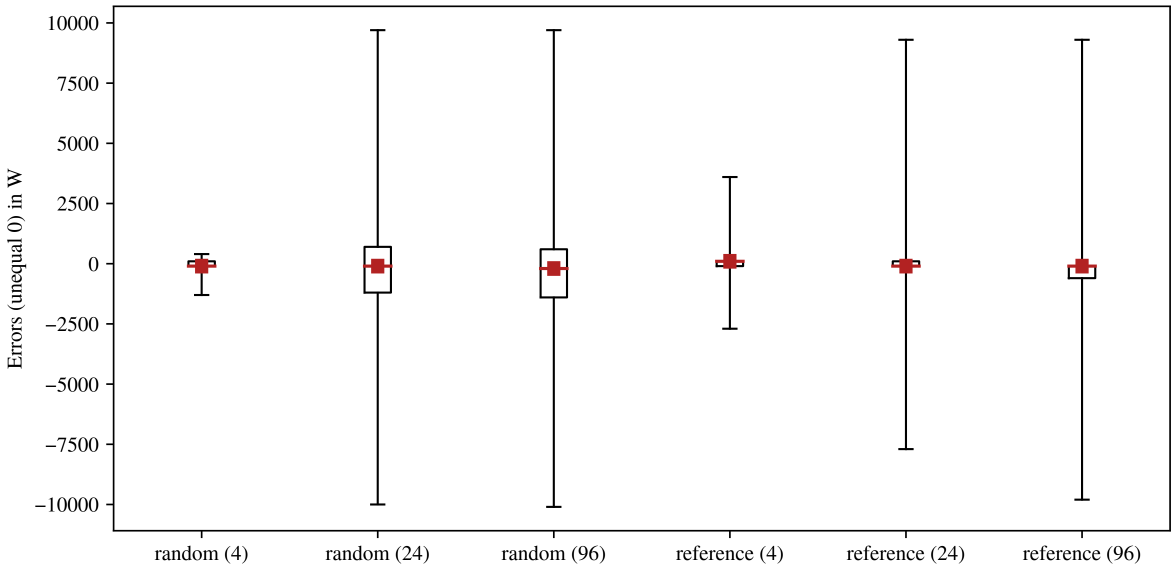

Figure 5.24: Errors (not equal to 0) of schedules infeasible for the detached family home. Actions this configuration range from -10.5 kW to 5 kW. No buffers have been applied during schedule generation.

FZI House of Living Labs

With its BESS, CHPP, HWT, GCB, and EVSE, the HoLL is the configuration with the most heterogenous DERs. In combination, a total load ranging from -7.9 kW to 24.4 kW is achievable. Steps of 100 W have been selected to create the set of discrete actions which was used by the surrogate.

Table 5.15: Evaluation results for the FZI House of Living Labs. The single time step metrics relate to a total of $324 \cdot 100,000$ classifications, performed in 100,000 ANN evaluations. For each combination of strategy and buffer size 10,000 schedules have been generated.

Single time step						
	True positive	True negative	False positive	False negative	FPR	FNR
	38.2278%	61.7311%	0.0178%	0.0232%	0.0289%	0.0607%
Multiple time steps						
	Purely random			Random reference		
	4 steps	24 steps	96 steps	4 steps	24 steps	96 steps
Feasible	97.39%	89.96%	51.67%	95.81%	84.58%	42.16%
Feasible (MILP)	99.31%	97.07%	77.40%	98.42%	95.27%	78.82%
MAE*	379.79	116.99	73.21	331.03	81.76	59.69
RMSE*	1113.20	600.55	460.54	1034.91	506.73	425.14
Multiple time steps with a 1% and 1°C buffer						
Feasible	99.26%	96.23%	74.39%	98.85%	95.55%	67.79%
Feasible (MILP)	99.96%	99.57%	97.69%	99.84%	99.51%	96.43%
MAE*	694.93	219.12	79.10	623.48	247.98	113.47
RMSE*	1510.09	855.31	500.78	1374.22	966.13	653.75
Multiple time steps with a 2% and 2°C buffer						
Feasible	99.50%	96.36%	76.08%	98.91%	95.55%	70.13%
Feasible (MILP)	99.99%	99.78%	98.87%	99.96%	99.85%	98.72%
MAE*	678.00	177.60	80.38	765.83	256.40	129.84
RMSE*	1521.09	771.34	508.94	1626.11	973.23	703.11

*only infeasible schedules (according simulation)

The results in Table 5.15 show that, on average, the classifier ANN misclassified $324 \cdot (0.0178\% + 0.0232\%) \approx 0.13$ of the 324 actions in each given state. This equates to one misclassified action every 7.5 steps. During load schedule generation, the errors of the predicted state heavily impair the identification of valid actions. After 96 time steps, 51.67% and 42.16% of the schedules were feasible, when no buffers were used. In contrast, the MILP was able to reproduce 77.40% and 78.82% of the generated schedules. The large gap can be explained with the very flexible DERs and the operation logic implemented in the simulation model. While the simulation model tries to use the CHPP only when the temperature of the HWT is low and prioritizes charging the BEV over charging the BESS, the MILP can control the DERs freely. Especially the possibility to charge the BESS before the BEV gives the MILP an advantage in reproducing the generated schedules, as only a charged BESS can be discharged. But then again, this higher flexibility comes with increased costs. A small buffer of 1% for the BESS and EVSE, as well as a 1 °C buffer for the HWT temperature, improved the share of feasible schedules drastically. Doubling the buffers to 2% and 2 °C yielded only minor improvements. Overall, with buffers of 2% and 2 °C, 98.87% and 98.72% of the generated schedules were achievable by the simplified HoLL, when forgoing the desired DER prioritization. Around 99% of the load schedules were feasible at the fourth time step. The MAE and RMSE values are smaller than those found in the CHPP configuration, which can again be explained by the presence of a battery storage. Like in the previous configuration, the errors grow with the introduction of a buffer, but to a lesser extent. The same explanation as before applies here, too. However, the effect seems to be dampened by the additional DERs.

The share of feasible load schedules over time is depicted in Figure 5.25 and Figure 5.26. All graphs show the same initial dip that can be observed in the EVSE configuration. Here again, the reason lies in the distribution of the initial states. In the great majority of initial states, there is a BEV present and a charging process is about to finish. With less BEVs and longer remaining charging times in the initial states, better results are very likely. The remaining progression over time is similar to the other configurations examined before, which isn't surprising, as the HoLL combines all of them. Figure 5.27 and Figure 5.28 depict the number of errors and the distribution of said errors, when no buffers are used. The 0.75-quartiles for the number of errors are at the values 1, 2, 5, and 8, depending on the strategy and time horizon. The maximum number of errors observed in a single schedule reaches 54 and 65 at 96 time steps, and is therefore quite high. With a 2% and 2 °C buffer, the highest observed numbers were 33 and 48 (not depicted). Errors range from -17.8 kW to 23.4 kW. For an error of -17.8 kW to appear, the surrogate must fail to charge the BEV, even though it is required. An error of 23.4 kW can occur when charging the BEV is attempted but fails. The 0.25 and 0.75-quartiles are -1 kW and -0.1 kW, as well as -1.5 kW and 1.4 kW for

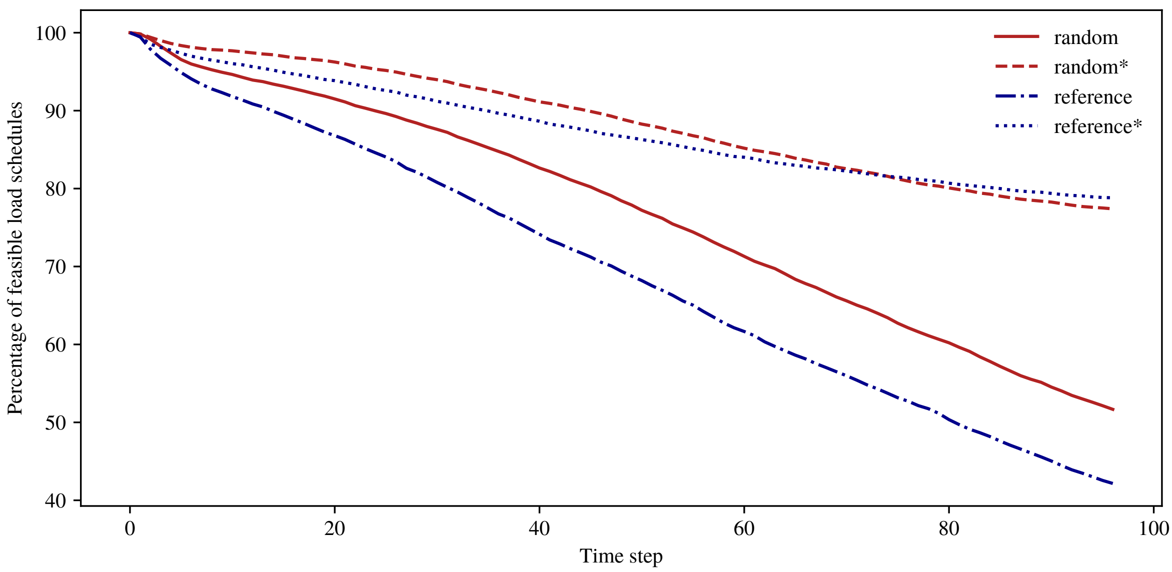

Figure 5.25: Feasibility of HoLL schedules over time.

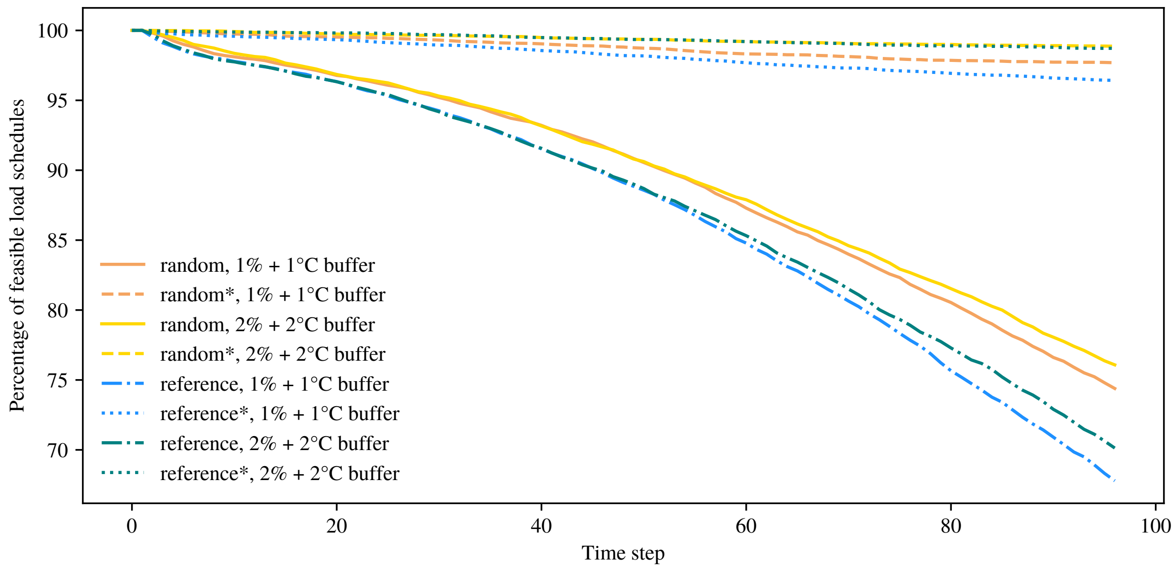

Figure 5.26: Feasibility of HoLL schedules over time.

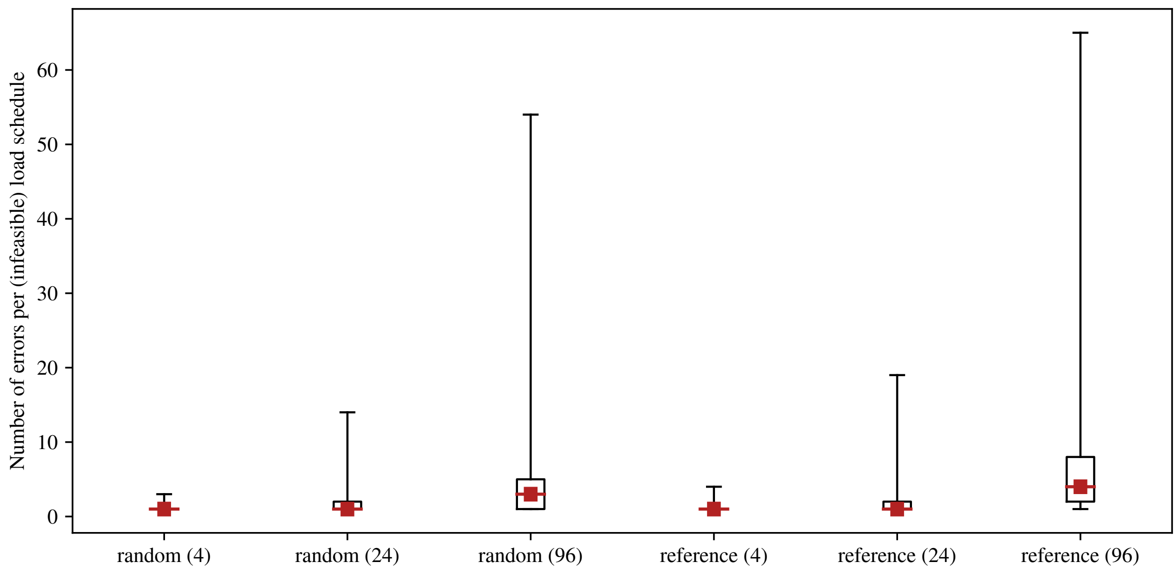

Figure 5.27: Number of erroneous actions per schedules infeasible for the HoLL. No buffers have been applied during schedule generation.

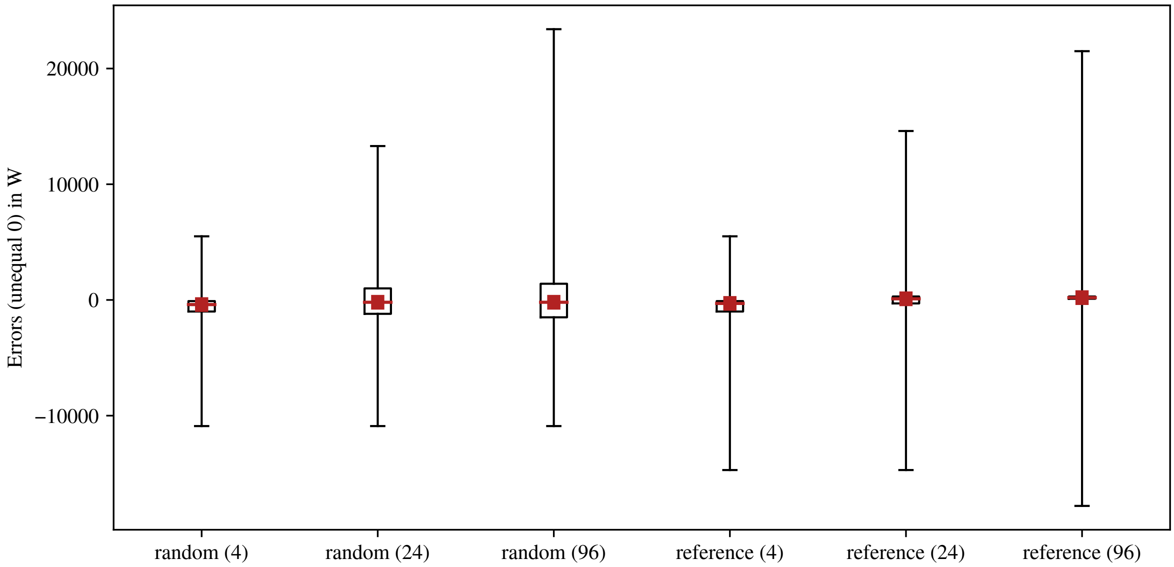

Figure 5.28: Errors (not equal to 0) of schedules infeasible for the HoLL. Actions this configuration range from -7.9 kW to 24.4 kW. No buffers have been applied during schedule generation.

the random strategy after 4 and 96 time steps, respectively. In case of the reference-based strategy, the interquartile range shrinks. The quartiles are at -1 kW and -0.1 kW, as well as 0.1 kW and 0.3 kW. Therefore, while there are more errors in the reference-based strategy, they tend to be smaller. However, the 0.01 and 0.99-quantiles after 96 time steps are -5.3 kW and 5.3 kW, as well as -4.5 kW and 5.8 kW for both strategies, which indicates that the biggest errors are comparable in size.

Aggregated Battery Energy Storage System

In total, the 100 aggregated BESS are able to provide or consume up to 6,250 kW. The exact bounds vary as the individual SOCs change. Like before, buffers for the BESS were applied during schedule generation by setting the associated input variables. In this experiment all batteries shared the same SOC limits. The action space has been approximated with a set of 401 discrete actions, resulting in a distance of 31.25 kW between two neighboring actions. Table 5.16 presents the results for the aggregated BESS.

With an accuracy of 99.89%, the classifier performed worse than the HoLL classifier, but better than the detached family home classifier. At 401 actions, 0.11% of false predictions means one misclassification every second to third step. The observed FPR of 0.2881% is much higher than in the previous configurations and a major factor contributing to the lower shares of feasible schedules. After only 4 time steps 98.02% and 61.95% of schedules were feasible, depending on how actions were selected. At 96 time steps, 45.55% and 0.07% remained. The reason for the better results of the random strategy was the same as in the BESS configuration, that is, the tendency of avoiding boundary regions. Figure 5.21 illustrates the progression over time in more detail. For the reference-based strategy, the graphs show major dips in the first few steps. A look into some state vectors predicted during the generation

Table 5.16: Evaluation results for the 100 aggregated batteries. The single time step metrics relate to a total of $401 \cdot 100,000$ classifications, performed in 100,000 ANN evaluations. For each combination of strategy and buffer size 10,000 schedules have been generated.

			Single time step			
	True positive	True negative	False positive	False negative	FPR	FNR
	82.9439%	16.9488%	0.0490%	0.0584%	0.2881%	0.0703%
			Multiple time steps			
	Purely random			Random reference		
	4 steps	24 steps	96 steps	4 steps	24 steps	96 steps
Feasible	98.02%	84.02%	45.55%	61.95%	9.31%	0.07%
Feasible (MILP)	99.78%	98.56%	82.03%	82.63%	42.32%	7.34%
MAE*	45021	9308	4811	109227	59798	59327
RMSE*	133098	54335	46078	236364	154788	160741
			Multiple time steps with a 2% buffer			
Feasible	99.70%	98.77%	91.05%	80.44%	53.95%	11.15%
Feasible (MILP)	100.00%	100.00%	98.23%	97.23%	73.12%	25.07%
MAE*	40104	4838	2497	52415	30526	30438
RMSE*	92702	29401	29579	122553	97134	106172
			Multiple time steps with a 5% buffer			
Feasible				99.88%	98.32%	58.72%
Feasible (MILP)				100.00%	99.04%	68.59%
MAE*				42969	9386	11557
RMSE*				96108	37580	61451

*only infeasible schedules (according simulation)

process revealed that the simulated and expected states diverge quickly in infeasible schedules. It is conceivable that the ANNs performed badly in some parts of the state space and well on others. While the share of feasible schedules grows considerably when evaluated with the MILP, it remained at only 7.34% for the reference-based strategy. In contrast to the previous configurations, buffers of 2% and 5% instead of 1% and 2% were applied. The feasibility over time with buffers is depicted in Figure 5.30. With a 2% buffer 91.05% and 11.15% of the generated schedules were valid. The MILP was able to reproduce 98.23% and 25.07%, respectively. At 4 time steps it were 99.7% and 80.44% when evaluated with the simulation model, as well as 100% and 97.23% when evaluated with the MILP. In an effort to see if the share of feasible schedules with length 96 can be raised over 50%, a buffer of 5% has been applied, which resulted in 58.72% for the reference-based strategy.

Figure 5.31 and Figure 5.33 show the number of errors without buffers and how they are distributed. The 0.75-quartiles are at the values 1, 2, 8, and 27. Noteworthy is the comparatively high 0.25-quartile of 14 errors for the

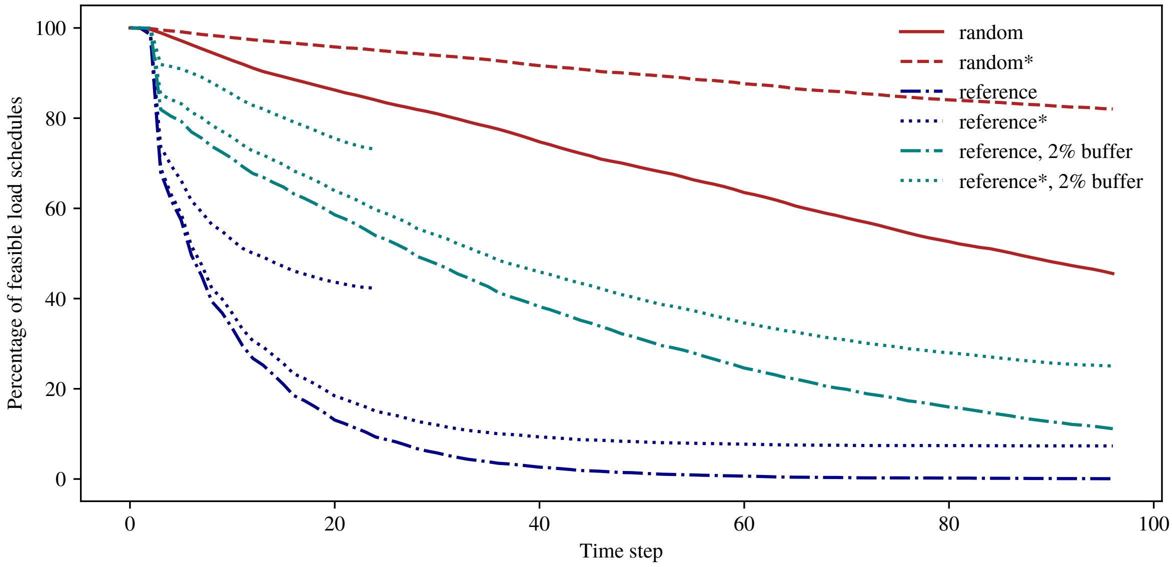

Figure 5.29: Feasibility of aggregated BESS schedules over time.

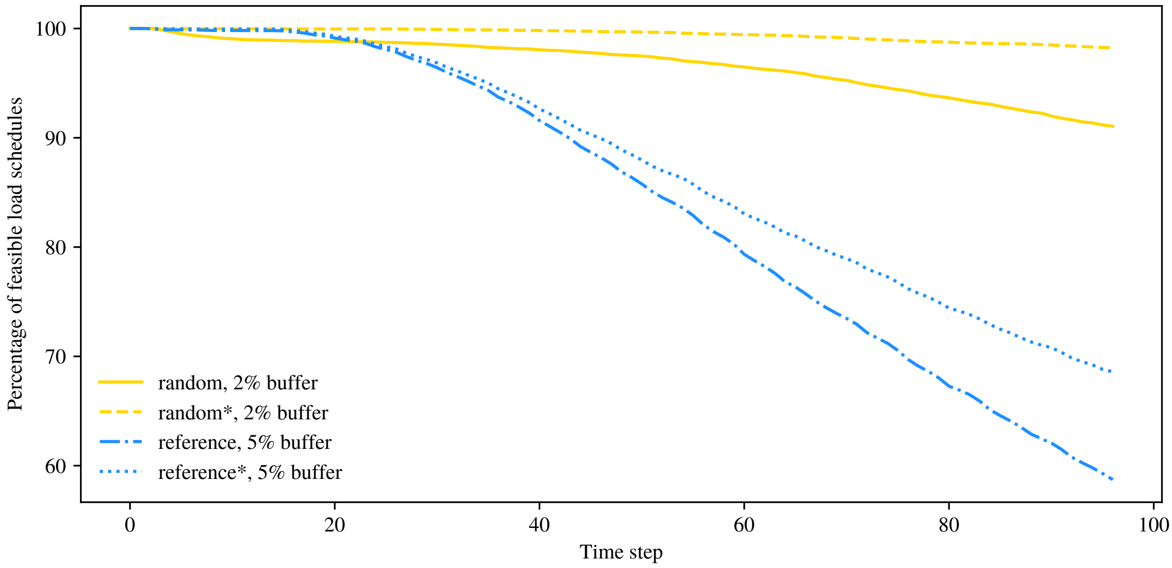

Figure 5.30: Feasibility of aggregated BESS schedules over time.

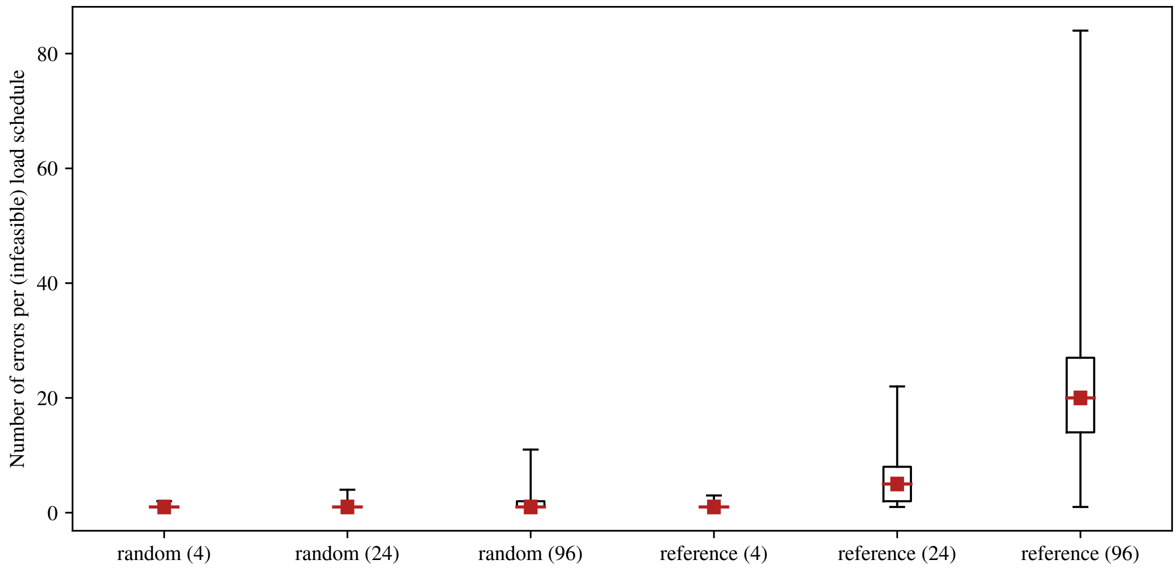

Figure 5.31: Number of erroneous actions per schedules infeasible for the aggregated BESS schedule. No buffers have been applied during schedule generation.

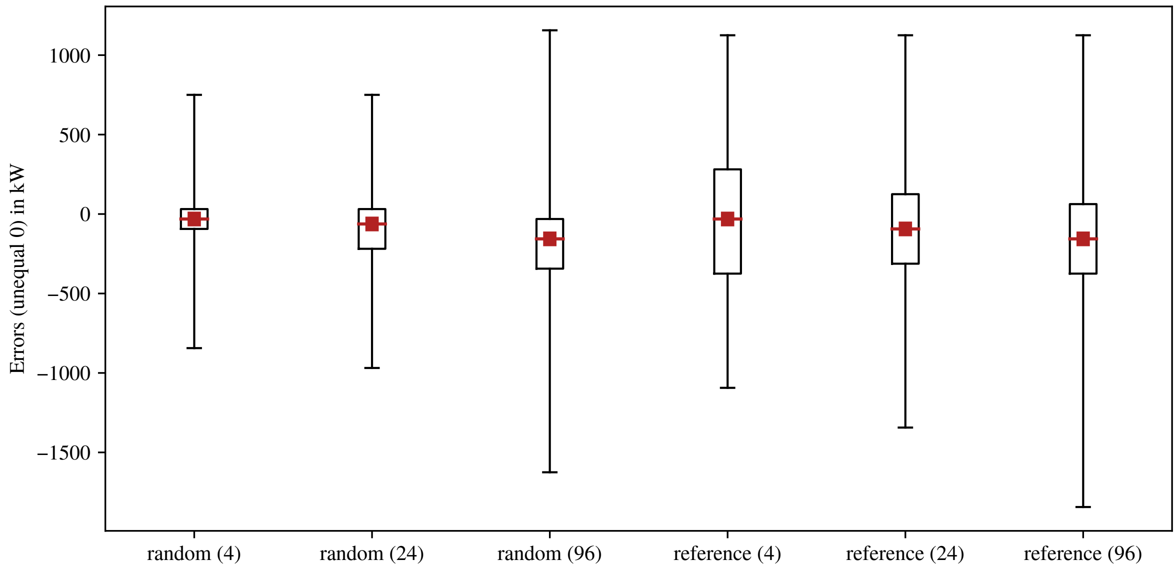

Figure 5.32: Errors (not equal to 0) of infeasible aggregated BESS schedules.

Figure 5.33: Errors (not equal to 0) of schedules infeasible for the aggregated BESS. Actions this configuration range from -6,250 kW to 6,250 kW. No buffers have been applied during schedule generation.

reference-based strategy at 96 time steps. The maximum numbers are 11 and 84 at schedule length 96. Of all the investigated configurations, the surrogate for the aggregated BESS generated the most faulty load schedules. The errors themselves ranged from around -1,8 kW to 1,2 kW, that is, almost 30% of the maximum charge and discharge power. The quartiles for schedule length 96 were at -343.8 kW and -31.3 kW, as well as -375 kW and 62.5 kW.

5.3.7 Discussion

In order to test how well ANN-based surrogates can approximate the flexibility of DERs, the goal of generating feasible load schedules had been selected. Throughout the different experiments the achieved share of valid schedules varied widely. Generating load schedules that consist of 4 steps yielded shares of 93% (EVSE) and higher, except for the 62% for the aggregated BESS. With the help of buffers the share can easily be pushed to around 99%. As the number of time steps increases, the number of feasible schedules decreases. How fast the share declines varies from configuration to configuration and with buffer size. The results ranged from 0.07% (aggregated BESS) to 100% (e.g., BESS with buffer) at 96 time steps. As was to be expected, the individual DERs were approximated much better than the aggregated configurations. With more DERs, there are more state variables and more complex dynamics to be learned, resulting from the interplay of different DERs. Additionally, if multiple DERs are present, the surrogate must learn how the requested load is distributed to the devices. For the configurations with only one DER and without buffers, 57% (EVSE) to almost 100% (BESS) feasible schedules of length 96 were achieved. At least 47% and 42% were reached with the two building configurations. Here again, buffers showed to massively improve the numbers, pushing them towards 100% in the single DER configurations, and at least raising them over 50% for the aggregating configurations. Even when no buffers are used, the majority of errors found in infeasible schedules were within a few percent of the maximum and minimum achievable power and, therefore, relatively small. Furthermore, most time steps of infeasible schedules were feasible. The results show that the strategy used for selecting actions in the state-based simulation approach can have massive consequences on the attained values, depending on the configuration and time horizon. Generally the purely random selection, which was the only one we considered in **[308]**, performs better, since boundary regions tend to be avoided when storages are involved. In contrast, the reference-based selection has been designed to decrease the volatility of the schedule and provide more realistic requests. The next step for exploring more realistic strategies would be to implement a DSM test scenario with simulated flexibility providers.

Aggregated systems were not only evaluated against their simulation model, but also the associated MILP. The MILP, being more flexible in its use of DERs, is able to reproduce more of the generated schedules. Therefore, a higher share of schedules is feasible in this second evaluation. In some configurations the result improved dramatically, for instance, up to the factor 10 in the aggregated BESS. Mostly, however, a few percent points are added. Schedules only reproducible by the MILP, but not by the simulation model, are feasible from a general perspective, but only if the desired control logic, which is implemented in the simulation model and defines the priority of the DERs, is violated. As a consequence, the implementation of such a schedule can come with extra costs. Whether this is acceptable or not depends on the specific application. Regardless, as the goal was to learn the flexibility from the simulation model, it still must be considered as being inadequate.

Given the good performance in generating short schedules of around 4 time steps, buffers are not necessarily required. While time steps of 15 minutes have been used in the experiments, the results can be expected to generalize well to other, comparable time step lengths. Requests with short-lived signals are especially relevant when the flexibility is needed to react to an event, such as a grid congestion, where it is usually sufficient to cover the time required for fixing the congestion and some additional periods for rebound effects to take place.

Hence, judging from the ability to create feasible schedules, the state-based simulation surrogates would be very suitable for generating such short-lived signals. In contrast, when optimizing the operation of DERs, a longer time horizon has to be considered. As the results show, in case schedules are generated only once a day, the likelihood of sending infeasible schedules to flexibility providers is rather high. However, usually in an optimization with random influences, schedules are updated periodically in order to react to recent events. MPC would be viable with surrogates, as at most the first time steps, often just the first step itself, are implemented. Nevertheless, the share of feasible schedules at the end of the time horizon has direct impact on the ability to identify good schedules. A surrogate which only produces few feasible schedules causes huge uncertainty for the optimizer. If the share of feasible schedules at the end of the planning horizon is high, the optimizer can be more confident that the planned path is achievable. Overall, if it is crucial that a flexibility provider is able to exactly follow the signals, buffers should be used.

Even though high shares of feasible schedules have been reached in most of the experiments, these numbers alone are not sufficient to evaluate how well the ANNs learned the flexibility offered by the simulation models. Instead, they only show how well the ANNs encode a fraction of the flexibility. The simulation model being able to reproduce the generated schedules is important for planning. However, a surrogate yielding the identical feasible schedule over and over again would easily achieve a share of 100%, even though the output does not show any flexibility. Only combined with the inverse experiment, that is, generating schedules from the simulation model and evaluating them with the surrogates, a conclusion is possible. Due to the versatility of the state-based simulation approach, this kind of test can easily be implemented. Table 5.17 provides the classification results for 100,000 schedules in each configuration. Of the 100,000 schedules, 50,000 were sampled as vectors of random numbers within the range of possible actions and evaluated with the simulation model in order to assign a label, 25,000 were generated with the random strategy, and the remaining 25,000 were generated with the reference-based strategy. This method results in at least 50% feasible schedules, as both strategies only choose valid actions. A random vector of numbers is highly unlikely to be a reproducible schedule. Nevertheless, for the configurations with battery storages some of them were valid. The classification task would be more challenging by producing schedules were only a single action is invalid, which could easily be implemented with the developed models. However, the presented partition into 50% random and 50% feasible schedules has already been used in [11] and is used here in order to allow the comparison of results. All configurations have been tested with buffers. For obtaining schedules with 4 and 24 time steps, the first 4 and 24 values of the full schedules have been extracted.

Table 5.17: Classification results using the state-based simulation approach and buffers.

	4 steps	24 steps	96 steps
BESS, 2% buffer	99.98%	99.54%	96.46%
EVSE, 2% buffer	98.62%	96.28%	90.85%
CHPP satisfying heat demand, 2 °C buffer	99.95%	99.69%	96.19%
Detached family home, 2% and 2 °C buffer	99.69%	97.84%	85.03%
HoLL, 2% and 2 °C buffer	99.22%	95.09%	74.64%
Aggregated BESS, 5% buffer	98.28%	78.09%	58.67%

The numbers show that the configuration with single DERs are performing best. The associated surrogates are mostly able to correctly identify whether the given schedules are valid or not. For the aggregated configurations, the results vary. While the detached family home reaches 85%, the aggregated BESS falls short of 59%. For comparison, a classifier randomly assigning labels is expected to reach 50%. Not listed here is the type of error. Misclassified schedules are mostly false negatives. Mostly there is not a single false positive. The highest numbers of false positives are 200 for the detached family home and 295 for the aggregated BESS, both at time step 4. At time step 96, the only configuration with false positives is the BESS, with a total of three. For the aggregated BESS, only around 20% of the feasible schedules of length 96 have been recognized as such. Infeasible ones, on the other hand, have all been identified correctly. In summary, the extremely low number of false positives further confirms that the surrogates have learned at least portions of the DERs dynamics. The numbers show that the flexibility is far from being captured completely by the surrogates in the long run. For the most complex configurations the numbers indicate that less than half of the achievable schedule space is covered. However, for short periods of time the surrogates are able to represent the DERs very accurately. In order to obtain a more complete representation of the flexibility and improve the result for long periods, the low number of false positives suggests that schedules should not be filtered as strictly. Two possible parameters for adjustment are the threshold of the classifier, and the fit of the ANNs. The classification threshold has been set to 0.5. An action with a rating of 0.5 or higher is deemed valid. By reducing this threshold, more schedules will be classified feasible by the surrogate. The ANN fit has an influence as well, as an overfitted model can cause exactly the effects observed here. A more strict early stopping—a window of 100 training epochs was granted before stopping early—is one possible way to obtain better generalizing ANNs. An added benefit would be the reduction of the required training time. However, both comes at the cost of more false positives and a reduced share of feasible schedules generated with the help of the surrogate.

Another important aspect to consider is that all results have been generated on the basis of simulation models. While the simulation models are designed to mimic real systems, they are only approximations themselves. In order to show that the generated results in this thesis also apply for real DERs, further tests with load profiles measured from existing hardware have been conducted. For three of the six configurations, namely the BESS, the CHPP satisfying heat demand, and the HoLL, data was available. However, the HoLL data was incomplete, as it lacked measurements for the EVSE and BESS. Therefore, only the BESS and the CHPP configurations could be evaluated. Table 5.18 presents the classification results. The BESS data originates from the research project *grid-control* **[310]**. It comprises 666 load schedules with 96 consecutive time steps of 15 minutes each. About one fourth of these schedules consists of only zero values. The associated BESS has a capacity of 9 kWh and can charge or discharge with up to 4.6 kW. Using the measured load and the recorded SOC values, the losses due to efficiency and self-discharge have been estimated and the simulation has been parameterized suitably. The charging efficiency has been set to 83%, and the relative self-discharge per hour was parameterized with 0.0075. Then, with the help of the simulation model, another surrogate has been trained. Both, the simulation model and the surrogate, were able to reproduce all 666 schedules. The data for the CHPP configuration has been taken from the ESHL at the KIT in Karlsruhe. The dataset contains 197 schedules and was tested with the surrogate used in the evaluation of the CHPP configuration. Out of these, only 185 could be reproduced by the simulation model. The surrogate achieved the identical result. Overall, these results show that the models are generally suitable to describe these two configurations.

Table 5.18: Classification results using the state-based approach and measured load profiles from real devices.

	Real schedules	Feasible (simulation)	Feasible
BESS	666	666	666
CHPP satisfying heat demand	197	185	185

During the development and experiments it became apparent that the major challenge for generating load schedules is the precision of the estimated states. As the presented numbers show, the classifier ANNs are very well capable of identifying feasible actions in a given state. The shares of false positives and false negatives are mostly well below 0.1%. Only once, with a rate of 0.45% a higher number was reached. The state estimates, while mostly being precise, are sometimes extremely inaccurate. Such state errors are larger in the more complex systems. While, for instance, in 100,000 tested transitions, the SOC error of the BESS surrogate was around 0.12 percent points at most, the SOC error produced by the EVSE surrogate reached up to 4.8 percent points. Since errors carry over to subsequent time steps, they accumulate and the true state diverges from the predicted state. Hence, with increasing schedule length, it becomes more and more important to reduce such errors. Sometimes, even trivial logic relationships, such as the minimum dwell times and the progression of the current dwell time, are not learned correctly. Training success is often strongly depending on the hyperparameters of the training process. With unsuitable hyperparameters, the resulting ANNs perform very poorly. In contrast to our previous work **[312, 308]**, where hyperparameters and the ANN topology were chosen by hand, a random search has been conducted in this thesis. More advanced methods for the ANN topology design and hyperparameter optimization like CMA-ES, could further improve the results by yielding better ANNs.

There are many parameters having an impact on the process of generating and exploiting surrogates. Aside from the various hyperparameters, the amount of available data, variables in the state vectors, size of the action set, and training algorithms all influence the training success. Naturally, learning surrogates for modeling flexibility with the help of ANNs benefits from better training algorithms. In the future, new algorithms may reduce the amounts of required data and the difficulties of learning logics. Regarding the level of detail of the approximation, there is a trade-off between complexity and detailedness. Assuming well performing ANNs, the set of possible actions defines which power levels are generally attainable. If the set is coarser, there are fewer possible inputs and outputs for the ANNs and the external entity must decide between fewer options. Therefore, the complexity is reduced in every single step, from model training to the exploitation itself. At the same time, with fewer actions to choose from, the DERs are represented less exactly. A larger action set, on the other hand, is only sensible if the surrogates are accurate, as the performance may drop even further. Aside from the actions, the remaining elements in the input and output vectors influence the training process. In general, there should only be data in the vectors which is truly required. If unnecessary or redundant elements are present, the ANN may learn spurious correlations, i.e., relationships which do not exist in reality, or may have trouble to learn correlations correctly. As mentioned before, in this thesis, the transition models were trained to provide an additional output specifying which action has been implemented. The idea behind this attempt was that if the resulting action is predicted close enough, there is no need to use an additional classifier ANN. Instead of using a classifier, the whole set of possible actions could be passed in the form of a batch, and if the resulting action and requested action are close enough, an action would be considered valid. While simple models performed well, for instance, the BESS surrogate being off 300 W at most, complex models did not. In case of the aggregated BESS, the difference between expected result and requested action was mostly 0, but reached prediction errors of up to 9,719 kW. Overall, the additional classifier ANNs performed better. Not learning the performed action in the transition model will most likely have an influence on the ANNs performance. It is, however, unclear if positive or negative. Another possibility for improving the long-term performance could be the unrolling of the ANN. The models have been trained by computing a single transition. When unrolling is applied, multiple transitions are computed in sequence and the loss is calculated from the resulting state.

One parameter not yet discussed is the length of a time step. If time steps are very short, the individual elements in the state vector will usually not change by much and only few events, which may distort the resulting state, can happen. The longer the time step becomes, the more effects may have an influence on the outcome. This means that when starting in the identical state, one and the same system may produce different follow-up states even though the same average power was consumed or provided. In case of the simulation model and the algorithms used here, this is not an issue, as the parameter is simply adapted in the algorithms and every formula remains exactly the same. When historic data is used instead, it is not as simple. To see this, take a BESS, for instance. The action 0 would be equivalent to doing nothing. If it keeps idling, the SOC does only change marginally. The same outcome, which is measured as average power, can be achieved by charging with 1 kW for half the time and then discharging with 1 kW the remaining time. While the average power is identical, the resulting state is not. Due to charging and discharging losses, the resulting SOC is lower this time. With these two examples, there are two data points for the identical action and state, and both produce different outcomes. As the MSE loss, which is used in this thesis, implicitly assumes a normal distribution, the ANN would tend to predict the average of the observed values. One option to deal with this would be the training of stochastic models. The stochastic information could also be used by the external entity. Another option is to filter such occurrences from the historic data, and search for periods where the power level remains stable. Nonetheless, this would result in even fewer suitable data, especially when time steps are long. Overall, when using longer time steps new problems may arise, but, at the same time, longer periods of time can be covered with fewer time steps which greatly improves the approximation quality.

Results in Comparison With Other Publications

Concluding this discussion and chapter is a comparison of the results achieved in this thesis and results found in the related literature. Aside from the ANN-based approaches we proposed in **[312]** and **[308]**, the literature presents surrogate modeling approaches based on SVDDs [63, 11, 65], Chi-shapes [69], and cascaded classifiers [12, 67, 68]. Subject of all of these publications is the encoding of the flexibility of DERs into learned models.

ANN-based approaches: In comparison to a dedicated classifier **[312]**, the state-based simulation surrogates perform worse. Accuracy and F_1-score of the BESS, CHPP, and detached family home configurations are listed in Table 5.19. The inferior performance is caused by the comparatively high number of false negatives. Possible measures for decreasing the number of false negatives and potentially increasing the scores have already been discussed. While the ANNs of the classifier approach seem to perform very well, it is unclear how they should be used to derive feasible load schedules. One possible solution to this question is proposed in Section 6.2.

Table 5.19: Comparison of old and new ANN-based classification results given by **[312]**.

	Classification with a dedicated classifier **[312]**		Classification using the state-based simulation (this thesis)	
	F_1-score	accuracy	F_1-score	accuracy
BESS	0.995	0.995	0.970	0.965
CHPP	0.993	0.995	0.960	0.962
Detached family home	0.972	0.974	0.824	0.850

Table 5.20 presents the best MAE values and their associated RMSE values achieved by any surrogate modeling approach in **[312]**. The BESS and CHPP values were produced with the generation approach, and the results for the aggregate of both by the repair approach. Consistently, the average error produced by the state-based simulation is

much smaller. While it is not possible to tell how well the generation and repair approaches cover the true flexibility, the classification results showed that the state-based simulation captures most of it in the considered configurations. However, the average errors may increase when the coverage is improved.

Table 5.20: Comparison of new and old ANN-based results. Listed are the best MAE and their corresponding RMSE values found in **[312]**, and the values generated with the random strategy using the state-based simulation approach.

	Generation and repair*		State-based simulation	
	MAE	RMSE	MAE	RMSE
BESS	1.8	4.7	0.002	0.020
CHPP	52.5	140.2	11.834	60.287
Detached family home	133.1	211.8	6.683	58.986

* best result achieved in **[312]**, regardless of approach

A first evaluation of the state-based simulation has been presented in **[308]**. The results are given in Table 5.21. For the BESS, the performance has improved, likely due to better performing neural networks. In our previous work, the hyperparameters have been selected by hand. The newly introduced hyperparameter search may have played a crucial role in this improvement. As we predicted in **[308]**, the introduction of a buffer for the BESS further improved the result. The numbers for the CHPP configuration and the detached family home are only roughly comparable, since the CHPP model and the action set have been adapted. In **[308]** the CHPP configuration could run into "dead-end" states, where no actions are valid, for instance, if the CHPP must run because of the minimum dwell time, but at the same time must stop because of a high HWT temperature. Instead of solving this by using buffers, in this thesis, the dwell time constraint is a soft constraint. Whenever the HWT temperature exceeds the bounds, the CHPP is free to ignore the dwell time. This has been implemented by adding further variables and logic to the CHPP and HWT models, which results in a higher model complexity. Moreover, the heat demand time series are more diverse in this thesis, as $3 \cdot 60$ instead of only 3 different series were passed into the generation process. The remaining dynamics are the same. Both CHPP configurations, the old one and the new one, produced very similar shares of feasible schedules. At the same time, the performance for the detached family home worsened considerably. Aside from the changed operation logic of the CHPP and more diverse input data, the main reason is most likely the changed action set. In **[308]** the degree of freedom present when implementing a given power level is passed to the external entity. If, for instance, the BESS has 201 possible actions and the CHPP has 2 possible actions, the external entity would select from a set of $2 \cdot 201 = 402$ actions. With a higher number of DERs or actions this approach leads to a very high number of possible actions. For this reason, in this thesis, the degree of freedom was eliminated. The surrogate not being able to follow this elimination logic close enough causes a drop in the share of feasible schedules. In contrast to the simulation model and the surrogate, the MILP is not bound to this logic, which is the reason for the improved MILP results given in parentheses.

Support vector data description: In most publications using SVDDs for encoding the flexibility of DERs, for instance, [63, 64, 286, 65, 287], the evaluation is based on the aggregated schedule achieved with the help of a surrogate-based coordination algorithm. In these coordination algorithms, the SVDD is used for transforming infeasible load schedules into feasible schedules. Schedule repair is in most cases performed by mapping an infeasible schedule into the latent feature space, where it is projected onto the SVDD sphere in order to make it feasible, and then calculating an approximate preimage within the schedule space. As every point on and inside

Table 5.21: Comparison of new and old state-based simulation results. Listed are the shares of feasible schedules given in **[308]**, and the values achieved with the random strategy using the state-based simulation approach. The shares evaluated by the MILP are given in parentheses.

	prior work **[308]**		this thesis	
Buffer	none	2 °C	none	2% and 2 °C
BESS	98.30%	-	99.61%	100%
CHPP*	95.30%	99.80%	90.08%	100%
Detached family home*,**	96.00%	99.10%	81.57% (90.31%)	90.03% (99.67%)

* CHPP restrictions in **[308]** are less complex

** actions in **[308]** are easier to disaggregate

the sphere is feasible in regard to the SVDD approximation, the infeasible schedule is turned feasible. For more information on SVDD and this mechanism, please see Section 2.4.5 and [11]. Since the performance of the repair mechanism and the coordination algorithm are intertwined, it is not possible to draw direct conclusions about the SVDD-based approximation performance. Fortunately, a dedicated analysis of the SVDD performance has been presented in [11]. The results are given in Table 5.22 and Table 5.23.

Table 5.22: Comparison of the classification performance using the state-based simulation and SVDD. All SVDD numbers are taken from [11].

	SVDD		ANN-based (no buffers)		
	32 time steps	96 time steps	4 time steps	32 time steps	96 time steps
Electric boiler	98.19%	96.94%	-	-	-
CHPP*	59.65%	99.38%	98.51%	88.77%	68.03%
EVSE	-	-	98.57%	95.67%	90.66%
Heat pump	86.82%	98.51%	-	-	-
BESS	95.32%**	97.29%**	99.96%	98.85%	95.38%

* SVDD for modulating CHPP and ANN for non-modulating CHPP

** Positive and negative labels interchanged to achieve better results

Bremer [11] presents a very thorough analysis of the SVDD models for an electric boiler, a modulating CHPP, a heat pump, and a BESS. A modulating CHPP, in contrast to the non-modulating CHPP modeled in this thesis, offers more than only an on- and off-switch. Instead, the power can be controlled within an interval, which makes the model more similar to the EVSE model, just with a reversed power flow. Using non-modulating CHPPs in our papers and this thesis has practical reasons, as the CHPP installed in the HoLL and ESHL are non-modulating as well. Table 5.22 lists the classification performance achieved when the models are used as classifiers. For schedules of 32 time steps, the ANN-based surrogates performed better. At 96 time steps, the SVDD showed better performance. Even though both CHPPs are not directly comparable, due to the similarity to the EVSE model, it can be assumed that a surrogate of the state-based simulation approach would have performed better for a modulating CHPP than for the non-modulating CHPP. Regarding the BESS it must be noted that in order to achieve the listed SVDD accuracy, it was necessary to interchange the positive and negative classification label. Without this additional step, around 70% could be reached [11, p. 164].

The only numbers for the reproducibility of schedules generated with the SVDD surrogates provided by Bremer are for the modulating CHPP and schedules consisting of 8 time steps. Since all values were given in the form of a graph, Table 5.23 provides only approximate data points, which have been read from the graph. Depending on the SVDD hyperparameter σ the results vary strongly. The smaller σ, the more similar an input and a support vector need to be in order to reach a positive classification result. This means that with a very small σ the SVDD repair mechanism approximately returns one of the support vectors, which are known to be feasible. Consequently, the share of feasible schedules is high, but the diversity may be low, depending on the support vectors. Nevertheless, the classification performance at $\sigma = 0.1$ is high, which indicates that the available flexibility is captured well with this parameter choice. In comparison to the 97% classification performance and 100% share of feasible schedules reached by the SVDD, the state-based simulation reached 91.82% and 98.95% for the non-modulating CHPP, as well as 96.1% and 79.4% for the EVSE. The performance for a modulating CHPP would probably lie somewhere in-between. It must be noted that the ANN-based numbers are for schedules of length 24. As the accuracy and share of feasible schedules generally decreases the longer the schedules become, the results at 8 time steps would be even better. Furthermore, for generating the listed numbers, no buffers were used during the evaluation.

Table 5.23: Comparison of the state-based simulation and SVDD-based surrogate. All SVDD numbers are taken from [11]

	SVDD, CHPP*, 8x15 minutes				ANN-based (no buffers), 24x15 minutes	
	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	CHPP*	EVSE
Schedule classification	97%	93%	81%	88%	91.82%	96.14%
Feasible schedules generated	100%	89%	33%	17%	98.95%	79.42%

* SVDD for modulating CHPP and ANN for non-modulating CHPP

Schedules with 96 time steps for modulating CHPPs are evaluated in [69]. The authors achieve feasible rates of 94.34% using a small and 75.21% using a large thermal storage. The worst CHPP and EVSE results achieved in this thesis are 89.05% and 57.10%. A modulating CHPP can again be expected to lie somewhere within this range. Overall, the state-based simulation and SVDD approach show similar performances. While the ANNs tend to outperform the SVDD in short time horizons, the SVDD is slightly ahead for longer schedules. However, a more thorough comparison with more types of DERs and also identical reference models is required in order to reach a final conclusion.

Chi-shapes: Chi-shapes are proposed as an alternative to the SVDD-based description in [69]. A chi-shape is basically a concave hull around a set of points described in the form of a polygon [69]. In contrast to the SVDD-based approach where the schedule is transformed in the latent feature space, computations for transforming a schedule from the infeasible to the feasible region are performed directly in the schedule space. However, in the experiments, the chi-shape approach was outperformed by the SVDD approach. In case of the two modulating CHPPs mentioned above, the chi-shape approach reached 34.2% and 74.5% feasible schedules instead of the 94.34% and 75.21% achieved by the SVDD.

Cascade classification: The last approach in this comparison is the cascade classification introduced in [12]. In this approach the feasibility of a schedule is judged with the help of a set of classifiers instead of a single one. This idea is closely related to the use of ANN-based classifiers and the fragmented classification approach derived in this chapter. Aside from artificial "hypersphere" and "hyper-banana" test sets, schedules from the same modulating CHPP used in the SVDD evaluations are considered in [12, 67, 66] and [68]. The modulating CHPP is the only DER considered. All relevant results in [12, 67, 66] and [68] are given in the form of graphs for the TP and TN rates plotted against the training set size. As some graphs lack detailed markings, the values listed here are by no means exact. All schedules cover 96 time steps. In [12], the cascade approach reaches a TP rate of around 99% and a TN rate near 100%. Individual CHPPs are also considered in [66] and [68]. Both papers present the identical graphs, depicting a TP rate of around 99%. The TN rate is 100%. An ensemble of 5 to 50 CHPPs is considered in [67]. For this aggregated setting a TP rate of around 99.9% and a TN rate of 100% are reached. All cascade classifiers have been trained with 15.000 or fewer samples to reach these numbers. The ANNs are trained with considerably more data. To put these classification rates into context, the state-based simulation reaches TP and TN rates of 36.1% and 100% for the non-modulating CHPP without buffers. With a 2% buffer the TP rate reaches 92.4%. In case of the EVSE, the TN rate is 100%, too. Without buffers, the TP rate is 81.3% and with a buffer of 2% it is 81.7%. Given these numbers, it is apparent that the cascade classifier exhibits superior classification performance. However, neither is the state-based simulation surrogate trained to be a classifier, nor has schedule generation been tested for the cascade classifier.

5.4 Summary

Surrogate models have the potential to serve as generic descriptions of the flexibility of DERs, which is required in order to control the associated DERs' operation. Furthermore, surrogates may be learned in an automated process, potentially without the need of manual modeling. Depending on the information processed and provided by a surrogate, it may either remain valid for extended periods of time or needs to be generated newly for each individual exploitation attempt. In this chapter, several approaches for encoding the flexibility of DERs into surrogate models have been illustrated, of which none is exclusive to ANNs. The identified approaches either serve the generation of entire load schedules likely to be feasible or the selection of a use case specific signal. Due to its versatility, the state-based simulation has been selected in order to establish a baseline for the capabilities of ANN-based surrogate modeling of the flexibility of DERs.

With the state-based simulation approach high shares of feasible load schedules could be achieved. Especially short schedules are very likely to be feasible and capture the underlying flexibility well. However, for long schedules and complex systems, the quality of approximation declines rather quickly. Not only is the share of feasible schedules falling, but also the coverage of the true flexibility decreasing. Buffers in the form of artificial constraints during the training and relaxed constraints during the evaluation help to counteract the approximation errors and improve the results. In order to cover more of the true flexibility, the number of false negatives must be decreased. This likely results in a decreased share of feasible schedules. Some possible options, like early stopping of the training process, have been outlined in the discussion.

In comparison with our previous work, the results in this thesis are the best ones achieved yet. Furthermore, the set of actions has been simplified, which is now handled much easier by the external entity, as the number of options decreased drastically. The SVDD-based approaches perform similarly well, judging from the classification results and feasible schedule rates. While the state-based simulation performs better in the short-term, the SVDD does better in the long-term. However, these rates alone have limited explanatory power concerning the diversity and usefulness of the generated schedules. Such a comparison should be conducted based on a practical use case in the future. In the role of a classifier, the cascade classifier is superior to the state-based simulation. However, the cascade classifier has not yet been used to generate load schedules. Finally, while all alternative surrogates can be trained with much fewer data, the ANN-based approaches are the only ones that do not have to be newly generated each time a description of the flexibility is required. In other words, while the ANN-based surrogate uses the identical ANNs in each DER state, the SVDD, chi-shape, and cascade classifier surrogates have to be generated again for each new DER state. Hence, for a test with 100.000 schedules starting each in a different state, like the classification test presented in this discussion, 100.000 surrogates have to be generated with up to 15.000 training samples each. The total number of 1.500.000.000 schedules surpasses the several million training samples of the ANNs by far. Considering this specialization of the alternative surrogate modeling approaches for a given state, the ANN performance across different states becomes far more impressive.

6 Controlling Distributed Energy Resources Using Artificial-Neural-Network-Based Surrogate Models

With the objective of reproducing the space of feasible load schedules, the previous chapter established a baseline for the quality of approximation achievable with ANN-based surrogate modeling. While the trained ANNs may perform very well in the task of forecasting some particular information, an external entity still needs to be able to make use of it. The primary goal of this chapter is to assess the ANN-based surrogate modeling approaches presented in Table 5.1 regarding their usefulness for deriving control signals, either for influencing the general behavior or directly controlling the operation. The central research question of this chapter is **RQ3** "How adequate are the different ANN-based modeling approaches for controlling schedules of flexibility providers?". This question is, like the others, divided into multiple subquestions. The chapter is structured as follows.

RQ3.1 Which criteria can be used for assessing the adequacy of a modeling approach?
Section 6.1 collects and selects criteria from the literature and develops a basic framework for assessing the different approaches.

RQ3.2 What is the assessment of the individual ANN-based modeling approaches?
With the help of the defined criteria, a qualitative assessment of the approaches found in Table 5.1 is conducted. In the first step, it is outlined in Section 6.2 how each approach may be used in order to solve exemplary scheduling tasks. The assessment and its results are then presented in Section 6.3. Afterwards, a quantitative evaluation of the state-based simulation approach is presented in Section 6.4

Finally, a summary is provided in Section 6.5.

6.1 Criteria for the Assessment

A set of criteria has to be defined first, in order to be able to compare the different surrogate modeling approaches. Since the approaches do not only provide distinct information, but even serve different general purposes, like the generation of feasible load schedules or the generation of price profiles (see Table 5.1), the assessment needs to resort to general criteria. Furthermore, each approach is more or less loosely tied to a specific exploitation pattern. Therefore, if an exploitation pattern by itself restricts the ability to influence DERs, the surrogate does as well. Another aspect complicating the assessment is the need for additional algorithms. Surrogate models provide approximated data. The algorithms built around them determine how well it is put to use. It is therefore necessary to evaluate each approach in the context of the employed algorithms. For this reason, before any ratings can be derived, it must at least be outlined how an external entity would identify appropriate control signals with each individual approach. A quantitative analysis, furthermore, requires the implementation and testing of said algorithms, which in turn calls for the development of realistic and relevant benchmark environments [288], a topic that only recently started to gain traction for energy related applications (see [288] and [289], for instance). Since a thorough analysis of the different ANN-based surrogate modeling approaches is not possible in the scope of this thesis, this chapter primarily aims at providing a fundament for future investigations of the topic at hand.

Comparative studies of (new) algorithms can be found quite regularly in the literature, where they are often tested against other well known algorithms. Conducting such a comparison is also called benchmarking [290]. As the goal of a benchmarking process is to identify the best algorithms for given optimization problems, it is generally based upon quantitative analyses using some performance measures [288]. Measures used in such an analysis rate, for instance, the algorithms' efficiency, reliability, and solution quality [290]. Efficiency measures include the number of fundamental evaluations, the running time, as well as the memory usage [290, 291]. An exemplary measure for reliability is the number of constraint violations [290]. The quality of a solution can be judged by comparing the time till a given target function value is achieved, or by the function value achieved within a set time frame [290, 292, 291]. Since it would be necessary to implement each single approach for a quantitative analysis, only qualitative judgements can be derived within the scope of this thesis. Out of those measures named above, the efficiency is the most sensible option for which a qualitative analysis can be conducted, since it is possible to (very) roughly estimate the required computational effort for training and using the required ANN models for each approach. Reliability and robustness strongly depend on the trained ANNs' ability to generalize. They can therefore not be estimated. Regarding the solution quality, it is to be expected that simpler approaches, i.e., approaches that reveal fewer options to influence the operation of DERs, are approximated better since all evaluations are faster and fewer operational options have to be considered. Within the boundaries of such a simple approach with less complex ANNs, it is therefore easier to find the best or a near optimal solution. However, due to the more limited options of influencing the DERs, the truly optimal solution may not be achievable in the simplified model. This illustrates the difficulty of comparing surrogate modeling approaches.

The main challenge in identifying qualitative criteria is the strong dependence on the specific use case. In a certain sense the task of comparing surrogate modeling approaches is similar to the task of comparing meta-heuristics themselves. Meta-heuristics, such as evolutionary algorithms and simulated annealing, are black-box optimization strategies [293] (see also Section 2.5). They provide general frameworks for implementing heuristics for arbitrary optimization problems. Nevertheless, when a meta-heuristic is implemented, at least some adaptations are advised in oder to incorporate problem-specific knowledge (see [293]). Wolpert and Macready [293] show with their "no free lunch" theorems that no optimization strategy universally outperforms every other strategy. Similar to the topic of surrogate modeling, there is a dependence on the considered use case and problem-specific adaptations. As a consequence, comparing the achieved results for a given type of problem alone is not sufficient to derive general conclusions. Instead, many problems [293] and, for a given problem, problem instances [294] should be investigated in order to rate each approach quantitatively. The literature proposes a variety of measures to compare meta-heuristics, including the general applicability [295, 296], efficiency [295, 296], effectiveness [296], user-friendliness [295, 296], and many similar ones (see for instance [296]). Most of these criteria are not directly applicable here, since the individual surrogate modeling approaches serve different purposes and the measures are often quantitative. Out of the identified measures, the most fitting criteria for a qualitative analysis of the modeling approaches could be the general applicability, as well as, once again, the (estimated) efficiency in terms of the required computational effort.

A qualitative comparison of software frameworks for meta-heuristics is presented in [297]. However, the assessment is based on implementation specifics, such as the supported stopping criteria and functions for importing and exporting data, and other general features.

In summary, in order to conduct the assessment, the optimization procedures for each approach must be outlined first. The outlined procedures can be seen as templates for future implementations. Once the procedures are defined, the complexity (as opposed to the efficiency) in terms of the computational effort can be rated qualitatively. Since both, flexibility provider and external entity, are involved in this process, individual ratings can be assigned. The complexity rating can be derived by looking at the structure of the associated ANNs, the dimensions of the possible input and output vectors, as well as the size of the search space, which all influence the time required to train a model and to identify possible solutions using a trained model. These parameters are, in turn, strongly dependent on the considered time horizon in time steps. The more time steps are considered, the larger the space of feasible schedules grows. Regarding the number of time steps, it is also important to consider whether the time horizon is fixed or not, and if it is limited by the surrogate. While some approaches can handle varying horizons, others can not. Another aspect relevant to DSM is the scalability of an approach. External entities like aggregators or grid operators usually intend to make use of large numbers of DERs. Even though surrogates have the ability to aggregate many DERs, as the previous chapter shows, there is the need to handle many surrogates in parallel and coordinate the solution finding processes. For this reason, qualitative ratings for the scalability are derived in the context of the outlined optimization procedures and the general applicability is discussed. It is assumed that all data needed for training the ANNs is freely available.

The assessment is concluded with a quantitative analysis of the state-based simulation approach presented in Section 6.4. It shows the performance in comparison to an SVDD-based model and an analysis using the same criteria applied in Chapter 5.

6.2 Optimizing with Artificial Neural Network Based Surrogates

While the basic functionality and usage patterns of the different surrogate modeling approaches are outlined in Section 5.2, the question of how to make the best use out of the data provided by the surrogates has not been addressed yet. For the sake of comprehensibility, it is assumed the following two optimization problems need to be solved by the external entity:

Problem I, following an aggregated target schedule:

$$
\begin{aligned}
\min_{a^i_{0:T-1},\, i=1,2\dots,N} \quad & \frac{1}{2}\left\lVert \bar{a} - \sum_{i=1}^{N} a^i_{0:T-1} \right\rVert_2^2 \\
\text{s.t.} \quad & a^i_{0:T-1} \in \mathbb{M}^i(s^i_0),\, i = 1, 2, \dots, N
\end{aligned}
$$

Number of flexibility providers	N
Length of schedules	T
Target schedule	$\bar{a} \in \mathbb{R}^T$
Schedule of flexibility provider i	$a^i_{0:T-1} = (a^i_0, a^i_1, \dots, a^i_{T-1}) \in \mathbb{R}^T$
Set of feasible schedules in state s^i_0	$\mathbb{M}^i(s^i_0) \subset \mathbb{R}^T$

Problem II, restricting the aggregated load at time $\tau < T$:

$$\begin{aligned} \min_{a^i_{0:T-1}, i=1,2,\dots,N} \quad & \sum_{i=1}^{N} c(a^i_{0:T-1}) \\ \text{s.t.} \quad & \underline{b} \le \sum_{i=1}^{N} a^i_\tau \le \bar{b} \\ & a^i_{0:T-1} \in \mathbb{M}^i(s^i_0), i = 1, 2, \dots, N \end{aligned}$$

Number of flexibility providers	N
Length of schedules	T
Cost of requesting a given schedule	$c(\cdot)$
Boundaries for the aggregated load	$\underline{b}, \bar{b}$
Schedule of flexibility provider i	$a^i_{0:T-1} = (a^i_0, a^i_1, \dots, a^i_{T-1}) \in \mathbb{R}^T$
Set of feasible schedules in state s^i_0	$\mathbb{M}^i(s^i_0) \subset \mathbb{R}^T$

The objective of the first optimization problem is to minimize the distance between the aggregated load of all involved flexibility providers and a given target schedule. This kind of problem has, for instance, been considered in [11]. The second optimization problem minimizes costs while restricting the aggregated load at time step τ to stay within given boundaries. Only a few surrogates discussed in this thesis provide information about costs. It is therefore assumed that flexibility providers and external entities close contracts in which they specify how the providers should be compensated. The cost is then calculated accordingly. Since the true sets of feasible schedules $\mathbb{M}^i(s^i_0)$ are unknown to the external entity, the schedules $a^i_{0:T-1}$ are selected from sets $\hat{\mathbb{M}}^i(s^i_0)$ consisting of elements, i.e., load schedules, sampled with the help of the surrogates. Given that the surrogates are approximate models, some elements of $\hat{\mathbb{M}}^i(s^i_0)$ may lie outside of $\mathbb{M}^i(s^i_0)$. As pointed out earlier in this thesis, if necessary, additional communication steps could be added in the exploitation process to ensure feasibility.

In general, if a surrogate can be used to derive either feasible load schedules or realizable signals for influencing DERs, simply generating a set of (random) feasible options and selecting the best one via a binary optimization problem is always possible (as proposed in **[313, 312]**). It is, however, apparent that a purely random choice is only suitable when the space of feasible schedules is small or the optimized target is only loosely depending on the exact load schedule. Whenever the search space is too large, a systematic rather than a purely random approach for generating feasible options should be used. To see this, consider trying to solve an optimization problem by simply drawing random points. The resulting set of points provides the options to choose from. The larger the set of admissible points grows, the smaller the chance of drawing a point even close to the optimum. Hence, a more systematic approach should generally yield better results. If, on the other hand, the set of optimal points is very large, drawing points randomly may still suffice. In terms of load schedules this is the case, when the exact load is only relevant at a few points in time, for instance if only the maximum load is of interest (and costs are secondary).

Table 6.1: Challenges associated with the individual surrogate modeling approaches and possible solutions. Most challenges are faced by the external entity (ExEn) who wants to make use of the flexibility offered by each flexibility provider (FlPr).

	Surrogate	Challenge	Faced by	Proposed solution
Schedule generation	Generation	Mapping from representation to load schedule	FlPr	May be learned, see [298]
		Generic crossover and/or mutation functionality	ExEn	May be learned, see [266]
			ExEn	Generic algorithms without problem-specific knowledge
	State-based simulation	Selection of good actions	ExEn	Reinforcement learning
	Repair	Consistent mechanism for schedule repair	FlPr	Stepwise correction
		Finding good schedule for each FlPr	ExEn	Heuristics determining desirable reference schedules, see [11]
	Fragmented classification	Selection of good actions	ExEn	Perform stepwise repair with surrogate and use solution for repair approach
	Classification	Identifying feasible schedules as input	ExEn	Backpropagation to the input
		Finding good schedule for each FlPr	ExEn	Heuristics determining desirable reference schedules
			ExEn	(Binary optimization using derived sets of schedules)
	Cost evaluation	Identifying feasible schedules as input	ExEn	Backpropagation to the input
		Finding good schedule for each FlPr	ExEn	Heuristics determining desirable reference schedules
			ExEn	(Binary optimization using derived sets of schedules)
Signal selection	Tariff outcome	Finding a good input for a single FlPr	ExEn	Backpropagation to the input
			ExEn	Heuristics inspired by black-box optimization
		Finding good inputs for each FlPr	ExEn	Heuristics inspired by black-box optimization
	Request outcome	Identifying suitable requests for each FlPr	ExEn	Binary optimization using derived sets of options
	Constraint outcome	Identifying suitable requests for each FlPr	ExEn	Binary optimization using derived sets of options
	Deception outcome	Finding a good input for a single FlPr	ExEn	Backpropagation to the input
			ExEn	Heuristics inspired by black-box optimization
		Finding good inputs for each FlPr	ExEn	Heuristics inspired by black-box optimization

Table 6.1 summarizes the various challenges to overcome in order to derive signals with the help of each surrogate and proposes solutions. For some challenges two alternatives are listed. In the following, the individual approaches are discussed one by one. It is important to keep in mind, this is a purely theoretical analysis and the stated solutions are merely propositions.

Generation: For learning a mapping $a_{0:T-1} = g(z, s_0)$ from an abstract representation z to a load schedule $a_{0:T-1} = (a_0, a_1, \dots, a_{T-1})$ in the given state s_0, there either needs to be an existing mapping to learn from, or a new one needs to be developed first. If the respective flexibility provider makes use of evolutionary algorithms, such a mapping is already available. If not, one must be created, which is a process requiring in-depth knowledge and experience if done by hand. However, there are approaches to automate this process via machine learning (e.g., [298]). As the composition of DERs varies and the learning process creating the mappings is influenced by randomness, each flexibility provider i transmits their own specific mapping $g^i(z, s_0)$ to the external entity, all with different representations z. While it would be possible to use these mappings for generating load schedules by simply passing random inputs z, the provided functionality suggests utilizing the mapping in an evolutionary algorithm. In theory, by combining the representation (or genotype) z of multiple flexibility providers to a new individual $\tilde{z} = (z^1, z^2, \dots, z^N)$ and computing the load schedule $a_{0:T-1} = \sum_{i=1}^{N} g^i(z^i, s_0^i)$, their joint operation can be optimized with an evolutionary algorithm. The target function of optimization *Problem I* simply changes to

$$\frac{1}{2}\left\|\bar{a} - \sum_{i=1}^{N} g^i(z^i, s_0^i)\right\|_2^2 .$$

In optimization *Problem II*, the constraint changes to

$$\underline{b} \leq a_\tau \leq \bar{b}.$$

Given the different mapping $g^i(z^i, s_0^i)$ for each flexibility provider, adapting mutation and crossover functionality for different combinations of flexibility providers seems very resource intensive. Hence, either should these functions be generic or learned in an automated way (e.g., [266]).

State-based simulation: In the state-based simulation, the surrogate consists of two parts, a classifier identifying feasible actions and a predictor estimating the state after performing a given action. The challenge is to decide which action (=power) to choose at each time step. In other words, the problem at hand is to find a good policy $\pi(a|s)$ specifying the probability of selecting action a in state s. Identifying such a policy is the goal of RL. The utilization of RL in order to solve this challenge therefore suggests itself. For a brief introduction to RL please see the Section 2.5.8. The implementation options are not limited to ANNs, learning classifier systems are another popular approach in RL [54, p.19], which could be applied. In the RL framework, policies are rated based on the expected rewards $r(s_t, a_t)$ accumulated over time. Hence, in order to solve the two optimization problems considered here, the target function must be expressed as total reward $\sum_{t=0}^{T-1} r(s_t, a_t)$.

In case of *Problem I* this is simple, as $\|x\|_2^2 = \sum_k x_k^2$ can be separated into individual terms. By selecting $r(s_t, a_t) = -\frac{(\bar{a}_t - a_t)^2}{2}$, the negated target function is exactly replicated. The function is negated since rewards are maximized, and maximizing the negative deviation yields the identical, minimal solution only with the target function value being negated. For optimizing the aggregated schedule, all flexibility providers must be considered simultaneously. This means that the global state $s_t = (s_t^1, s_t^2 \dots, s_t^N)$ is composed of all individual states, and the action $a_t = (a_t^1, a_t^2, \dots a_t^N)$, which is selected with the help of the policy $\pi(a|s)$, is a vector comprising all individual actions. Therefore, the reward function is

$$r(s_t, a_t) = -\frac{(\bar{a}_t - \sum_{i=1}^{N} a_t^i)^2}{2}.$$

With the states s_t^i provided by the surrogates and the reward function defined, all theoretical prerequisites for applying RL are satisfied. However, the feasibility of actions has not yet been considered. Probably the simplest way to make use of the information provided by the surrogates' action classifiers is separately checking if the selected actions $a_t \sim \pi(a|s_t)$ are feasible and selecting the closest feasible actions, i.e., power levels, instead if they are not (compare post-posed shielding in RL [299]). Alternatively, if the state transition model itself is able to perform this correction and output the corrected result, a classifier is not needed at all. Either way, the corrected action is passed to the reward function instead of the infeasible one.

As outlined in the fundamentals, RL provides many options for learning policies. Model-based RL methods are especially promising, if the state transition ANN given by the surrogate can directly be integrated into the gradient computation for deriving policy updates. If ANN-based RL is applied, the result is a trained ANN that can be used to determine which actions should be taken in response to given states. In this case, determining actions is very easy once a policy function has been trained. It is furthermore possible to improve the policy during operation. *Problem II* can be solved similarly, by adapting the rewards to consider costs instead and by further restricting the action selection. In contrast to the first optimization problem, the aggregated action is part of a constraint. Should the aggregated load in time step τ not satisfy the constraints $\underline{b} \leq \sum_{i=1}^{N} a_\tau^i \leq \bar{b}$, a valid combination of actions needs to be identified and used instead. To avoid having to train an individual policy for each combination of τ and constraints $\underline{b}, \bar{b}$, these three parameters could be added to the state vector.

Repair: Teaching an ANN how to repair infeasible load schedules, so that it transforms them into feasible ones, is a nontrivial task. The first issue arising in this process is the question of how the intended repair should be achieved. Consider an empty BESS with a capacity of 10 kWh and the schedule $a_{0:3} = (1, 10, 2, 1)$, where each element is the charging power in kW consumed over the period of one hour, for example. When neglecting efficiency, SOC dependent behavior, and similar factors, it is apparent that the schedule is infeasible, as it adds up to a total of 14 kWh to be stored. One possibility to fix this schedule would be to cut back the biggest load $(1, 6, 2, 1)$. Alternatively, the loads could be scaled $\frac{10}{14} \cdot (1, 10, 2, 1)$. The schedule could also be traversed chronologically replacing the values step by step with the closest feasible power, resulting in $(1, 9, 0, 0)$, as 9 kWh free capacity are left in step two and afterwards the BESS is full. Even in this simple example, there are countless possibilities to fix this schedule. In **[312]** we attempted to repair load schedules by finding the closest feasible schedule with the help of a MILP. However, depending on the metric for determining the distance between schedules, there may be many good or even optimal solutions. Take the MAE, for instance. All three schedules given in the example above have an MAE of $\frac{4}{4} = 1$. Hence, any of these schedules could be the solution of the MILP. Furthermore, with parameters like the optimality gap, time limits, and potential random effects due to parallelization, even with the identical input, different repaired solutions may be found by the MILP. For an ANN, in order to learn a meaningful relation between input and output, it must be present in the training data. Like the results in **[312]** indicate, this is not necessarily the case with schedules repaired by a MILP. One way to deal with this problem could be to measure distances with other metrics that give a more clear solution, if existent. Another option, which is also the proposed solution found in Table 6.1, is to stick to the chronological step by step repair. This method of repair always produces the identical output for a given input. Once a well performing surrogate for the repair task is available, there are different options to solve optimization problems.

Bremer [11] demonstrates how different heuristics, namely particle swarm optimization (PSO), artificial bee colony, and harmony search, can be combined with such a schedule repair approach. The integration is achieved by running the algorithm on the space of all schedules, which comprises all feasible and infeasible schedules, and doing a repair whenever a point needs to be evaluated. In the case of PSO, as an example, this means that the particles can freely move within a hypercube, but are mapped individually to feasible schedules only for the purpose of evaluating the target function value at the given location. If a combination of N flexibility providers needs to be evaluated, like in *Problem I*, the dimension of the search space grows from T to $T \times N$. In the PSO example, each particle therefore consists of N schedules of length T, which are repaired individually with the help of the respective surrogate. Hence, all N schedules must be processed with their associated surrogate every time a target function value is computed. Aside from using these meta-heuristics, Bremer proposes a repair-based heuristic for the decentralized coordination, which aims to solve *Problem I* without repairing all N schedules every single iteration. The heuristic repeatedly selects one flexibility provider at a time, who then greedily determines the a good schedule to follow while ignoring all constraints. In case of *Problem I*, the best schedule would be the difference between the target and the current aggregated schedule of all other flexibility providers, i.e., its own schedule not included. If the flexibility provider was able to follow this greedy solution, the sum of all schedules would exactly meet the target. However, the resulting greedy schedule is generally not feasible, hence, a repair is conducted before updating the own schedule accordingly. Until some stopping criterion is reached, this process is repeated. To solve *Problem II*, in addition to determining a desirable (greedy) schedule, it is necessary to check if the aggregated schedules do not exceed the given boundaries in time step τ. If they do, the only option is to continue until at least one admissible combination is found. While it could help to pass only schedules to the repair function that already satisfy the given bounds, there is no guarantee that the repaired schedule will do as well.

Fragmented classification: The fragmented classification makes use of one or multiple ANNs to predict whether a schedule $a_{0:t-1} = (a_0, a_1, \ldots, a_{t-1})$ of length t is feasible $f(a_{0:t-1}) = 1$ or not $f(a_{0:t-1}) = 0$. In contrast to the state-based simulation with its dedicated model for identifying feasible load levels for each time step, the fragmented classification provides this information only indirectly. Starting with a schedule $a_{0:0} = (a_0)$ of length one, the classifier can be used to find all feasible loads a_0 by simply testing each $f((a_0))$. With the help of frameworks like TensorFlow and PyTorch, all these classifications can easily be parallelized by passing batched inputs to the ANNs. Afterwards, a_0 can be selected from the identified admissible options. Since a schedule only becomes infeasible once an infeasible power is appended, the same process can be repeated for a_t by computing $f(a_{0:t}) = f((a_0, a_1, \ldots, a_t))$ until length T is reached, that is, until $t = T - 1$. Identifying and selecting actions is thus the main challenge associated with this approach. However, the previously proposed solution of applying RL for this kind of problem is not directly applicable here, as the state of each time step is a required policy input. While the initial state is made available by the flexibility provider, all subsequent states are unknown, as the classifiers do not provide the predicted state.

One possibility to deal with this issue could be the augmentation of the state vector, adding the selected loads one by one, but this would require the RL model to learn even more about the DERs' dynamics. For this reason another solution is proposed: given the surrogate performs well in the classification task, the approach can be used to make step by step repairs of load schedules. For a desired schedule $\bar{a}_{0:T-1} = (\bar{a}_0, \bar{a}_1, \ldots, \bar{a}_{T-1})$, the repaired schedule $\tilde{a}_{0:T-1}$ is built by selecting the admissible action closest to the given target in each time step. In the first step, $\bar{a}_0$ can be tested by computing $f((\bar{a}_0))$. If it is feasible, that is, $f((\bar{a}_0)) = 1$, we have $\tilde{a}_{0:0} = (\bar{a}_0)$. If it isn't, the closest feasible $\tilde{a_0}$ is searched and $\tilde{a}_{0:0} = (\tilde{a_0})$. Repeating this process for $\bar{a}_1$, $\bar{a}_2$ and so on, yields the repaired schedule. With the help of this repair mechanism, the same approaches described for the schedule repair surrogate are applicable.

Classification: Identifying feasible load schedules $a_{0:T-1}$ using only a classifier $f(a_{0:T-1})$ is rather challenging, as feasible schedules only make up a small fraction of the space of all schedules [11]. Providing external entities with further information to make the search process easier is conceivable, but would probably require a lot of additional knowledge. Backpropagation, which is a fundamental mechanism used for updating the weights of an ANN, could be a possible solution to deal with this issue without the need for extra data. In a typical backpropagation step, the ANN input is a fixed function argument and the weights are updated according to the gradients computed by determining the derivative of the loss function with respect to the weights. However, it is possible to compute the derivative with respect to the input vector instead, while, this time, the weights remain fixed. The classifier's rating $f(a_{0:T-1}) \in [0, 1]$ indicates the confidence for a schedule to be feasible. A higher value means a stronger belief. As gradients always point in direction of the steepest ascent, the gradient $\nabla_{a_{0:T-1}} f(a_{0:T-1})$ points in direction of supposed feasibility. It therefore provides an indication of how an infeasible input should be updated to improve the rating. By repeated updates, the schedule could be transformed step by step until it is supposedly feasible, that is, until a given threshold for the rating is exceeded. Performing such a procedure multiple times with different starting inputs and different step sizes for the gradient steps, could yield a set of feasible schedules. Whether this proposed procedure is efficient and results in diverse load schedules or not remains to be seen. While this, again, can be seen as a repair mechanism, it is to be expected to need even more computational resources, because of the repeated gradient computations. Therefore, the methods proposed for the repairing approach that require frequent schedule repairs are not entirely deemed suitable. The distributed coordination approach presented in [11] and outlined above, on the other hand, performs fewer repairs by determining a desired schedule for every flexibility provider and repairing this schedule in hope of it being similar to the desired one. Such an approach could still be applicable for the exemplary *Problem I* and *Problem II*, even though more computations are needed with the backpropagation steps.

Cost evaluation: Predicting the cost $f(a_{0:T-1})$ of realizing a schedule $a_{0:T-1}$ is somewhat related to predicting a schedule's feasibility. The cost $f(a_{0:T-1})$ is not to be confused with the cost $c(a_{0:T-1})$ found in the target function of *Problem II*. While $c(a_{0:T-1})$ is the cost paid by the external entity for requesting schedule $a_{0:T-1}$, $f(a_{0:T-1})$ is the operational cost faced by the flexibility provider for following said schedule. In order for the ANN output $f(a_{0:T-1})$ to be meaningful, it is generally not sufficient to train with feasible schedules alone. As pointed out before, there are relatively few feasible schedules is the space of all schedules $\mathbb{R}^T$. Hence, training with only feasible samples, the ANN output would be unpredictable for infeasible regions. By associating infeasible schedules with very high costs, that is, $f(a_{0:T-1}) = \mathbf{M}$, like the penalty function method does, it is possible to distinguish between valid and invalid ones. Therefore, given a trained ANN capable of fulfilling this prediction task, the same solution as proposed for the classification approach can be applied. Since a smaller cost instead of a larger rating indicates feasibility, here, the gradient needs to be subtracted instead of added. Furthermore, with this surrogate, there is the added benefit of having cost estimates at hand. For instance, if the planned schedule $\bar{a}_{0:T-1}$ is known, the cost of deviating could be estimated by computing $c(a_{0:T-1}) = f(a_{0:T-1}) - f(\bar{a}_{0:T-1})$.

Tariff outcome: The trained ANN in this approach provides the expected schedule $a_{0:T-1} = f(c_{0:T-1})$ as a response to a tariff $c_{0:T-1} = (c_0, c_1, \ldots, c_{T-1})$ given as input. Using backpropagation as described above, it could be possible to derive tariffs incentivizing a flexibility provider in such a way that they exactly produce a desired schedule $a_{0:\bar{T}-1}$. Alternatively, a tariff could be searched by performing systematic adaptations until a target is approximated close enough. The field of black-box optimization could provide approaches to implement such a systematic search algorithm (see Section 2.5.5 for further information on black-box optimization). For *Problem I*, when adding the surrogate, the target function value is

$$z = \frac{1}{2} \left\| \bar{a} - \sum_{i=1}^{N} f(c_{0:T-1}^{i}) \right\|_2^2 .$$

Aside from the surrogate $f(\cdot)$ being only part of the objective rather than the objective itself, the objective function can still be treated as a black-box. For *Problem I* and *Problem II*, the external entity is only indirectly searching $a_{0:T-1}^{i} \in \mathbb{M}^{i}(s_0^i), i = 1, 2, \ldots, N$ by trying to identify tariffs $c_{0:T-1}^{i} \in \mathbb{R}^{T}, i = 1, 2, \ldots, N$. The target function of *Problem II* changes analogously and becomes

$$\sum_{i=1}^{N} c(c_{0:T-1}^{i}).$$

Tariffs pose as an incentive to change the planned schedule of DERs and, thus, only allow to influence their operation. Flexibility providers are free to change their plans at any time. Because of this, there is no guarantee that a predicted solution for *Problem I* or *Problem II* plays out as planned. Nevertheless, as in *Problem II* only the load a_τ in time step τ is relevant, the problem is much easier to solve reliably than *Problem I*.

Request outcome and constraint outcome: These two surrogate modeling approaches are very similar to each other. Both predict the fulfillment of a given request or constraint by providing an estimate of the duration it can be satisfied. As outlined before, further outputs could provide additional information on the change in consumed or produced energy and possible rebound effects. The major difference between the two is that the former approach requests a defined change in load, while the latter one only imposes constraints and leaves the exact change in load up to the flexibility provider. In comparison to the other surrogate modeling approaches discussed here, the two approaches possess rather limited functionality and are only suited to influence the operation of DERs for a limited time. Therefore, they are not proper options for solving *Problem I*. *Problem II*, on the other hand, is perfectly solvable. Instead of searching $a_{0:T-1}^{i} \in \mathbb{M}^{i}(s_0^i), i = 1, 2, \ldots, N$, the external entity now searches the load difference $\Delta a_\tau^i \in \mathbb{M}^{i}(s_\tau^i), i = 1, 2, \ldots, N$ (request outcome) or checks beforehand whether the bounds $\underline{b}$ and $\bar{b}$ can be satisfied or not (constraint outcome). The base load $\bar{a}_{0:T-1}^{i}$, which is required to compute the resulting load $\sum_{i=1}^{N} \bar{a}_\tau^i + \Delta a_\tau^i$ in time step τ, and the predicted state s_τ^i are provided by the flexibility provider i. Due to the limited use of the surrogates, the space of possible input combinations is comparatively small. It is hence possible to simply evaluate every desirable option and pick the best combinations for the different flexibility providers.

Deception outcome: This approach is based on the manipulation of environmental state variables, which are sent to the flexibility providers in order to deceive them and incentivize a change in their behavior. These environmental variables are an input for the ANN, which in turn predicts the resulting schedule. Basically this approach is a generalization of the tariff outcome approach. Instead of a tariff, which could also be interpreted as part of the environmental state, some other variable is passed. For this reason, the identical solutions are proposed.

6.3 Qualitative Assessment

Now, with the optimization approaches outlined in the previous section, a qualitative evaluation can be conducted. The criteria comprise the adaptability of the time horizon, the complexity of implementing and using the surrogate modeling approach, as well as its scalability. Complexity, in this context, refers to the required amount of computational resources. All results are compiled in Table 6.2. Independent of the model, a longer time horizon can be achieved by either increasing the number of time steps, or increasing the length of a time step. Usually the length of a time step is a fixed parameter. If it was changed during operation, new surrogates would have to be trained, regardless of the surrogate modeling approach. Increasing the number of time steps, on the other hand, adds more dimensions to the space of load schedules, which in turn makes training and using the model more demanding. While each trained ANN takes a fixed number of inputs for computing a fixed number of outputs, some models are applicable for load schedules of variable lengths and others are not. For each surrogate, it is therefore distinguished if it operates on a fixed or a flexible number of time steps. Also, by design, some surrogates can only cover short periods of time. Whether they are only applicable for short time horizons or not, is indicated by the labels “short” and “any”, respectively. For complexity and scalability, the ratings *low* (L), *medium* (M), and *high* (H) are assigned. While a high scalability is generally a good feature, a high complexity is not. The complexity is not rated for the overall approach, but for three separate steps of the exploitation process: firstly, the generation of the surrogate itself by each flexibility provider. Secondly, the training of models by the external entity needed for the proposed solution method. Lastly, the exploitation with the help of the surrogate, that is, the external entity’s search for the best signals to send to the flexibility providers. Based on these complexity ratings, the overall scalability is judged. Please note that even though arguments for all judgement are provided, the ratings are based on very general assumptions and are (inevitably) to some degree subjective.

Table 6.2: Qualitative assessment of the suitability to control DERs. In case of being applicable for “any” time horizon, a surrogate modeling approach is rated with a long horizon, e.g., 96 quarter hours, in mind. A short time horizon is, for instance, a single hour and comprises only few time steps. Complexity refers to the required amount of computational resources. Possible ratings are *low* (L), *medium* (M), and *high* (H). Providing a rating for the *deception outcome* approach is not possible without making further assumptions.

	Surrogate	Time horizon		Complexity			Scalability
				Training FlPr	Training ExEn	Exploitation	
Schedule generation	Generation	fixed	any	M/H	(H)	M	M
	State-based simulation	flexible	any	H	H	L	M
	Repair	(fixed)	any	M		M	H
	Fragmented classification	(flexible)	any	M/H		M/H	M
	Classification	fixed	any	L		H	L
	Cost evaluation	fixed	any	M		H	L
Signal selection	Tariff outcome	fixed	any	M		M	M
	Request outcome	flexible	short	L		L	H
	Constraint outcome	flexible	short	L		L	H
	Deception outcome	*	*	*	*	*	*

The surrogate of the *generation* approach takes the state, as well as an abstract load schedule representation as an input and computes the associated load schedule. This load schedule has a fixed size, as the number of output neurons is fixed. If an RNN is used, the number is flexible, but this also requires a representation able to scale in length. The number of output neurons and thereby the schedule length can freely be chosen prior to training. Hence, the approach is theoretically applicable for any time horizon. In practice, the associated computational requirements limit the number of time steps, which is also true for all other ANN-based surrogate modeling approaches. The surrogate is trained by the flexibility provider, either using data gathered from an already existing EA implementation, or by automatically generating a new mapping by means of machine learning. A *medium* to *high* complexity rating has been assigned for this step, as multidimensional inputs are mapped to multidimensional outputs, and the task of creating such a mapping for an EA requires the ability to abstract. If an external entity achieves the transformation of representations with the help of custom learned models, instead of using generic functions able to handle generic representations as inputs, the associated models must be learned. This involves, again, the learning of mappings from multidimensional to multidimensional vectors, and must be repeated for any combination of flexibility providers to optimize them collectively. Therefore, a *high* complexity rating has been chosen. For the exploitation, an EA is executed, a meta-heuristic often able to perform well with comparatively small computational resources. However, the usage of ANNs in one or multiple EA steps highly intensifies the computational complexity, which is the reasoning for the assigned *medium* rating. Taking all these ratings into consideration, scalability has been rated *medium* as well, since the EA can be expected to scale well with the number of flexibility providers (see for instance [300] which tests different EAs, a PSO algorithm, and a scatter search algorithm on problems with scalable numbers of variables), but the burden of the ANN computations is added.

The second listed surrogate modeling approach is the *state-based simulation*. Since the state transition model is used like an RNN, creating load schedules by adding one element after another, schedules of any length can be created. This versatility comes at the cost of needing increasingly precise models with growing schedule length, as the errors introduced with each transition add up. Furthermore, not only a state transition model needs to be trained, but also a classifier. For this reason, a *high* complexity rating has been chosen. The proposed solution for deriving signals is RL, which involves the learning of a policy. While such a policy can be learned from scratch and improved during the operation, it is reasonable for an external entity to train a basic policy beforehand. In order to train the policy, a very large number of ANN computations is required. Therefore, a *high* rating was given. During exploitation, with a trained policy at hand, the remaining effort is *low*, as the policy provides the action to choose for a given state. Even though the exploitation has been rated *low* complexity, scalability is deemed only *medium* as the training for both, the flexibility provider and the external entity, is quite resource intensive. One factor is the increasing size of the state vector with each additional flexibility provider. A possible way to deal with this issue is to make use of an autoencoder, like proposed in [113]. For all remaining surrogate modeling approaches, there is no need for the external entity to train ANNs.

Next is the *repair* approach. It returns a (likely) feasible schedule, given a potentially infeasible one. If the underlying repair mechanism works step by step, an RNN could be used, allowing the application for any time horizon. When no RNN is used, the number of time steps in the schedule is fixed, but can be chosen arbitrarily before training the model. In this approach one ANN is trained, mapping multidimensional inputs to multidimensional outputs, which is rated *medium* complexity. The exploitation is rated likewise, as the proposed heuristic involves repeated updates of the load schedules. In contrast to the *generation* approach a *high* rating has been selected for the scalability, since no additional ANNs or generic transformation functions are needed.

Schedules of variable length are classified in the *fragmented classification* approach. Hence, the time horizon is flexible, but may be bounded, depending on the trained ANN's topology. Prior to training, any time horizon may be selected. Even though the ANN only needs to learn a mapping from multiple to a single dimension, the complexity has been classified as *medium* to *high*. The reason is that either the ANN must be able to handle schedule inputs of varying length, or individual ANNs for each schedule length up to T must be trained. In the first case, the classification task becomes more complex, as the ANN essentially needs to learn a complete flexibility model to reenact every step of the schedule. The latter variant requires the training of many ANNs, where the sheer number of ANNs leads to a high complexity for a large time horizon T. As for the exploitation it is proposed to use the classifier to repair schedules step by step, the remaining ratings are based on those given to the *repair* approach. In contrast to the repairing ANN, which directly yields the repaired schedule, the classifiers must be used many times in order to repair a single schedule, one value after another. Hence, the exploitation complexity is *medium* to *high*, and the scalability is only *medium*.

The *classification* and *cost evaluation* approaches are very similar to each other. They take a fixed input comprising the state and a schedule, and provide either a feasibility rating or the estimated cost for the given schedule. The length of the schedule passed to the ANN is generally fixed with an arbitrary number of time steps. Using an RNN is conceivable and would lead to flexible time horizons, but would yield the *fragmented classification* approach instead of the *classifier*. Since the cost of realizing a schedule includes the information whether a schedule is feasible or not, the cost predicting ANN is facing the more difficult task. Hence, the classifier is rated *low* and the cost predictor *medium* complexity. The identified exploitation strategy for both surrogates requires the repeated use of gradient descent algorithms, and is therefore very costly, leading to a *high* complexity score. Scalability is thus deemed *low*.

In the *tariff outcome* approach, load schedules are influenced by adapting price profiles. The surrogate predicts schedules from price profiles, both with the same arbitrary but fixed length. Since a price change in one time step affects the resulting load of other time steps as well, the whole schedule must be computed all at once. The task involves mapping a multidimensional input to a multidimensional output and is therefore rated with a *medium* complexity. From a general perspective, the set of possible tariffs is huge, but in practice it would be limited by certain constraints, for instance, minimum and maximum prices, or a given fixed average (e.g., [5]). It is furthermore possible to discretize the price levels appearing in the price profile, as price differences between time steps must be large enough to compensate for efficiency losses. Take, for instance, a BESS with a 0.92% round-trip efficiency and assume it could be charged from the grid at a cost of 25 cents per kWh. This would only be economical if at a later time step energy is more expensive or can be fed into the grid for more than $\frac{25}{0.92} = 27.17$ cents per kWh. Hence, minor price variations in a price profile, lead to minor schedule changes at most. For the exploitation, heuristics based on black-box optimization techniques have been proposed as main solution. Taking the less, but still large input space into account a *medium* rating has been selected for the complexity and scalability.

The *request outcome* and *constraint outcome* approaches both share the same ratings, as they are very similar in their use: Both surrogates estimate the duration a request or imposed constraint can be fulfilled. Since the returned duration varies from input to input, the time horizon is flexible. Depending on the exact request or constraint, the result may even be zero time steps, meaning the tested input is not feasible for the flexibility provider. Interventions based on such mechanisms are only sensible in the short term, as the set of options the external entity chooses from is limited and stacking such requests is not directly possible. The rather limited set of possible inputs reduces the complexity. Furthermore, the dimensions of the inputs and outputs the ANNs must process are smaller than for all the other approaches. Therefore, *low* complexity scores in combination with a *high* scalability rating have been assigned.

The procedure for deriving scores employed in this chapter is not applicable to the *deception outcome* approach, as it is unclear how the input to the surrogate model would be structured. However, ratings similar to those of the *tariff outcome* approach can be expected.

6.4 Exemplary Implementation

In order to prove that ANN-based surrogates can be used to exploit the flexibility of DERs, a search algorithm based on MCTS has been implemented for solving optimization *Problem I* (following an aggregated target schedule) with the help of the state-based simulation approach. A basic explanation of MCTS is provided Section 2.5.8.

6.4.1 Monte-Carlo-Tree-Search-Based Target Schedule Approximation

Typically, MCTS is applied in an MPC fashion, i.e., only one action is selected and implemented, and then the algorithm is executed again. In its search for the best choice, MCTS builds a tree in which it tracks explored actions and states, growing both in breadth and depth. However, in order to find a load schedule, a solution covering the entire planning horizon must be generated. It is therefore crucial to ensure that the optimization algorithm advances to terminal states, instead of exploring too far in breadth. This is achieved with two adaptations: firstly, making a full depth search in each iteration, before starting at the root again, and, secondly, adding all trajectories gained through simulation steps to the tree.

A challenge in finding good solutions is the vast number of possible actions. Take the aggregate of 10 BESSs, for instance. Usually, each BESS should be able to either charge or discharge, meaning at least half of the individual action space is feasible. With 201 discrete actions (100 charging, 100 discharging, 1 idle), as tested in the previous chapter, this means that at any time there are at least $(100+1)^{10} \approx 1.105 \cdot 10^{20}$ possible combinations. Exploring each option is obviously impossible. An aggravating factor is the high cost of running simulations. In order to select an edge and compute the state for the associated child node, two ANNs need to be evaluated for every single DER in the ensemble. This high cost is the reason as to why each simulated trajectory is entirely added to the tree, instead of discarding the computed states. Furthermore, each node is only expanded with a few randomly selected child nodes. If a node is visited more than once, it lies on the best trajectory known yet and is therefore expanded again.

In summary, the adapted algorithm makes a full depth search in each iteration and adds simulation results to its exploration tree. The search is conducted by expanding each visited node randomly, running a simulation for each newly added child, propagating the simulation results upwards to the root, and then selecting the best child node. This is repeated until the terminal state is reached, which marks the end of an iteration. Subsequent iterations may visit the same branches again. Algorithm 1 provides a detailed explanation of the individual steps.

The first input is $\bar{a} \in \mathbb{R}^T$, which is the aggregated target schedule the N flexibility providers should achieve. This schedule is determined by the external entity, according to their needs. The individual flexibility providers $i = 1, \ldots, N$ supply their current state s_0^i, forecasts p_t^i, and binary masks b_t^i for the time steps $t = 1, \ldots, T$. Forecasts and the binary masks are only needed in case the surrogate does not make its own predictions. Each forecast p_t^i and mask b_t^i is a vector of the same dimension as s_t^i. In order to inject the forecast into the schedule generation process, the state s_{t+1}^i is replaced with $(1 - b_{t+1}^i) \cdot s_{t+1}^i + b_{t+1}^i \cdot p_{t+1}^i$ each time a new state has been predicted. The mask b_t^i determines which elements are replaced by the external forecast and which are not.

Algorithm 1: MCTS-based target schedule approximation

Data: Target schedule $\bar{a} \in \mathbb{R}^T$; initial state s_0^i, state-based simulation surrogate $(A^i(s^i), f^i(s^i, a^i))$, external forecast $(p_t^i)_{t=0,\dots,T-1}$, and binary mask $(b_t^i)_{t=0,\dots,T-1}$ for each flexibility provider $i = 1, \dots, N$; maximum number of iterations K; number of simulations per node L

Result: Schedules $a_{0:T-1}^i \in \mathbb{R}^T, i = 1, \dots, N$

begin

- n_{root} = Node(parent = *None*, $t = 0$, $s = (s_0^i)_{i=1,\dots,N}$, a = *None*, $r_{<t} = 0$, $r_a = 0$, $r_{\text{sim}} = 0$, $n = 0$)
- **for** $k = 0$ *to* $K - 1$ **do** // do K iterations
  - $n_{\text{search}} = n_{root}$ // **selection**, start at the root
  - **for** $t = 0$ *to* $T - 1$ **do** // iterate full time horizon (full depth search)
    - **for** $l = 0$ *to* L **do** // **expansion** of each node via L simulation runs
      - $n_{\text{sim}} = n_{\text{search}}$ // **simulation** starting in node n_{search}
      - **for** $t = n_{sim}.t$ *to* $T - 1$ **do** // simulate steps until time step T is reached
        - $\Delta\bar{a}_t = \bar{a}_t$ // aggregated power difference to target
        - $j = 0$ // counter
        - **foreach** i *in randomizeOrder*$([1, \dots, N])$ **do** // iterate flexibility providers
          - a_t^i = randomAction$(A^i, n_{\text{sim}}.s^i, \Delta\bar{a}_t, N, j)$ // select action via Algorithm 2
          - $\Delta\bar{a}_t = \Delta\bar{a}_t - a_t^i$ // update aggregated delta
          - $s_{t+1}^i = f(n_{\text{sim}}.s^i, a_t^i)$ // predict subsequent state
          - **if** $t < T - 1$ **then**
            - $s_{t+1}^i = (1 - b_{t+1}^i) \cdot s_{t+1}^i + b_{t+1}^i \cdot p_{t+1}^i$ // inject external forecasts
          - **end**
          - $j = j + 1$
        - **end**
        - n_{sim} = Node(parent = n_{sim}, $t = t + 1$, $s = (s_{t+1}^i)_{i=1,\dots,N}$, $a = (a_t^i)_{i=1,\dots,N}$, $r_{<t} = n_{\text{sim}}.r_{<t} + n_{\text{sim}}.r_a$, $r_a = -$|delta_target|, $r_{\text{sim}} = 0$, $n = 0$)
      - **end**
      - **if** $n_{sim}.r_{<t} + n_{sim}.r_a = 0$ **then**
        - **return** $a_{0:T-1}^i$ *extracted from nodes leading to* n_{sim} // found an optimal solution
      - **end**
      - $r_{\sum a} = n_{\text{sim}}.r_a$ // n_{sim} is currently a leaf
      - **while** $n_{sim} \neq n_{root}$ **do** // **backup** simulation results
        - $n_{\text{sim}} = n_{\text{sim}}$.parent // step towards root
        - $n_{\text{sim}}.r_{\text{sim}} = n_{\text{sim}}.r_{\text{sim}} + r_{\sum a}$ // update total simulation reward
        - $n_{\text{sim}}.n = n_{\text{sim}}.n + 1$ // increment simulation count
        - $r_{\sum a} = r_{\sum a} + n_{\text{sim}}.r_a$ // subsequent simulation reward
      - **end**
    - **end**
    - n_{search} = child of n_{search} with the highest reward // **selection** of next node
  - **end**
- **end**
- **return** $a_{0:T-1}^i$ *extracted from sequence of nodes with the highest reward*

end

All surrogates $(A^i(s^i), f^i(s^i, a^i))$ needed by the state-based simulation approach are already known by the external entity, as they have been communicated beforehand and updates are only required whenever the underlying model changes. Further parameters are the maximum number of iterations K, that is, the number of times a full depth search starting at the root is conducted, and the number of simulation runs per visited node. Since simulation runs are added to the tree, expansion is conducted via simulation and the number of simulation runs equals the number of added child nodes. The algorithm returns schedules of length T for the N individual flexibility providers.

Each tree node stores a pointer to its parent node, the associated time step t, a list s of the individual state vectors s_t^i, a list a of the individual actions a_{t-1} selected in the previous state, and several rewards explained in the following. The variable $r_{<t}$ stores the reward accumulated by the parent nodes. Rewards gained by action a_{t-1} are stored in r_a. Finally, the total of all subsequent rewards gained from a_t to a_{T-1} is held by r_{sim}. The expected reward at a node is given by $\frac{r_{\text{sim}}}{n}$, where n is the number of simulated trajectories leading through the respective node.

Actions are selected randomly, one by one for each flexibility provider $i = 1, \ldots, N$, using Algorithm 2. The chance of a combination of actions to be selected depends on their absolute distance from the target load. Combinations with a greater distance from the target are less likely to be chosen. In order to achieve this, the remaining total distance from the target schedule $\Delta\bar{a}_t$, the number of flexibility providers N, and the number of already selected actions j are passed to the algorithm each time an action is selected. Further inputs are the classifier $A^i(s_i)$, which is part of the state-based simulation surrogate, and the (predicted) state s^i. The classifier assigns a rating from the interval $[0, 1]$ to each considered action. Actions are deemed feasible and added to the set $\hat{A}^i$ if their rating exceeds the threshold α. The algorithm returns a single random action a^i, meant for flexibility provider i, drawn randomly from $\hat{A}^i$.

The chance of drawing the m-th action is determined by the weight vector w and equals $\frac{w_m}{\sum_n w_n}$. This weight vector is built in two steps, in which actions leading towards the target are assigned with higher weights. Initially, w is a vector of ones, which means a uniform chance of selecting any action in $\hat{A}^i$. The total power difference from the target $\Delta\bar{a}_t$ is split equally between the individual flexibility providers that have not been assigned with an action yet. If every flexibility provider was able to provide $\bar{a}^i = \frac{\Delta\bar{a}_t}{N-j}$, the target would exactly be met. This individual target determines which actions receive higher weights. Firstly, actions close to $\bar{a}^i$ are considered. Whether actions are close or not is defined by the parameter β. All actions within distance β of $\bar{a}^i$ are considered to lie in its neighborhood. The weights associated with these neighboring actions are updated in a way giving them a total chance of $\gamma \cdot 100\%$ to be selected. Hence, after this update, only with a chance of $(1-\gamma) \cdot 100\%$ an action outside the neighborhood is chosen. In a second step, the weight of the single feasible action closest to $\bar{a}^i$ is set to $|\hat{A}^i| \cdot \delta \cdot (1 + j)$. With a multiple of $|\hat{A}^i|$, the chance of selecting this option greatly increases each time an action has been fixed. The parameter δ can be used to tune the chance and reduce or increase the significance of this step. In the experiment presented below, δ equals 1. Neglecting the first weight adaptation and with $\delta = 1$, the chance of selecting the closest action is around 50% in the first, around 66% in the second, and around 75% in the fourth iteration, and so on. It should be noted that the closest feasible action is not necessarily an element of the neighborhood, since the neighborhood may be empty. Further parameter choices in the experiment are $\alpha = 0.5$ and $\gamma = 0.75$. The distance β is set to 5% of the range between the theoretical minimum and maximum power.

Since Algorithm 2 selects actions for one flexibility provider after another and becomes increasingly greedy in this process, the chance of randomly drawing a non-greedy action is very low for flexibility providers in the back of the queue. In order distribute the greedy choices more evenly, Algorithm 1 randomizes the order of the flexibility providers in each simulation step.

Algorithm 2: Random action selection with target

Data: Classifier $A^i(s^i)$, state s^i, threshold for classifying actions feasible α, total difference from target load $\Delta\bar{a}_t$, number of flexibility providers N, number of selected actions j, neighborhood radius β, neighborhood impact $\gamma \in (0, 1)$, impact of the closest action δ

Result: Action a^i

begin

  $\widetilde{A^i} = A^i(s^i)$ // rate actions

  $\hat{A}^i = \{a \mid a$ is associated with a rating in $\widetilde{A^i}$ greater than $\alpha\}$ // classify feasibility

  if $|\hat{A}^i| = 0$ **then** // if $\hat{A}^i$ has zero elements, i.e., is empty

    return *action a^i with the highest classifier rating in $\widetilde{A^i}$*

  else

    $\bar{a}^i = \Delta\bar{a}_t/(N - j)$ // split target load between flexibility providers

    $w = (1, 1, \ldots, 1) \in \mathbb{R}^{|\hat{A}^i|}$ // weight vector for random selection

    /* find actions close to the individual target, i.e., neighbors, and increase their weight */

    $\hat{A}^i{}_{\text{neighborhood}} = \{a \in \hat{A}^i \mid |a - \bar{a}^i| < \beta\}$

    if $|\hat{A}^i| > |\hat{A}^i{}_{neighborhood}|$ **then**

      /* $|\hat{A}^i|$ and $|\hat{A}^i{}_{\text{neighborhood}}|$ are not identical */

      set weight of each action in $\hat{A}^i{}_{\text{neighborhood}}$ to

$$\frac{\frac{|\hat{A}^i| - |\hat{A}^i{}_{\text{neighborhood}}|}{\gamma} - (|\hat{A}^i| - |\hat{A}^i{}_{\text{neighborhood}}|)}{|\hat{A}^i{}_{\text{neighborhood}}|}$$

      /* neighboring actions are now selected with a chance of $\gamma \cdot 100\%$ */

    end

    /* identify the feasible action closest to the individual target, i.e., the most greedy choice */

    $a = \arg\min_{a \in \hat{A}^i} |\bar{a}^i - a|$

    set weight associated with a to $|\hat{A}^i| \cdot \delta \cdot (1 + j)$

    /* the fewer flexibility providers remain, the higher the chance of selecting a */

    return *a^i, randomly drawn from $\hat{A}^i$ based on weights w*

  end

end

All parameter values, as well as the algorithms Algorithm 1 and Algorithm 2 are naive choices and implementations, yielding a greedy heuristic. They are meant to prove that ANN-based surrogates of the flexibility of DERs are capable to provide sufficient information to control flexibility providers. Finding particularly good algorithms is not possible in the scope of this thesis, but should be a goal of future research. In the following, the conducted experiments are presented.

6.4.2 Results

The goal of optimization *Problem I* is to achieve a given target schedule with an ensemble of flexibility providers. Following the example of [11], all flexibility providers are identical and only differ in their initial state. Aggregated target schedules are generated with the help of the simulation models, each time starting in new randomly drawn states. The schedules are therefore feasible and the best target function value achievable is 0, meaning a perfect match. In order to generate more realistic and less volatile schedules, actions are selected randomly with the reference-based strategy (see Section 5.3.5) and 8 equally sized reference intervals. The same reference schedule is shared between all DER, as in reality a DSMgr generally would like a uniform reaction across all flexibility providers, instead of random ones. Drawing purely random actions would likely lead to better results, as the general evaluation in Chapter 5 suggests. In [11], target schedules are either random and guaranteed to be feasible, or based on standard load profiles and possibly infeasible.

Results for the schedule search experiment are presented in Table 6.3 and Table 6.4. For each tested configuration and time horizon, 100 aggregated target schedules were generated and passed to Algorithm 1. Then, each of the resulting $100 \cdot N$ schedules was evaluated with the help of the simulation models. Each search was conducted with $K = 100$ iterations and $L = 5$ simulations in each selected node. All errors in Table 6.3 relate to the total schedule, achieved by all DERs in combination. The absolute error in energy is determined by computing the absolute difference between the target schedule and the simulated aggregated load of all DERs in terms of energy. Dividing the absolute error in energy by the total amount of energy provided or drawn yields the relative error. The MAE is the average absolute difference between the target power and the achieved load. Finally, the expected MAE is computed using the output of Algorithm 1 and the ANNs, instead of the simulation results. MAE and expected MAE deviate if the ANN-based surrogates generate infeasible load schedules.

Table 6.3: Comparison of results for the joint control of 10 identical DERs. Each device starts from a different initial state.

	State-based simulation, 2% and 2°C buffer			SVDD, 10 modulating CHPPs	
	10 BESSs	10 EVSEs	10 CHPPs	Random	Standard load profile
		8 time steps			
Absolute error (energy in kWh)	0.214	1.379	2.496	0.023	0.013
Relative error (energy)	0.004	0.011	0.110	0.0005	0.0003
MAE (power in kW)	0.107	0.689	1.248	0.011	0.007
Expected MAE (power in kW)	0.107	0.689	1.055	-	-
		32 time steps			
Absolute error (energy in kWh)	2.412	11.900	10.753	4.995	37.124
Relative error (energy)	0.016	0.055	0.077	0.031	0.238
MAE (power in kW)	0.302	1.488	1.344	0.624	4.640
Expected MAE (power in kW)	0.302	1.485	1.311	-	-

The results listed in Table 6.3 vary between DERs and time horizons. Overall, the BESS surrogate yielded the lowest errors. In the 8 time steps time horizon, that is, 2 hours in steps of 15 minutes, the EVSE surrogate performed better than the CHPP surrogate. At 32 time steps, the EVSE surrogate and CHPP surrogates produced similar results. The MAEs were 1.488 kW and 1.344 kW, respectively. However, while the EVSE ensemble can potentially draw up to 220 kW from the grid, the CHPP aggregate can only provide up to 55 kW in sum. Therefore, the relative MAE of the CHPP is greater. In regard to the expected MAE and MAE, the BESS surrogate generated identical values, which is due to the achieved 100% feasibility that is also shown in Table 6.4. For the EVSE surrogate, the values at 32 time steps deviate slightly. The rather big deviation for the CHPP surrogate was caused by the fast drop in feasibility in the first 8 time steps visible in Table 6.4. As the time horizon increased the difference between expected MAE and MAE became smaller.

Like in Chapter 5, the results of the ANN-based exploitation and SVDD-based exploitation are not directly comparable, as both were tested with different types of CHPPs. However, the similarity of the EVSE and the modulating CHPP has already been pointed out. For the 8 time steps time horizon the ANN-based results are factor 10 to 100 worse. Allowing more iterations of the search algorithm would probably lead to better results as the 32 time steps time horizon suggests, where the ANN-based results were similar to those achieved with the SVDD.

Table 6.4: Performance of the state-based simulation in a DSM task. Evaluation of 1000 load schedules for each DER and time horizon.

State-based simulation, 2% and 2°C buffer			
	BESS	EVSE	CHPP
8 time steps			
Feasible	100.0%	100.0%	97.3%
MAE*	-	-	873.125
RMSE*	-	-	1744.679
32 time steps			
Feasible	100.0%	99.7%	97.2%
MAE*	-	73.333	218.149
RMSE*	-	411.582	836.597

*only infeasible schedules (according simulation)

For each configuration and time horizon, the 100 target schedules resulted in 1000 individual BESS, EVSE, or CHPP schedules, which have also been evaluated with the help of the simulation models. Table 6.4 lists the results, providing clues how well the ANN-based surrogates describe the flexibility of the DERs in a practical DSM application. Like in the previous experiment, using the BESS with a buffer resulted in 100% feasible schedules and no errors. The EVSE results are comparable to the previous results from Chapter 5, too. Only the CHPP surrogate performed worse. In contrast to the previous 100% feasibility when using a buffer, slightly above 97% of the produced schedules were feasible. An analysis of the state trajectories revealed that these errors were mostly caused by incorrect forecasts of the operation mode, even when the temperature of the associated HWT was well within the acceptable range, and by incorrect classification of actions, deeming actions feasible even though they would allow the CHPP to turn on or off before the minimum dwell times have been reached. Since only a 1°C buffer was applied in Chapter 5 and the prediction quality of ANNs can vary throughout different parameter ranges, it is conceivable that a smaller buffer may lead to a better result. However, more thorough experiments need to be conducted in order to come to a final conclusion.

Overall, the experiment shows that the state-based simulation approach can indeed be used to search for load schedules in a DSM application. Almost all generated schedules were truly feasible and hence the found solutions could mostly be implemented by the flexibility providers. The presented algorithms are a first step in the direction of practical DSM applications. Further research identifying better algorithms is needed, as there is potential for improvement, not only in terms of better algorithms, but also in terms of a computationally more efficient implementation.

6.5 Summary

Not every surrogate modeling approach is fitting for every use case, as the discussion of the exemplary optimization problems *Problem I* and *Problem II* in this chapter illustrates. Generally, approaches of the *signal selection* category are more limited in their use than those of the *schedule generation* kind. Whether an approach is suitable or not, needs to be assessed on a per-case basis, but even within the *schedule generation* class, there are differences. The examples and Table 6.2 show that the *state-based simulation* approach is amidst the most versatile ones. Furthermore, the conducted experiment demonstrated that the approach can be used to implement DSM applications.

All approaches have in common that they come at the high cost of having to train ANNs. Nonetheless, even though this is resource intensive, it is only required once every time the configuration of modeled DERs changes. Other surrogate modeling approaches, like for instance the SVDD based decoder [11], need a newly generated surrogate in every new state. In order to conduct the ensemble scheduling experiment from the previous section with 10 flexibility providers and 100 target schedules, one needs 1,000 SVDD models, but only 10 state-based simulation surrogates with a total of 20 ANNs. Furthermore, in a practical application, it is possible to make use of transfer learning (see [27] on the topic of transfer learning), which starts the training process with an ANN trained for a similar application and may lead to better results within short time compared to learning an ANN from scratch. This has the potential to greatly reduce the required amount of computations.

7 Conclusion and Outlook

The exploitation of flexibility, especially on the former demand side, is crucial for the reliable operation of our future energy system. In this thesis, the exploitation in general is analyzed and ANN-based surrogate modeling approaches are derived and assessed.

Exploitation of flexibility: There exist various mechanisms to achieve control of DERs and thereby exploit their flexibility. However, each mechanism has its own, distinct characteristics. Five general patterns for the exploitation of flexibility have been identified based on prior publications and an extensive literature review (see Section 4.2). The patterns can be used to classify and thereby compare different exploitation mechanisms on an abstract level.

A more detailed comparison is possible by additionally analyzing the employed types of models. With the help of the aforementioned literature review, a set of different general modeling approaches has been derived and related to the exploitation patterns, all while keeping the goal of communicating flexibility in mind (see Section 4.3). The results show that the modeling approaches — and therefore the ways of communicating flexibility — vary from exploitation pattern to exploitation pattern. While for the direct exploitation DERs are modeled individually, in the abstract and market-based patterns devices are clustered. In the indirect and state information based patterns, models can provide a huge variety of information, depending on the exact implementation. In addition, the abstractness and genericness of the individual modeling has been assessed. Therefore, it is possible to compare the genericness of selected exploitation mechanisms, by relating the employed models to the identified general modeling approaches. Furthermore, by doing so, potential similar and alternative solutions become apparent. Overall, the identified categories and relationships provide a general framework for analyzing exploitation mechanisms and their models.

There are many trade-offs to consider when developing a mechanism for the exploitation of flexibility and specifying a model. Depending on the exact use case, each have their own advantages and disadvantages. The systematics developed and presented in this thesis can aid in the conception of new exploitation mechanisms by providing an overview of solutions and pointing out typical modeling approaches. Furthermore, the review itself lists plenty of examples for the different types of models.

ANN-based surrogate models: In our prior publications we already showed that ANNs are capable of learning the flexibility of DERs. The general approach of utilizing ANN-based surrogate models to learn and describe the flexibility of DERs is further concretized in this thesis. By asking which information is needed to control DERs and analyzing the previously identified exploitation patterns and modeling approaches in this regard, multiple potential ANN-based surrogates have been identified and outlined (see Section 5.2). A selection of the most promising approaches has then been compiled and presented. While the majority of the selected models is intended for the generation of load schedules, the remaining models provide information for deriving specific signals, such as tariffs or curtailment signals, required only in particular use cases.

One of the identified ANN-based surrogate modeling approaches is the *state-based simulation*, which is a learned state space model augmented with a classifier for identifying feasible control inputs. Due to its design and the large amount of information it provides, it is one of the most versatile approaches to encode the flexibility of DERs into ANNs. For this reason, this approach has been selected and evaluated in order to establish a baseline for the achievable quality of approximation provided by ANN-based flexibility surrogates (see Section 5.3). The results provide further proof that it is indeed possible to encode the flexibility of DERs into ANNs. A total of six DER configurations, ranging from individual DERs to a large ensemble consisting of 100 BESSs, has been tested and a multitude of criteria has been assessed. The criteria include the share of truly feasible load schedules generated with the help of the surrogate, as well as the classification performance, which is used to analyze whether the models are able to capture the entire space of feasible schedules. The results varied from configuration to configuration. An increasing number of DERs and more complex constraints generally resulted in less accurate models. One method to improve these results is by applying buffers during the training and relaxing the constraints during application. While this narrows down the space of feasible schedules and thus reduces the overall exploitable flexibility, it greatly increases the likelihood of generating feasible schedules. Overall, the trained surrogates performed better in producing feasible schedules than they did in recognizing feasible schedules. This was likely caused by overfitted models. Better generalizing surrogates probably would exhibit a better classification performance, at the cost of a worse generation performance. However, whether a larger percentage of feasible schedules or a better description of the true flexibility is more important must be assessed in the context of a specific use case.

Controlling DERs with the help of ANN-based surrogates: The suitability of the identified ANN-based surrogate modeling approaches has been assessed in a qualitative analysis (see Section 6.1 to Section 6.3). This analysis outlines characteristics that have to be considered when selecting one of the approaches for a practical implementation. Furthermore, it is proposed how control signals could be derived with the help of the individual types of surrogate models. Overall, these results can provide guidance for future research in the field of ANN-based flexibility surrogates.

Additionally, quantitative experiments have been conducted for the *state-based simulation* approach (see Section 6.4). In each experiment, load schedules for ensembles of 10 identical DERs have been generated with the goal of reproducing a given, feasible, aggregated target schedule. The results show that the great majority of schedules generated in the optimization process was feasible. They also show that the findings of the previous analysis hold up in a practical scheduling task and thereby demonstrate the potential of ANN-based surrogates. Nevertheless, further research in this area is required in order to develop the necessary algorithms and evaluate the approach in a real DSM application.

In all experiments, the schedules have been generated once and then the resulting total deviation from the target has been measured. However, in a real application, an MPC-based control scheme would likely be implemented, which means that in each time period new schedules are generated by the external entity and distributed to the flexibility providers. Especially in combination with MPC, the investigated surrogates have the potential to perform well, since they showed very good performance for short time horizons. Focusing on short time horizons may also lower the computational complexity, as a shallower exploration of the state space may suffice. Nevertheless, whether this is the case or not remains a question for future research. Additionally, and by no means limited to MPC, flexibility providers could provide feedback in case a requested schedule is not feasible.

Public materials: All source code, including the implemented simulation and optimization models, the trained ANNs, and the logs from the training and evaluation runs have been published on GitHub at `https://github.com/kfoerderer/ANN-based-surrogates` and, together with this document, in the KITopen repository.

7.1 Research Questions Briefly Answered

In the following, the research questions introduced in Chapter 1 are explicitly and briefly answered. Please see Chapter 4, Chapter 5, and Chapter 6, for detailed analyses and results regarding **RQ1**, **RQ2**, and **RQ3**, as well as the individual subquestions.

RQ1 *Which models can be used for communicating flexibility?*
Whether a type of model is suitable or not, depends on the specific exploitation pattern. A set of general modeling approaches has been derived with the help of a literature review and related to the individual patterns. They range from models precisely formulated for individual DERs, to coordination schemes that do not even rely on the communication of models. See Figure 4.3 for an overview.

RQ1.1 *What are the motivations for communicating flexibility?*
The primary motivations identified are monetary incentives and the qualitative security of supply. However, many further motivations may exist, including environmental incentives, legal compulsion, and the visualization of flexibility.

RQ1.2 *How can flexibility be communicated and exploited?*
By making use of one or multiple exploitation patterns and the associated modeling approaches.

RQ1.3 *Which modeling approaches do exist and how can they be categorized?*
A set of general modeling approaches has been derived with the help of the mentioned literature review and related to the individual patterns. See Figure 4.3 for an overview.

RQ2 *What is the quality of approximation ANN-based surrogate models can achieve?*
There are different ways of encoding the flexibility of DERs into ANNs, of which many are derived and outlined in this thesis. Not all of them are directly comparable. Hence, it is only possible to establish a baseline for the achievable quality. One of the more versatile ANN-based surrogates, namely the *state-based simulation* approach, has been assessed in this thesis. It performed well in the load schedule generation tasks, yielding high percentages of feasible schedules. However, for more complex DERs and ensembles of DERs buffers are required. Furthermore, the true flexibility was only partially captured by in the experiments. There probably exist ANN-based surrogates exhibiting a better performance for different use cases.

RQ2.1 *What are the advantages and disadvantages of using surrogate models?*
Surrogate models can be generated from data in an automated process. Given a suitable data basis, they may even be generated from recorded data, potentially eliminating the need of manual modeling. Furthermore, they may act as generic models for the flexibility of all types or combinations of DERs.

RQ2.2 *How can ANNs serve as surrogates for the flexibility of DERs?*
A variety of approaches has been derived and outlined in this thesis. The most promising ones are listed in Table 5.1.

RQ2.3 *What is the quality of the trained ANN-based surrogate models?*
As explained earlier, the different approaches are not all directly comparable. A baseline has been established, using the *state-based simulation* approach. It performed well in the load schedule generation tasks, yielding high percentages of feasible schedules when buffers are used. The true flexibility was only partially captured by in the experiments. For different use cases, other ANN-based surrogate modeling approaches may perform even better.

RQ3 *How adequate are the different ANN-based modeling approaches for controlling schedules of flexibility providers?*
Since the individual surrogate-modeling approaches provide different kinds of data, they are generally limited to specific use cases. In theory, each identified surrogate is able to act as a model for the flexibility of DERs, as long as the considered use case is suitable and the quality of the trained model is sufficiently high. However, the utilization of ANN-based surrogates introduces high computational complexity required for training and utilizing the models. To compare the individual characteristics, a qualitative assessment of the identified approaches has been conducted. Furthermore, for the *state-based simulation* approach, a first quantitative evaluation is provided, demonstrating the general possibility to use such surrogates in scheduling tasks. Nevertheless, for a real application further improvements and performance optimizations are required.

RQ3.1 *Which criteria can be used for assessing the adequacy of a modeling approach?*
Due to the different nature of each surrogate modeling approach, only a few qualitative criteria are applicable for a comparison of all of them. After considering different criteria from the field of benchmarking and meta heuristics, the focus was laid on the computational complexity and scalability.

RQ3.2 *What is the assessment of the individual ANN-based modeling approaches?*
In order to assess an approach, it is necessary to specify how the exploitation of flexibility should be achieved. Depending on the algorithm built around the specific ANN-based surrogate modeling approach, the assessment can vary massively. A summary of the proposed methodology for deriving control signals with the help of each surrogate and the resulting assessments can be found in Table 6.1 and Table 6.2. Additionally, in a series of experiments it has been demonstrated that the *state-based simulation* approach could be used by an external entity to solve scheduling tasks. However, further research on better algorithms is required.

7.2 Outlook

When we first proposed the utilization of ANNs as surrogate models for the flexibility of DERs, we proposed five different approaches **[313]**. Later we identified a sixth option **[308]**. Building upon these, in this thesis, a total of 10 promising approaches is named, each with its own characteristics. Further research is needed to assess each single option, preferably in practical use cases. Moreover, even though a more practical test of the *state-based simulation* approach has been conducted in this thesis, only a first step towards a real use case has been made. Further work on the training and selection of the best models, as well as better performing optimization algorithms are required. For instance, during the training process, training samples could be generated and selected intelligently, instead of purely random.

ANN-based flexibility surrogates are statistical models and their output can generally be seen as a set of predictions. Utilizing suitable ANN output dimensions and loss functions, it is possible to predict entire distributions instead of single data points. This ability of ANNs is also used in this thesis in order to predict discrete state variables by selecting the mode of the predicted distribution, i.e., the point with the highest probability. In future research, the entire stochastic information could be used in order to make the models more robust. Moreover, all outputs could be modeled as distributions.

Other aspects that should be evaluated are the training using only recorded data and the interplay of the *state-based simulation* approach with model-based RL. Learning a flexibility model from recorded data is one of the primary requirements for being able to generate them in an automated process and without the need of manual modeling. The major challenges, however, are the limited amount of data and the need of data that reflects the true flexibility. This is related to the challenge of exploring the state space in RL. ANN-based flexibility surrogates are especially a promising concept when model-based RL is employed by potential flexibility providers in order to control their DERs, since the learned model may be re-used. Future research should hence investigate the connection of the *state-based simulation* approach and model-based RL more closely.

A Appendix

A.1 Ideas for Learning Abstract Representations

Generative models like variational autoencoders (VAE, see [301, 27]) or generative adversarial networks (GAN, see [302]) are able to transform random inputs into synthetic samples. By training such a model, one does not only receive a mapping from some random representation to a synthetic sample, but also inputs which can encode distinct features, similar to genotypes. In [303], this has been demonstrated for a GAN which generates pictures of faces. By averaging the inputs for male faces with glasses, male faces without glasses, and female faces without glasses respectively, the authors created three average representations encoding the concepts male, female, and glasses. In the next step, they simply computed "(male with glasses) − (male without glasses) + (female without glasses)", and the result generated by the GAN was indeed a female face with glasses. Several further examples for generative models can be found in [27]. In the following it is outlined how VAEs and GANs, selected as exemplary approaches due to their popularity, could be used as surrogates. This may also be seen as a blueprint for implementing other generative models, as they could be utilized in a similar fashion. The generative models discussed here are both based on ANNs. Therefore they come with all the benefits and challenges of ANNs, including the need for huge amounts of data for training. Hence, to implement these approaches, it is still necessary for the flexibility provider to have access to large sets of feasible load schedules. If the historic data does not suffice, be it too few or too homogenous samples, and dataset augmentation cannot help, synthetic load schedules need to be generated. This task could be performed by the flexibility provider's EMS. Furthermore, if it is not intended to create a new surrogate every time the external entity requires an updated description of the flexibility, the models need to be able to take the current state into account (compare Figure 4.2). For both, VAE and GAN, there are extended versions which take an additional condition as an input, namely the conditional VAE [304] and the conditional GAN [305]. For modeling flexibility, this condition is the current state, as it defines which load schedules are feasible and which are not. The current state comprises all data necessary to derive whether a load schedule is feasible or not. This can also include forecasts, e.g., for the consumption of electricity or the availability of an EV.

A conditional VAE is depicted in Figure A.1. Variables known during the individual phases, like x and y in the training process, are illustrated in green color. The resulting model, which can be used as a surrogate, is highlighted in orange. Aside from the usual input y, which is encoded and decoded, an additional input x is introduced. This x is the condition, and passed to the encoder and decoder. During the training, the encoder is estimating the parameters required for sampling from the distribution $q_\Phi(z|x, y)$, which approximates the intractable posterior $p_\Theta(z|x, y)$ [301]. With these parameters z can be sampled and passed to the decoder, which is meant to reconstruct the original y by drawing from the distribution $p_\Theta(y|x, z)$. In order to generate a synthetic output with a VAE, a sample of z is required. Which distribution z has to be drawn from is dictated by the loss function used to train the VAE. Figure A.1 depicts the choice of a standard normal distribution, but many other options are available, as [301] outlines. For the purpose of modeling flexibility the VAE learns to generate feasible load schedules conditioned on the state. This model is trained by the flexibility provider and can be passed to the external entity. Given the current state and the distribution z is sampled from, new load schedules can be generated. As the model is trained with feasible load schedules, the syntheticaly generated load schedules should be likely to be feasible as well. Nevertheless, there is no guarantee for feasibility. Different options to deal with this uncertainty, such as conducting multiple iterations with feedback from the flexibility provider, or using a secondary model, have already been named in the introduction to Section 5.2.

A GAN, like a VAE, requires the training of two individual parts, a generator and a discriminator. Similar to the conditional VAE, the conditional GAN integrates a condition into the GAN by simply passing it as an additional input to the two respective models. Such a conditional GAN is depicted in Figure A.2. The notation is similar to [305], but has been adapted so it harmonizes with that of the conditional VAE presented in the previous paragraph. Information known during the individual phases is again highlighted in green, and the resulting surrogate in orange color. Like the standard GAN the conditional GAN takes some random input z, for instance Gaussian noise, generates a synthetic sample y, and passes it to a classifier which tries to distinguish synthetic from real samples. The additional condition is x. In order to use such a conditional GAN as a model for flexibility, it could be trained to generate load schedules conditioned on the current state. Here again, the flexibility provider trains both models and then transmits the generator part to the external entity. Using this generator model, and knowing the distribution of the noise z, the external entity is able to generate new load schedules. This conditional GAN can be seen as a combination of the generator and classificator approach we evaluated in **[312]**. As the approaches themselves produced good results on the test sets in **[312]**, it is conceivable that the combination of both in the form of a conditional GAN does as well. Especially the classifier, that is, the discriminator in the GAN, showed to be very accurate in the identification of feasible load schedules. Hence, the generator of the GAN must learn to produce feasible load profiles in order to beat the discrimnator. Furthermore, as stated before with the example of male and female faces, it has been shown that the inputs z of a trained GAN do encode concepts relevant to the synthetic output.

These are only two of many more options for generative models that could be employed to generate load schedules from some representation. Others could be implemented similarly in order to reach the overall goal of modeling flexibility. Whether the state as an input is required or not depends on whether the surrogate should be able to serve as a general model for the flexibility or only represent a temporary snapshot. In summary, generative models provide an approach for avoiding the development of abstract representations, similar to genotypes of EAs, and mappings from these representations to load schedules.

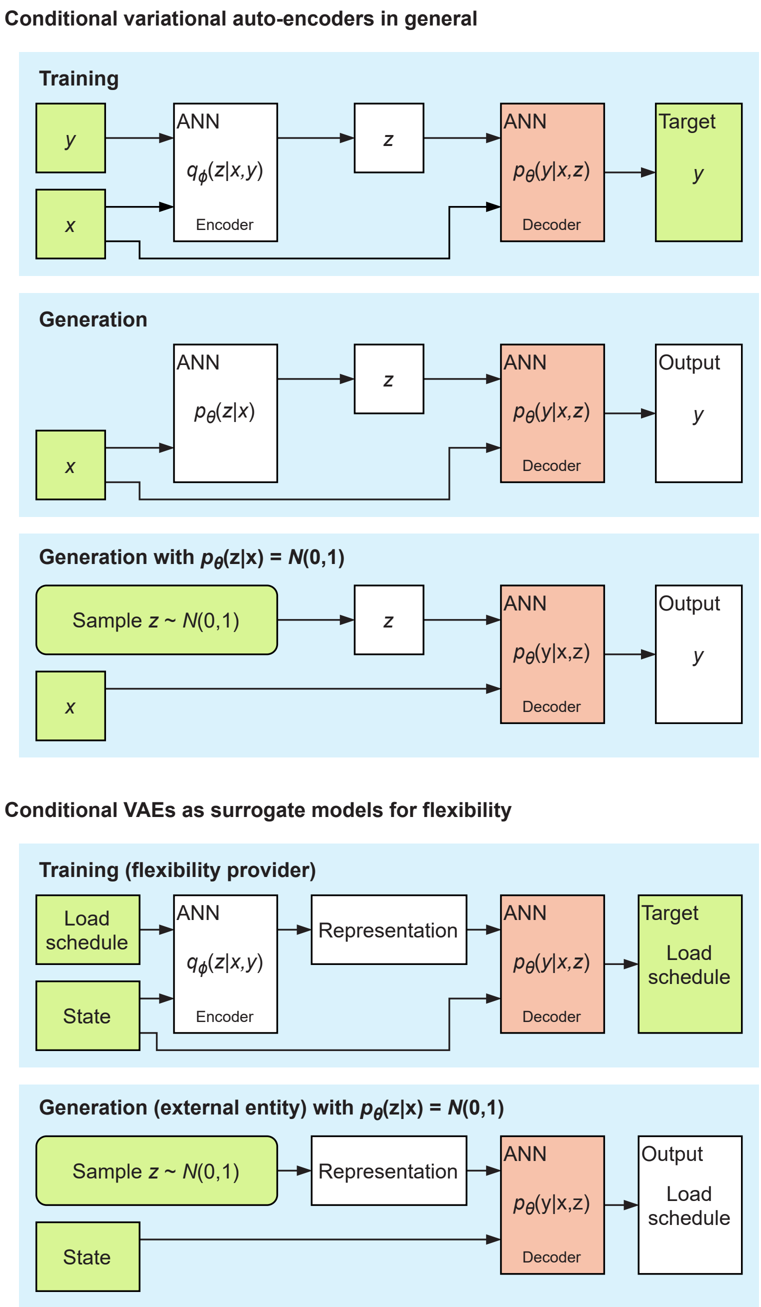

Figure A.1: Generation of load schedules using a conditional VAE, with notation according to [304]. Variables highlighted in green color are known prior to evaluating the ANN outputs. The resulting surrogate model is highlighted in orange.

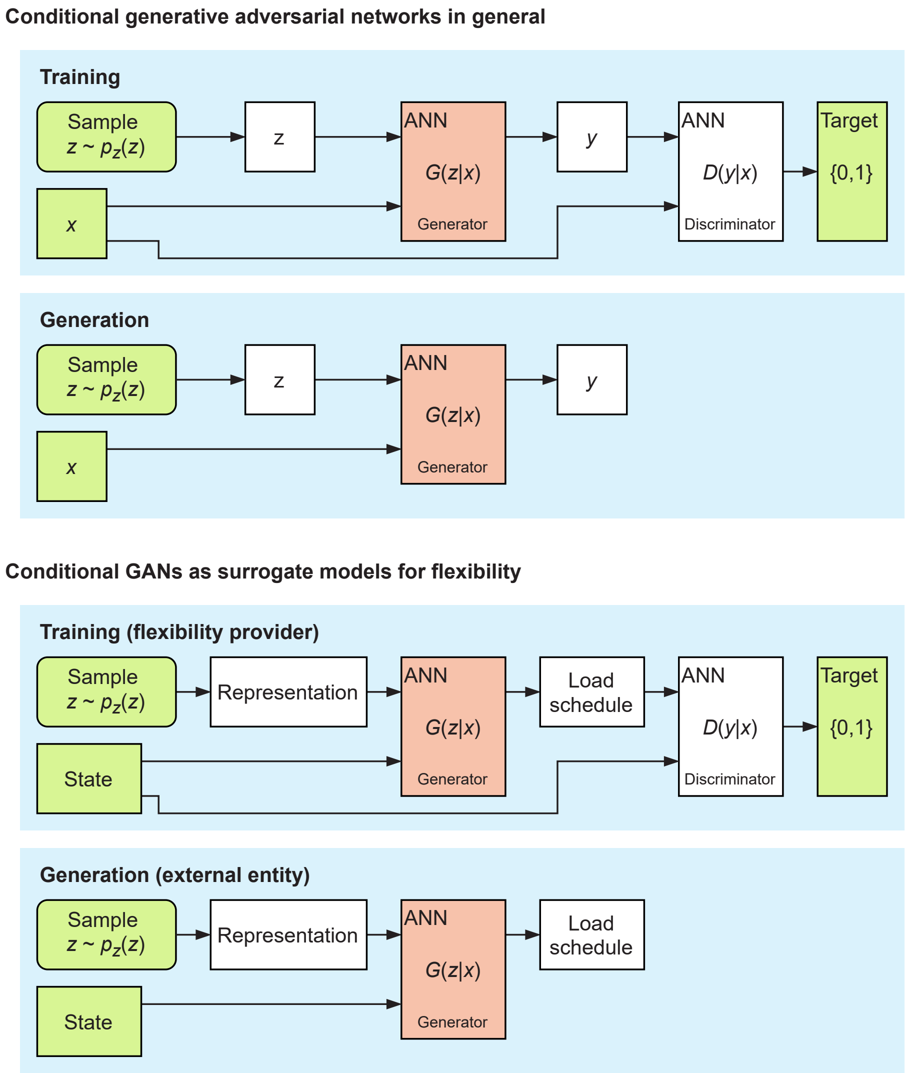

Figure A.2: Generation of load schedules using a conditional GAN, with notation similar to [305]. Variables highlighted in green color are known prior to evaluating the ANN outputs. The resulting surrogate model is highlighted in orange.

Bibliography

[1] European Commission, Secretariat-General. *COMMUNICATION FROM THE COMMISSION The European Green Deal COM/2019/640 final*. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM:2019:640:FIN. Accessed on: 2021/01/11. 2019.

[2] *Time series for the development of renewable energy sources in Germany based on statistical data from the Working Group on Renewable Energy-Statistics (AGEE-Stat)(Status: February 2021)*. https://www.erneuerbare-energien.de/EE/Navigation/DE/Service/Erneuerbare_Energien_in_Zahlen/Zeitreihen/zeitreihen.html. Accessed on: 2021/05/06. 2021.

[3] *Gesetz für den Ausbau erneuerbarer Energien (Erneuerbare-Energien-Gesetz - EEG 2021)*. https://www.gesetze-im-internet.de/eeg_2014/. Accessed on: 2021/01/08. 2020.

[4] Birgit Haller, Ole Langniß, Albrecht Reuter, and Nicolas Spengler, eds. *1,5°Csellsius: Energiewende zellulär - partizipativ - vielfältig umgesetzt*. Smart Grids-Plattform, 2020. ISBN: 978-3-00-067236-1.

[5] Ingo Mauser. "Multi-modal Building Energy Management". PhD thesis. Karlsruher Institut für Technologie (KIT), 2017. 487 pp. DOI: 10.5445/IR/1000070625.

[6] Yann Rebours and Daniel Kirschen. *A Survey of Definitions and Specifications of Reserve Services*. Tech. rep. Release 2. University of Manchester, 2005.

[7] European Commission, Directorate-General for Energy. *COMMISSION REGULATION (EU) 2017/1485 of 2 August 2017 establishing a guideline on electricity transmission system operation (Text with EEA relevance)*. 2017.

[8] DIN Deutsches Institut für Normung e.V. *Voltage characteristics of electricity supplied by public electricity networks; German version EN 50160:2010 + Cor.:2010 + A1:2015 + A2:2019 + A3:2019*. Beuth Verlag GmbH, 2020.

[9] European Commission. *COMMISSION REGULATION (EU) 2016/631 of 14 April 2016 establishing a network code on requirements for grid connection of generators (Text with EEA relevance)*. 2016.

[10] Hal R. Varian. *Intermediate microeconomics : a modern approach*. 8. ed. New York, NY [u.a.]: Norton, 2010. ISBN: 9780393935332; 9780393934243.

[11] Jörg Bremer. "Constraint-Handling mit Supportvektor-Dekodern in der verteilten Optimierung". PhD thesis. Ammerländer Heerstraße 114-118, 26129 Oldenburg, Germany: Carl von Ossietzky Universität Oldenburg, Feb. 2015.

[12] Judith Neugebauer, Oliver Kramer, and Michael Sonnenschein. "Classification Cascades of Overlapping Feature Ensembles for Energy Time Series Data". In: *Data Analytics for Renewable Energy Integration*. Ed. by Wei Lee Woon, Zeyar Aung, and Stuart Madnick. Cham: Springer International Publishing, 2015, pp. 76–93. ISBN: 978-3-319-27430-0.

[13] Helen Sawall., Andreas Scheuriker., and Daniel Stetter. "Flexibility Definition for Smart Grid Cells in a Decentralized Energy System". In: *Proceedings of the 7th International Conference on Smart Cities and Green ICT Systems - Volume 1: SMARTGREENS,* INSTICC. SciTePress, 2018, pp. 130–139. ISBN: 978-989-758-292-9. DOI: 10.5220/0006803401300139.

[14] Ricardo J. Bessa, David Rua, Cláudia Abreu, Paulo Machado, José R. Andrade, Rui Pinto, Carla Gonçalves, and Marisa Reis. "Data Economy for Prosumers in a Smart Grid Ecosystem". In: *Proceedings of the Ninth International Conference on Future Energy Systems*. e-Energy '18. Karlsruhe, Germany: Association for Computing Machinery, 2018, pp. 622–630. ISBN: 9781450357678. DOI: 10.1145/3208903.3210282.

[15] L. Zhao, W. Zhang, H. Hao, and K. Kalsi. "A Geometric Approach to Aggregate Flexibility Modeling of Thermostatically Controlled Loads". In: *IEEE Transactions on Power Systems* 32.6 (2017), pp. 4721–4731. DOI: 10.1109/TPWRS.2017.2674699.

[16] International Electrotechnical Commission. *International electrotechnical vocabulary*. http://www.electropedia.org. Accessed on: 2021/01/20.

[17] Elisabeth Drayer. "Resilient Operation of Distribution Grids with Distributed-Hierarchical Architecture". PhD thesis. University of Kassel, 2018. DOI: 10.19211/KUP9783737605939.

[18] Fredy Hernan Rios Silva. "Stigmergy-based Load Scheduling in a Demand Side Management Context". PhD thesis. Karlsruher Institut für Technologie (KIT), 2016. DOI: 10.5445/IR/1000055801.

[19] Martin Ward. "A definition of abstraction". In: *Journal of Software Maintenance: Research and Practice* 7.6 (1995), pp. 443–450. DOI: 10.1002/smr.4360070606.

[20] Clive L. Dym. *Principles of Mathematical Modeling (Second Edition)*. Second Edition. Burlington: Academic Press, 2004. ISBN: 978-0-12-226551-8. DOI: https://doi.org/10.1016/B978-012226551-8/50004-1.

[21] B. Sohlberg and E.W. Jacobsen. "GREY BOX MODELLING – BRANCHES AND EXPERIENCES". In: *IFAC Proceedings Volumes* 41.2 (2008). 17th IFAC World Congress, pp. 11415–11420. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20080706-5-KR-1001.01934.

[22] Aurélie Foucquier, Sylvain Robert, Frédéric Suard, Louis Stéphan, and Arnaud Jay. "State of the art in building modelling and energy performances prediction: A review". In: *Renewable and Sustainable Energy Reviews* 23 (2013), pp. 272–288. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2013.03.004.

[23] Eleni I. Vlahogianni. "Optimization of traffic forecasting: Intelligent surrogate modeling". In: *Transportation Research Part C: Emerging Technologies* 55 (2015). Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue, pp. 14–23. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2015.03.016.

[24] Sun Hye Kim and Fani Boukouvala. "Machine learning-based surrogate modeling for data-driven optimization: a comparison of subset selection for regression techniques". In: *Optimization Letters* (2019). ISSN: 1862-4480. DOI: 10.1007/s11590-019-01428-7.

[25] Kevin McBride and Kai Sundmacher. "Overview of Surrogate Modeling in Chemical Process Engineering". In: *Chemie Ingenieur Technik* 91.3 (2019), pp. 228–239. DOI: 10.1002/cite.201800091.

[26] Dirk Gorissen, Ivo Couckuyt, Eric Laermans, and Tom Dhaene. "Multiobjective global surrogate modeling, dealing with the 5-percent problem". In: *Engineering with Computers* 26.1 (2010), pp. 81–98. ISSN: 1435-5663. DOI: 10.1007/s00366-009-0138-1.

[27] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[28] Thomas M. Mitchell. *Machine Learning*. 1st ed. USA: McGraw-Hill, Inc., 1997. ISBN: 0070428077.

[29] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning with Applications in R*. The address: Springer, New York, NY, 2017. ISBN: 978-1-4614-7138-7. DOI: 10.1007/978-1-4614-7138-7.

[30] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. "Searching for Activation Functions". In: *CoRR* abs/1710.05941 (2017).

[31] Aäron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. "Conditional Image Generation with PixelCNN Decoders". In: *CoRR* abs/1606.05328 (2016).

[32] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. "WaveNet: A Generative Model for Raw Audio". In: *CoRR* abs/1609.03499 (2016).

[33] Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Ryan Sepassi, George Tucker, and Henryk Michalewski. "Model-Based Reinforcement Learning for Atari". In: *CoRR* abs/1903.00374 (2019).

[34] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90020-8.

[35] G. Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4 (Dec. 1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274.

[36] Torch Contributors. *PyTorch documentation*. https://pytorch.org/docs/stable/index.html. Accessed on: 2021/01/28.

[37] Sergey Ioffe and Christian Szegedy. "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167 (2015).

[38] D.M.J. Tax. "One-class classification". PhD thesis. Netherlands: TU Delft, June 2001. ISBN: 90-75691-05-X.

[39] Bernhard Schölkopf, Robert C. Williamson, Alex Smola, and John Shawe-Taylor. *SV Estimation of a Distribution's Support*. NIPS 99. 1999.

[40] Robert P.W. Tax David M.J. and Duin. "Support Vector Data Description". In: *Machine Learning* 54.1 (Jan. 2004), pp. 45–66. ISSN: 1573-0565. DOI: 10.1023/B:MACH.0000008084.60811.49.

[41] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on Modern Convex Optimization*. Society for Industrial and Applied Mathematics, 2001. DOI: 10.1137/1.9780898718829.

[42] Robert J. Vanderbei. *Linear Programming : Foundations and Extensions*. 5th ed. 2020. Cham: Springer International Publishing, 2020. ISBN: 9783030394158.

[43] Christodoulos A. Floudas and V. Visweswaran. "Quadratic Optimization". In: *Handbook of Global Optimization*. Boston, MA: Springer US, 1995, pp. 217–269. ISBN: 978-1-4615-2025-2. DOI: 10.1007/978-1-4615-2025-2_5.

[44] Mark French. *Fundamentals of Optimization : Methods, Minimum Principles, and Applications for Making Things Better*. Cham: Springer, 2018. ISBN: 9783319761923.

[45] Agoston E. Eiben. *Introduction to Evolutionary Computing*. 2nd ed. 2015. Berlin, Heidelberg: Springer, 2015. ISBN: 9783662448748.

[46] Fabian Dunke. "Online Optimization with Lookahead". PhD thesis. 2014. DOI: 10.5445/IR/1000042132.

[47] Johannes Jahn. *Introduction to the Theory of Nonlinear Optimization*. 4th ed. 2020. Cham: Springer International Publishing, 2020. ISBN: 9783030427603.

[48] Oliver Stein. *Grundzüge der Nichtlinearen Optimierung*. Berlin, Heidelberg: Springer Spektrum, 2018. ISBN: 9783662555934.

[49] Jürgen Branke, Michael Stein, and Hartmut Schmeck. "A Unified Framework for Metaheuristics". In: *Genetic and Evolutionary Computation — GECCO 2003*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 1568–1569. ISBN: 978-3-540-45110-5.

[50] Charles Audet and Warren Hare. *Derivative-free and blackbox optimization*. 1st ed. Cham: Springer, 2017. ISBN: 978-3-319-68912-8. DOI: https://doi.org/10.1007/978-3-319-68913-5.

[51] Carlos Bordons. *Model Predictive Control of Microgrids*. Ed. by Félix Garcia-Torres and Miguel A. Ridao. 1st ed. 2020. Cham: Springer, 2020. ISBN: 9783030245702.

[52] Michael Lederman Littman. *Algorithms for sequential decision making*. Brown University Providence, RI, 1996.

[53] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*. The MIT Press and McGraw-Hill Book Company, 2001. ISBN: 0-262-03293-7.

[54] Richard S. [VerfasserIn] Sutton. *Reinforcement learning : an introduction*. Second edition. Adaptive computation and machine learning. Cambridge, Massachusetts: The MIT Press, [2018]. ISBN: 9780262039246.

[55] Sergey Levine. *UC Berkeley CS 294-112, Lecture Slides: Deep Reinforcement Learning*. http://rail.eecs.berkeley.edu/deeprlcourse-fa18/. Accessed on: 2021/02/10. 2018.

[56] A. M. Kosek, G. T. Costanzo, H. W. Bindner, and O. Gehrke. "An overview of demand side management control schemes for buildings in smart grids". In: *2013 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*. Aug. 2013, pp. 1–9. DOI: 10.1109/SEGE.2013.6707934.

[57] Solène Goy and Donal Finn. "Estimating Demand Response Potential in Building Clusters". In: *Energy Procedia* 78 (2015). 6th International Building Physics Conference, IBPC 2015, pp. 3391–3396. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2015.11.756.

[58] Abdul Afram and Farrokh Janabi-Sharifi. "Review of modeling methods for HVAC systems". In: *Applied Thermal Engineering* 67.1 (2014), pp. 507–519. ISSN: 1359-4311. DOI: https://doi.org/10.1016/j.applthermaleng.2014.03.055.

[59] Lukáš Ferkl and Jan Široký. "Ceiling radiant cooling: Comparison of ARMAX and subspace identification modelling methods". In: *Building and Environment* 45.1 (2010). International Symposium on the Interaction between Human and Building Environment Special Issue Section, pp. 205–212. ISSN: 0360-1323. DOI: https://doi.org/10.1016/j.buildenv.2009.06.004.

[60] Katarzyna M. Luc, Alfred Heller, and Carsten Rode. "Energy demand flexibility in buildings and district heating systems – a literature review". In: *Advances in Building Energy Research* 13.2 (2019), pp. 241–263. DOI: 10.1080/17512549.2018.1488615.

[61] Philipp Ringler, Dogan Keles, and Wolf Fichtner. "Agent-based modelling and simulation of smart electricity grids and markets – A literature review". In: *Renewable and Sustainable Energy Reviews* 57 (2016), pp. 205–215. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2015.12.169.

[62] Jörg Bremer, Barbara Rapp, and Michael Sonnenschein. "Support vector based encoding of distributed energy resources' feasible load spaces". In: *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*. 2010, pp. 1–8. DOI: 10.1109/ISGTEUROPE.2010.5638940.

[63] J. Bremer, B. Rapp, and M. Sonnenschein. "Encoding distributed search spaces for virtual power plants". In: *2011 IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*. 2011, pp. 1–8. DOI: 10.1109/CIASG.2011.5953329.

[64] J. Bremer and M. Sonnenschein. "A distributed greedy algorithm for constraint-based scheduling of energy resources". In: *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 2012, pp. 109–115.

[65] A. Nieße, M. Sonnenschein, C. Hinrichs, and J. Bremer. "Local soft constraints in distributed energy scheduling". In: *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 2016, pp. 1517–1525.

[66] Judith Neugebauer, Oliver Kramer, and Michael Sonnenschein. "Improving Cascade Classifier Precision by Instance Selection and Outlier Generation". In: *Proceedings of the 8th International Conference on Agents and Artificial Intelligence*. ICAART 2016. Rome, Italy: SCITEPRESS - Science and Technology Publications, Lda, 2016, pp. 96–104. ISBN: 9789897581724. DOI: 10.5220/0005702100960104.

[67] J. Neugebauer, J. Bremer, C. Hinrichs, O. Kramer, and M. Sonnenschein. "Generalized cascade classification model with customized transformation based ensembles". In: *2016 International Joint Conference on Neural Networks (IJCNN)*. 2016, pp. 4056–4063. DOI: 10.1109/IJCNN.2016.7727727.

[68] Judith Neugebauer, Oliver Kramer, and Michael Sonnenschein. "Instance Selection and Outlier Generation to Improve the Cascade Classifier Precision". In: *Agents and Artificial Intelligence: 8th International Conference, ICAART 2016, Rome, Italy, February 24-26, 2016, Revised Selected Papers*. Cham: Springer International Publishing, 2017, pp. 151–170. ISBN: 978-3-319-53354-4. DOI: 10.1007/978-3-319-53354-4_9.

[69] Joerg Bremer. and Sebastian Lehnhoff. "A Cascading Chi-shapes based Decoder for Constraint-handling in Distributed Energy Management". In: *Proceedings of the 10th International Joint Conference on Computational Intelligence - Volume 1: IJCCI,* INSTICC. SciTePress, 2018, pp. 184–191. ISBN: 978-989-758-327-8. DOI: 10.5220/0006926101840191.

[70] Yaochu Jin. "Surrogate-assisted evolutionary computation: Recent advances and future challenges". In: *Swarm and Evolutionary Computation* 1.2 (2011), pp. 61–70. ISSN: 2210-6502. DOI: https://doi.org/10.1016/j.swevo.2011.05.001.

[71] Nicole Ludwig, Simon Waczowicz, Ralf Mikut, and Veit Hagenmeyer. "Mining Flexibility Patterns in Energy Time - Series from Industrial Processes". In: *Proceedings. 27. Workshop Computational Intelligence, Dortmund, 23. - 24. November 2017. Hrsg.: F. Hoffmann*. KIT Scientific Publishing, 2017, pp. 13–32. ISBN: 978-3-7315-0726-0.

[72] Nicole Nadine Ludwig. "Data-Driven Methods for Demand-Side Flexibility in Energy Systems". PhD thesis. Karlsruher Institut für Technologie (KIT), 2020. 176 pp. DOI: 10.5445/IR/1000123985.

[73] José R. Vázquez-Canteli and Zoltán Nagy. "Reinforcement learning for demand response: A review of algorithms and modeling techniques". In: *Applied Energy* 235 (2019), pp. 1072–1089. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.11.002.

[74] R. E. Swaney and I. E. Grossmann. "An index for operational flexibility in chemical process design. Part I: Formulation and theory". In: *AIChE Journal* 31.4 (1985), pp. 621–630. DOI: https://doi.org/10.1002/aic.690310412.

[75] A. F. Hernandez and M. G. Gallivan. "An exploratory study of discrete time state-space models using kriging". In: *2008 American Control Conference*. 2008, pp. 3993–3998. DOI: 10.1109/ACC.2008.4587117.

[76] M.R.M. Cruz, D.Z. Fitiwi, S.F. Santos, and J.P.S. Catalão. “A comprehensive survey of flexibility options for supporting the low-carbon energy future”. In: *Renewable and Sustainable Energy Reviews* 97 (2018). cited By 11, pp. 338–353. DOI: 10.1016/j.rser.2018.08.028.

[77] Andreas Ulbig and Göran Andersson. “Analyzing operational flexibility of electric power systems”. In: *2014 Power Systems Computation Conference*. 2014, pp. 1–8. DOI: 10.1109/PSCC.2014.7038383.

[78] Gordon Walker. “What are the barriers and incentives for community-owned means of energy production and use?” In: *Energy Policy* 36.12 (2008). Foresight Sustainable Energy Management and the Built Environment Project, pp. 4401–4405. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2008.09.032.

[79] Laurynas Šikšnys, Torben Bach Pedersen, Muhammad Aftab, and Bijay Neupane. “Flexibility Modeling, Management, and Trading in Bottom-up Cellular Energy Systems”. In: *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. e-Energy ’19. Phoenix, AZ, USA: Association for Computing Machinery, 2019, pp. 170–180. ISBN: 9781450366717. DOI: 10.1145/3307772.3328296.

[80] Samuel Gyamfi and Susan Krumdieck. “Price, environment and security: Exploring multi-modal motivation in voluntary residential peak demand response”. In: *Energy Policy* 39.5 (2011), pp. 2993–3004. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2011.03.012.

[81] C.M. Flath and S. Gottwalt. “Price-based load coordination revisited: augmenting open-loop coordination approaches”. In: *Business Research* 9.1 (2016), pp. 157–178. DOI: 10.1007/s40685-016-0033-8.

[82] G. Singh Aujla, S. Garg, S. Batra, N. Kumar, I. You, and V. Sharma. “DROpS: A demand response optimization scheme in SDN-enabled smart energy ecosystem”. In: *Information Sciences* 476 (2019), pp. 453–473. DOI: 10.1016/j.ins.2018.09.047.

[83] H. Hao, D. Wu, J. Lian, and T. Yang. “Optimal Coordination of Building Loads and Energy Storage for Power Grid and End User Services”. In: *IEEE Transactions on Smart Grid* 9.5 (2018), pp. 4335–4345. DOI: 10.1109/TSG.2017.2655083.

[84] Tobias Kornrumpf, Jan Meese, Markus Zdrallek, Nils Neusel-Lange, and Marvin Roch. “Economic dispatch of flexibility options for Grid services on distribution level”. In: *2016 Power Systems Computation Conference (PSCC)*. 2016, pp. 1–7. DOI: 10.1109/PSCC.2016.7540836.

[85] S. Rahnama, S.E. Shafiei, J. Stoustrup, H. Rasmussen, and J. Bendtsen. “Evaluation of aggregators for integration of large-scale consumers in smart grid”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*. Vol. 19. 2014, pp. 1879–1885.

[86] Friederike Wenderoth, Elisabeth Drayer, Robert Schmoll, Michael Niedermeier, and Martin Braun. “Architectural and functional classification of smart grid solutions”. In: *Energy Informatics* 2.1 (Sept. 2019), p. 33. ISSN: 2520-8942. DOI: 10.1186/s42162-019-0083-1.

[87] Jörg Bremer and Michael Sonnenschein. “Model-based Integration of Constrained Search Spaces into Distributed Planning of Active Power Provision.” In: *Computer Science & Information Systems* 10.4 (2013), pp. 1823–1854.

[88] Christina Süfke and Michael Tophinke. “Smart Grid Goes Flexible: Methods for Cost Reduction by Efficient Resource Utilization of Operating Equipment*”. In: *2018 International Conference on Smart Grid and Clean Energy Technologies (ICSGCE)*. 2018, pp. 43–48. DOI: 10.1109/ICSGCE.2018.8556794.

[89] J.S. Vardakas, N. Zorba, and C.V. Verikoukis. “Performance evaluation of power demand scheduling scenarios in a smart grid environment”. In: *Applied Energy* 142 (2015), pp. 164–178. DOI: 10.1016/j.apenergy.2014.12.060.

[90] F.J. Gomez, M. Aguilera Chaves, L. Vanfretti, and S.H. Olsen. "Multi-Domain Semantic Information and Physical Behavior Modeling of Power Systems and Gas Turbines Expanding the Common Information Model". In: *IEEE Access* 6 (2018), pp. 72663–72674. DOI: 10.1109/ACCESS.2018.2882311.

[91] L. Zhu, D. Shi, and X. Duan. "Standard function blocks for flexible IED in IEC 61850-Based substation automation". In: *IEEE Transactions on Power Delivery* 26.2 (2011), pp. 1101–1110. DOI: 10.1109/TPWRD.2010.2091154.

[92] Chen Zhenyu, Shi Kun, Li Dezhi, Chen Jingsheng, Lin Bin, and Li Yiyun. "Research on Information Exchange Model for Flexible Adjustment and Control of Large-scale Demand Response". In: *2018 8th International Conference on Electronics Information and Emergency Communication (ICEIEC)*. 2018, pp. 154–157. DOI: 10.1109/ICEIEC.2018.8473472.

[93] Wei Deng, Wei Pei, Ziqi Shen, Zhenxing Zhao, and Hui Qu. "Electric vehicle information exchange based on IEC 61850". In: *2014 IEEE Conference and Expo Transportation Electrification Asia-Pacific (ITEC Asia-Pacific)*. 2014, pp. 1–6. DOI: 10.1109/ITEC-AP.2014.6940888.

[94] W. Pei, W. Deng, Z. Shen, and Z. Qi. "Operation of battery energy storage system using extensional information model based on IEC 61850 for micro-grids". In: *IET Generation, Transmission and Distribution* 10.4 (2016), pp. 849–861. DOI: 10.1049/iet-gtd.2014.1123.

[95] N. Hargreaves. *Extension of IEC 61970 for electrical energy storage modelling*. 2014, pp. 135–160.

[96] Z. Pan, Q. Guo, and H. Sun. "Feasible region method based integrated heat and electricity dispatch considering building thermal inertia". In: *Applied Energy* 192 (2017), pp. 395–407. DOI: 10.1016/j.apenergy.2016.09.016.

[97] D. Wu, N. Radhakrishnan, and S. Huang. "A hierarchical charging control of plug-in electric vehicles with simple flexibility model". In: *Applied Energy* 253 (2019). DOI: 10.1016/j.apenergy.2019.113490.

[98] N. Sadeghianpourhamami, N. Refa, M. Strobbe, and C. Develder. "Quantitive analysis of electric vehicle flexibility: A data-driven approach". In: *International Journal of Electrical Power and Energy Systems* 95 (2018), pp. 451–462. DOI: 10.1016/j.ijepes.2017.09.007.

[99] V. Croce, M. Lazzaro, G. Paternò, D. Ziu, E. Riva Sanseverino, and A. Monti. "Smart district energy optimization of flexible energy units for the integration of local energy storage". In: *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*. 2017, pp. 1–6. DOI: 10.1109/EEEIC.2017.7977597.

[100] Fanxin Kong, Qiao Xiang, Linghe Kong, and Xue Liu. "On-Line Event-Driven Scheduling for Electric Vehicle Charging via Park-and-Charge". In: *2016 IEEE Real-Time Systems Symposium (RTSS)*. 2016, pp. 69–78. DOI: 10.1109/RTSS.2016.016.

[101] Chris Develder, Nasrin Sadeghianpourhamami, Matthias Strobbe, and Nazir Refa. "Quantifying flexibility in EV charging as DR potential: Analysis of two real-world data sets". In: *2016 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 2016, pp. 600–605. DOI: 10.1109/SmartGridComm.2016.7778827.

[102] R. Pedersen, J. Schwensen, S. Sivabalan, C. Corazzol, S.E. Shafiei, K. Vinther, and J. Stoustrup. "Direct control implementation of a refrigeration system in smart grid". In: *Proceedings of the American Control Conference*. 2013, pp. 3954–3959.

[103] A. Abiri-Jahromi and F. Bouffard. "Contingency-Type Reserve Leveraged Through Aggregated Thermostatically-Controlled Loads-Part I: Characterization and Control". In: *IEEE Transactions on Power Systems* 31.3 (2016), pp. 1972–1980. DOI: 10.1109/TPWRS.2015.2466175.

[104] Kirsten M. Nielsen, Palle Andersen, and Tom S. Pedersen. “Analysis of the storage capacity in an aggregated heat pump portfolio”. In: *2015 10th Asian Control Conference (ASCC)*. 2015, pp. 1–6. DOI: 10.1109/ASCC.2015.7244393.

[105] K. Paridari and L. Nordström. “Flexibility prediction, scheduling and control of aggregated TCLs”. In: *Electric Power Systems Research* 178 (2020). DOI: 10.1016/j.epsr.2019.106004.

[106] Lin Zhao and Wei Zhang. “A geometric approach to virtual battery modeling of thermostatically controlled loads”. In: *2016 American Control Conference (ACC)*. 2016, pp. 1452–1457. DOI: 10.1109/ACC.2016.7525121.

[107] K. Wang, L. Xu, W. Sun, Y. Wang, and L. Xu. “A simplified regression model for dynamic demand response process of building loads”. In: *IOP Conference Series: Materials Science and Engineering*. Vol. 486. 1. 2019. DOI: 10.1088/1757-899X/486/1/012106.

[108] B. Coffey, F. Haghighat, E. Morofsky, and E. Kutrowski. “A software framework for model predictive control with GenOpt”. In: *Energy and Buildings* 42.7 (2010), pp. 1084–1092. DOI: 10.1016/j.enbuild.2010.01.022.

[109] K. Wang, R. Yin, L. Yao, J. Yao, T. Yong, and N. Deforest. “A Two-Layer Framework for Quantifying Demand Response Flexibility at Bulk Supply Points”. In: *IEEE Transactions on Smart Grid* 9.4 (2018), pp. 3616–3627. DOI: 10.1109/TSG.2016.2636873.

[110] Mehdi Maasoumy, Catherine Rosenberg, Alberto Sangiovanni-Vincentelli, and Duncan S. Callaway. “Model predictive control approach to online computation of demand-side flexibility of commercial buildings HVAC systems for Supply Following”. In: *2014 American Control Conference*. 2014, pp. 1082–1089. DOI: 10.1109/ACC.2014.6858874.

[111] Thibault Péan, Jaume Salom, and Ramon Costa-Castelló. “Configurations of model predictive control to exploit energy flexibility in building thermal loads”. In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 3177–3182. DOI: 10.1109/CDC.2018.8619452.

[112] S. Huang and D. Wu. “Validation on aggregate flexibility from residential air conditioning systems for building-to-grid integration”. In: *Energy and Buildings* 200 (2019), pp. 58–67. DOI: 10.1016/j.enbuild.2019.07.043.

[113] Indrasis Chakraborty, Sai Pushpak Nandanoori, and Soumya Kundu. “Virtual Battery Parameter Identification Using Transfer Learning Based Stacked Autoencoder”. In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, pp. 1269–1274. DOI: 10.1109/ICMLA.2018.00206.

[114] D. Villa, C. Martín, F.J. Villanueva, F. Moya, and J.C. López. “A dynamically reconfigurable architecture for smart grids”. In: *IEEE Transactions on Consumer Electronics* 57.2 (2011), pp. 411–419. DOI: 10.1109/TCE.2011.5955174.

[115] Lukas Barth and Dorothea Wagner. “Shaving Peaks by Augmenting the Dependency Graph”. In: *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. e-Energy ’19. Phoenix, AZ, USA: Association for Computing Machinery, 2019, pp. 181–191. ISBN: 9781450366717. DOI: 10.1145/3307772.3328298.

[116] Nicole Ludwig, Lukas Barth, Dorothea Wagner, and Veit Hagenmeyer. “Industrial Demand-Side Flexibility: A Benchmark Data Set”. In: *Proceedings of the Tenth ACM International Conference on Future Energy Systems*. e-Energy ’19. Phoenix, AZ, USA: Association for Computing Machinery, 2019, pp. 460–473. ISBN: 9781450366717. DOI: 10.1145/3307772.3331021.

[117] R. Jovanovic, A. Bousselham, and I.S. Bayram. “Residential demand response scheduling with consideration of consumer preferences”. In: *Applied Sciences (Switzerland)* 6.1 (2016). DOI: 10.3390/app6010016.

[118] Mercedes Vallés, Antonio Bello, Javier Reneses, and Pablo Frías. “Probabilistic characterization of electricity consumer responsiveness to economic incentives”. In: *Applied Energy* 216 (2018), pp. 296–310. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.02.058.

[119] Johannes Gärttner, Christoph M. Flath, and Christof Weinhardt. “Load Shifting, Interrupting or Both? Customer Portfolio Composition in Demand Side Management”. In: *Computational Management Science*. Ed. by Raquel J. Fonseca, Gerhard-Wilhelm Weber, and João Telhada. Cham: Springer International Publishing, 2016, pp. 9–15. ISBN: 978-3-319-20430-7.

[120] L. Lin, X. Cai, B. Xu, and S. Xia. “Wind farm-LA coordinated operation mode and dispatch model in wind power accommodation promotion”. In: *Energies* 11.5 (2018). DOI: 10.3390/en11051227.

[121] Hanmin Cai, Andreas Hutter, Elisa Olivero, Pierre Roduit, and Pierre Ferrez. “Load shifting for tertiary control power provision”. In: *2015 IEEE 5th International Conference on Power Engineering, Energy and Electrical Drives (POWERENG)*. 2015, pp. 469–475. DOI: 10.1109/PowerEng.2015.7266363.

[122] Fatemeh Tahersima, Palle Andersen, and Per Printz Madsen. “Economic energy distribution and consumption in a microgrid Part1: Cell level controller”. In: *2013 IEEE International Conference on Control Applications (CCA)*. 2013, pp. 308–313. DOI: 10.1109/CCA.2013.6662776.

[123] M. Alcázar-Ortega, G. Escrivá-Escrivá, and I. Segura-Heras. “Methodology for validating technical tools to assess customer Demand Response: Application to a commercial customer”. In: *Energy Conversion and Management* 52.2 (2011), pp. 1507–1511. DOI: 10.1016/j.enconman.2010.10.018.

[124] D. Wang, S. Parkinson, W. Miao, H. Jia, C. Crawford, and N. Djilali. “Hierarchical market integration of responsive loads as spinning reserve”. In: *Applied Energy* 104 (2013), pp. 229–238. DOI: 10.1016/j.apenergy.2012.10.054.

[125] Liu Ming, Sun Haixia, Fu Xiangyun, Sun Haoran, Zhang Zhifu, and Cao Haimeng. “Research and application of interactive control technology of source and load in photovoltaic village”. In: *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*. 2017, pp. 1–6. DOI: 10.1109/EI2.2017.8245361.

[126] B. Geng, J.K. Mills, and D. Sun. “Two-stage charging strategy for plug-in electric vehicles at the residential transformer level”. In: *IEEE Transactions on Smart Grid* 4.3 (2013), pp. 1442–1452. DOI: 10.1109/TSG.2013.2246198.

[127] S. Nikolovski, H.R. Baghaee, and D. Mlakic. “ANFIS-Based Peak Power Shaving/Curtailment in Microgrids Including PV Units and BESSs”. In: *Energies* 11.11 (2018). DOI: 10.3390/en11112953.

[128] J. Lassila, J. Haakana, V. Tikka, and J. Partanen. “Methodology to analyze the economic effects of electric cars as energy storages”. In: *IEEE Transactions on Smart Grid* 3.1 (2012), pp. 506–516. DOI: 10.1109/TSG.2011.2168548.

[129] H. Chen, Z. Hu, H. Zhang, and H. Luo. “Coordinated charging and discharging strategies for plug-in electric bus fast charging station with energy storage system”. In: *IET Generation, Transmission and Distribution* 12.9 (2018), pp. 2019–2028. DOI: 10.1049/iet-gtd.2017.0636.

[130] Nahal Tamadon, Ebrahim Shayesteh, Marco Cupelli, and Antonello Monti. “Local Balancing of Low-Voltage Networks by Utilizing Distributed Flexibilities as Part of the InterFlex Field Trial”. In: *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*. 2018, pp. 3568–3575. DOI: 10.1109/IECON.2018.8591450.

[131] Mingchao Xia, Yuguang Song, and Qifang Chen. “The coordinated control of thermostatically controlled loads and energy storage considering the lock-out effect”. In: *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*. 2019, pp. 870–875. DOI: 10.1109/ISGT-Asia.2019.8880933.

[132] J. Li, Y. Fu, Z. Xing, X. Zhang, Z. Zhang, and X. Fan. “Coordination Scheduling Model of Multi-Type Flexible Load for Increasing Wind Power Utilization”. In: *IEEE Access* 7 (2019), pp. 105840–105850. DOI: 10.1109/ACCESS.2019.2932141.

[133] S. Gottwalt, J. Gärttner, H. Schmeck, and C. Weinhardt. “Modeling and valuation of residential demand flexibility for renewable energy integration”. In: *IEEE Transactions on Smart Grid* 8.6 (2017), pp. 2565–2574. DOI: 10.1109/TSG.2016.2529424.

[134] N. Good, E. Karangelos, A. Navarro-Espinosa, and P. Mancarella. “Optimization under Uncertainty of Thermal Storage-Based Flexible Demand Response with Quantification of Residential Users’ Discomfort”. In: *IEEE Transactions on Smart Grid* 6.5 (2015), pp. 2333–2342. DOI: 10.1109/TSG.2015.2399974.

[135] Q. Lyu, H. Gong, N. Yang, X. Xu, N. Zhang, and H. Wang. “An evaluation method ofwind power integration in power systems with flexible combined heat and power plant”. In: *Energies* 12.21 (2019). DOI: 10.3390/en12214129.

[136] Zhou Xia, Zhang Chaohai, Lu Zhenzhen, and Dai Yuchen. “Study of coordinating optimization for demand response participating in spinning reserve”. In: *2015 IEEE Innovative Smart Grid Technologies - Asia (ISGT ASIA)*. 2015, pp. 1–6. DOI: 10.1109/ISGT-Asia.2015.7387042.

[137] Lukas Barth, Veit Hagenmeyer, Nicole Ludwig, and Dorothea Wagner. “How Much Demand Side Flexibility Do We Need? Analyzing Where to Exploit Flexibility in Industrial Processes”. In: *Proceedings of the Ninth International Conference on Future Energy Systems*. e-Energy ’18. Karlsruhe, Germany: Association for Computing Machinery, 2018, pp. 43–62. ISBN: 9781450357678. DOI: 10.1145/3208903.3208909.

[138] CUI Gaoying, YI Yongxian, CAO Xiaojun, and GAO Ciwei. “Dynamic Aggregation Model Considering Load Fluctuation and Its Participation in Economic Dispatch of Power System”. In: *IOP Conference Series: Materials Science and Engineering* 199 (May 2017), p. 012075. DOI: 10.1088/1757-899x/199/1/012075.

[139] Liudong Zhang, Yubo Yuan, Bing Chen, and Dawei Su. “A robust interval economic dispatch model accommodating large-scale wind power generation with consideration of price-based demand response”. In: *2017 29th Chinese Control And Decision Conference (CCDC)*. 2017, pp. 6679–6684. DOI: 10.1109/CCDC.2017.7978379.

[140] Chenlu Zou, Xue Cui, Heng Wang, Bin Zhou, and Yang Liu. “Research on Power System Scheduling Improving Wind Power Accommodation Considering Thermal Energy Storage and Flexible Load”. In: *IOP Conference Series: Earth and Environmental Science* 108 (Jan. 2018), p. 052084. DOI: 10.1088/1755-1315/108/5/052084.

[141] A. Nikoobakht, J. Aghaei, R. Khatami, E. Mahboubi-Moghaddam, and M. Parvania. “Stochastic flexible transmission operation for coordinated integration of plug-in electric vehicles and renewable energy sources”. In: *Applied Energy* 238 (2019), pp. 225–238. DOI: 10.1016/j.apenergy.2018.12.089.

[142] M.-A. Nasr, S. Nikkhah, G.B. Gharehpetian, E. Nasr-Azadani, and S.H. Hosseinian. “A multi-objective voltage stability constrained energy management system for isolated microgrids”. In: *International Journal of Electrical Power and Energy Systems* 117 (2020). DOI: 10.1016/j.ijepes.2019.105646.

[143] Tianyou Li, Zhixuan Liu, Yanxin Chai, Yue Xiang, Junyong Liu, and Zeming Li. “Automatic Simulation of Active Distribution Network Based on Multi-Agent Technology”. In: *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*. 2019, pp. 3293–3298. DOI: 10.1109/ISGT-Asia.2019.8881103.

[144] T T Zheng, Y W Li, P W Yang, X L Chen, R Zheng, S H Miao, H Ji, and H R Yin. “Power system joint energy and reserve scheduling model considering variable speed pumped hydro storage for mitigating wind curtailment”. In: *IOP Conference Series: Earth and Environmental Science* 354 (Oct. 2019), p. 012081. DOI: 10.1088/1755-1315/354/1/012081.

[145] M.A. Mirzaei, A. Sadeghi-Yazdankhah, B. Mohammadi-Ivatloo, M. Marzband, M. Shafie-khah, and J.P.S. Catalão. “Integration of emerging resources in IGDT-based robust scheduling of combined power and natural gas systems considering flexible ramping products”. In: *Energy* 189 (2019). DOI: 10.1016/j.energy.2019.116195.

[146] Liu Yunjia, Meng Tenglong, Liu Jialiang, and Qu Zhanzhan. “Independent microgrid day-ahead optimization based on demand response”. In: *2017 29th Chinese Control And Decision Conference (CCDC)*. 2017, pp. 5809–5814. DOI: 10.1109/CCDC.2017.7978205.

[147] I. Ali and S.M. Suhail Hussain. “Communication Design for Energy Management Automation in Microgrid”. In: *IEEE Transactions on Smart Grid* 9.3 (2018), pp. 2055–2064. DOI: 10.1109/TSG.2016.2606131.

[148] A. Golshani, W. Sun, Q. Zhou, Q.P. Zheng, J. Wang, and F. Qiu. “Coordination of Wind Farm and Pumped-Storage Hydro for a Self-Healing Power Grid”. In: *IEEE Transactions on Sustainable Energy* 9.4 (2018), pp. 1910–1920. DOI: 10.1109/TSTE.2018.2819133.

[149] F. Xiao and Q. Ai. “New modeling framework considering economy, uncertainty, and security for estimating the dynamic interchange capability of multi-microgrids”. In: *Electric Power Systems Research* 152 (2017), pp. 237–248. DOI: 10.1016/j.epsr.2017.07.001.

[150] J.M. Alemany, B. Arendarski, P. Lombardi, and P. Komarnicki. “Accentuating the renewable energy exploitation: Evaluation of flexibility options”. In: *International Journal of Electrical Power and Energy Systems* 102 (2018), pp. 131–151. DOI: 10.1016/j.ijepes.2018.04.023.

[151] M.K. Petersen, K. Edlund, L.H. Hansen, J. Bendtsen, and J. Stoustrup. “A taxonomy for modeling flexibility and a computationally efficient algorithm for dispatch in Smart Grids”. In: *Proceedings of the American Control Conference*. 2013, pp. 1150–1156.

[152] H. Hao, A. Somani, J. Lian, and T. E. Carroll. “Generalized aggregation and coordination of residential loads in a smart community”. In: *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. Nov. 2015, pp. 67–72. DOI: 10.1109/SmartGridComm.2015.7436278.

[153] L. Ni, W. Liu, F. Wen, Y. Xue, Z. Dong, Y. Zheng, and R. Zhang. “Optimal operation of electricity, natural gas and heat systems considering integrated demand responses and diversified storage devices”. In: *Journal of Modern Power Systems and Clean Energy* 6.3 (2018), pp. 423–437. DOI: 10.1007/s40565-017-0360-6.

[154] L. Barth, N. Ludwig, E. Mengelkamp, and P. Staudt. “A comprehensive modelling framework for demand side flexibility in smart grids”. In: *Computer Science - Research and Development* 33.1-2 (2018), pp. 13–23. DOI: 10.1007/s00450-017-0343-x.

[155] M. Giuntoli and D. Poli. “Optimized thermal and electrical scheduling of a large scale virtual power plant in the presence of energy storages”. In: *IEEE Transactions on Smart Grid* 4.2 (2013), pp. 942–955. DOI: 10.1109/TSG.2012.2227513.

[156] E. Valsomatzis, K. Hose, and T.B. Pedersen. “Balancing Energy Flexibilities Through Aggregation”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8817 (2014), pp. 17–37. DOI: 10.1007/978-3-319-13290-7_2.

[157] E. Valsomatzis, K. Hose, T.B. Pedersen, and L. Šikšnys. "Measuring and comparing energy flexibilities". In: *CEUR Workshop Proceedings*. Vol. 1330. 2015, pp. 78–85.

[158] Luis A. Duffaut Espinosa, Mads Almassalkhi, Paul Hines, and Jeff Frolik. "Aggregate modeling and coordination of diverse energy resources under packetized energy management". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 1394–1400. DOI: 10.1109/CDC.2017.8263849.

[159] M. Negrete-Pincetic, A. Nayyar, K. Poolla, F. Salah, and P. Varaiya. "Rate-Constrained Energy Services in Electricity". In: *IEEE Transactions on Smart Grid* 9.4 (2018), pp. 2894–2907. DOI: 10.1109/TSG.2016.2623275.

[160] R. Pedersen, J. Schwensen, B. Biegel, J. Stoustrup, and T. Green. "Aggregation and control of supermarket refrigeration systems in a smart grid". In: *IFAC Proceedings Volumes (IFAC-PapersOnline)*. Vol. 19. 2014, pp. 9942–9949.

[161] Z. Xu, D.S. Callaway, Z. Hu, and Y. Song. "Hierarchical Coordination of Heterogeneous Flexible Loads". In: *IEEE Transactions on Power Systems* 31.6 (2016), pp. 4206–4216. DOI: 10.1109/TPWRS.2016.2516992.

[162] S. Stinner, K. Huchtemann, and D. Müller. "Quantifying the operational flexibility of building energy systems with thermal energy storages". In: *Applied Energy* 181 (2016), pp. 140–154. DOI: 10.1016/j.apenergy.2016.08.055.

[163] A. Ulbig and G. Andersson. "Analyzing operational flexibility of electric power systems". In: *International Journal of Electrical Power and Energy Systems* 72 (2015), pp. 155–164. DOI: 10.1016/j.ijepes.2015.02.028.

[164] A. Ulbig, M.A. Bucher, and G. Andersson. "Operational Flexibility of Power Systems". In: *Renewable Energy Integration: Practical Management of Variability, Uncertainty, and Flexibility in Power Grids: Second Edition*. Elsevier, 2017, pp. 201–216. DOI: 10.1016/B978-0-12-809592-8.00015-9.

[165] H. Ji, C. Wang, P. Li, G. Song, H. Yu, and J. Wu. "Quantified analysis method for operational flexibility of active distribution networks with high penetration of distributed generators". In: *Applied Energy* 239 (2019), pp. 706–714. DOI: 10.1016/j.apenergy.2019.02.008.

[166] A. Majzoobi and A. Khodaei. "Application of Microgrids in Supporting Distribution Grid Flexibility". In: *IEEE Transactions on Power Systems* 32.5 (2017), pp. 3660–3669. DOI: 10.1109/TPWRS.2016.2635024.

[167] Jacob Hansen, Jesper Knudsen, and Anuradha M. Annaswamy. "Demand response in smart grids: Participants, challenges, and a taxonomy". In: *53rd IEEE Conference on Decision and Control*. 2014, pp. 4045–4052. DOI: 10.1109/CDC.2014.7040018.

[168] R. De Coninck and L. Helsen. "Quantification of flexibility in buildings by cost curves - Methodology and application". In: *Applied Energy* 162 (2016), pp. 653–665. DOI: 10.1016/j.apenergy.2015.10.114.

[169] P. Schott, J. Sedlmeir, N. Strobel, T. Weber, G. Fridgen, and E. Abele. "A generic data model for describing flexibility in power markets". In: *Energies* 12.10 (2019). DOI: 10.3390/en12101893.

[170] Shantanu Chakraborty, Pablo Hernandez-Leal, and Michael Kaisers. "Coordinating Distributed and Flexible Resources: A Case-study of Residential Cooperatives". In: *2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. 2018, pp. 1–6. DOI: 10.1109/ISGTEurope.2018.8571562.

[171] M. Ding and K. Luo. "A multi-agent energy coordination control strategy in microgrid island mode". In: *Lecture Notes in Electrical Engineering* 238 LNEE (2014), pp. 529–536. DOI: 10.1007/978-1-4614-4981-2_58.

[172] Felix N. Claessen, Michael Kaisers, and Han La Poutré. “Efficient balancing by effort-based activation of demand response services”. In: *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. 2016, pp. 1–6. DOI: 10.1109/ISGTEurope.2016.7856232.

[173] J.P. Iria, F.J. Soares, and M.A. Matos. “Trading small prosumers flexibility in the energy and tertiary reserve markets”. In: *IEEE Transactions on Smart Grid* 10.3 (2019), pp. 2371–2382. DOI: 10.1109/TSG.2018.2797001.

[174] E. Amicarelli, T. Q. Tran, and S. Bacha. “Optimization algorithm for microgrids day-ahead scheduling and aggregator proposal”. In: *2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I CPS Europe)*. 2017, pp. 1–6. DOI: 10.1109/EEEIC.2017.7977487.

[175] C.W. Yu, S.H. Zhang, L. Wang, and T.S. Chung. “Integrating interruptible IPP contracts in the probabilistic production simulation of power market systems”. In: *International Journal of Electrical Power and Energy Systems* 29.7 (2007), pp. 557–564. DOI: 10.1016/j.ijepes.2007.01.005.

[176] D.J. Shiltz, S. Baros, M. Cvetkovic, and A.M. Annaswamy. “Integration of Automatic Generation Control and Demand Response via a Dynamic Regulation Market Mechanism”. In: *IEEE Transactions on Control Systems Technology* 27.2 (2019), pp. 631–646. DOI: 10.1109/TCST.2017.2776864.

[177] Dima El Nabouch, Natalie Matta, Rana Rahim-Amoud, and Leila Merghem-Boulahia. “An Agent-Based Approach for Efficient Energy Management in the Context of Smart Houses”. In: *Highlights on Practical Applications of Agents and Multi-Agent Systems*. Ed. by Juan M. Corchado, Javier Bajo, Jaroslaw Kozlak, Pawel Pawlewski, Jose M. Molina, Vicente Julian, Ricardo Azambuja Silveira, Rainer Unland, and Sylvain Giroux. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 375–386. ISBN: 978-3-642-38061-7.

[178] S. Chen, Q. Chen, and Y. Xu. “Strategic Bidding and Compensation Mechanism for a Load Aggregator With Direct Thermostat Control Capabilities”. In: *IEEE Transactions on Smart Grid* 9.3 (2018), pp. 2327–2336. DOI: 10.1109/TSG.2016.2611611.

[179] Michael D. Knudsen and Sergi Rotger-Griful. “Combined price and event-based demand response using two-stage model predictive control”. In: *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 2015, pp. 344–349. DOI: 10.1109/SmartGridComm.2015.7436324.

[180] Samuel Pfaffen, Karl Werlen, and Stephan Koch. “Evaluation of business models for the economic exploitation of flexible thermal loads”. In: *IECON 2013 - 39th Annual Conference of the IEEE Industrial Electronics Society*. 2013, pp. 4745–4750. DOI: 10.1109/IECON.2013.6699902.

[181] F. Tahersima, J. Stoustrup, P. Andersen, and P.P. Madsen. “Economic energy distribution and consumption in a microgrid Part 2: Macrocell level controller”. In: *IFAC Proceedings Volumes* 47.3 (2014). 19th IFAC World Congress, pp. 4553–4559. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20140824-6-ZA-1003.00078.

[182] Peiyun Liu, Tao Ding, Yuankang He, and Tianen Chen. “Integrated Demand Response in Multi-Energy Market Based on Flexible Loads Classification”. In: *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*. 2019, pp. 4346–4350. DOI: 10.1109/ISGT-Asia.2019.8881718.

[183] Y. Jia, Z. Mi, Y. Yu, Z. Song, and C. Sun. “A bilevel model for optimal bidding and offering of flexible load aggregator in day-ahead energy and reserve markets”. In: *IEEE Access* 6 (2018), pp. 67799–67808. DOI: 10.1109/ACCESS.2018.2879058.

[184] K.O. Aduda, T. Labeodan, W. Zeiler, and G. Boxem. “Demand side flexibility coordination in office buildings: A framework and case study application”. In: *Sustainable Cities and Society* 29 (2017), pp. 139–158. DOI: 10.1016/j.scs.2016.12.008.

[185] M. Khoshjahan, P. Dehghanian, M. Moeini-Aghtaie, and M. Fotuhi-Firuzabad. “Harnessing Ramp Capability of Spinning Reserve Services for Enhanced Power Grid Flexibility”. In: *IEEE Transactions on Industry Applications* 55.6 (2019), pp. 7103–7112. DOI: 10.1109/TIA.2019.2921946.

[186] C. Ordoudis, P. Pinson, and J.M. Morales. “An Integrated Market for Electricity and Natural Gas Systems with Stochastic Power Producers”. In: *European Journal of Operational Research* 272.2 (2019), pp. 642–654. DOI: 10.1016/j.ejor.2018.06.036.

[187] Kexue Chen, Dong Han, and Weiqing Sun. “Modeling for operational strategy of flexible ramping capacity in deregulated power system considering EV aggregators participation”. In: *2018 International Conference on Power System Technology (POWERCON)*. 2018, pp. 4175–4180. DOI: 10.1109/POWERCON.2018.8602092.

[188] M. Shaaban, W.-S. Tan, and M.P. Abdullah. “A multi-timescale hybrid stochastic/deterministic generation scheduling framework with flexiramp and cycliramp costs”. In: *International Journal of Electrical Power and Energy Systems* 99 (2018), pp. 585–593. DOI: 10.1016/j.ijepes.2018.02.004.

[189] T. Kulms, A.-K. Meinerzhagen, S. Koopmann, and A. Schnettler. “A simulation framework for assessing the market and grid driven value of flexibility options in distribution grids”. In: *Journal of Energy Storage* 17 (2018), pp. 203–212. DOI: 10.1016/j.est.2018.03.003.

[190] Dezhi Li, Junyong Wu, Kun Shi, Tengfei Ma, Yin Zhang, and Ruoyu Zhang. “Bi-level optimal dispatch model for micro energy grid based on load aggregator business”. In: *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*. 2017, pp. 1–6. DOI: 10.1109/EI2.2017.8245321.

[191] E. Heydarian-Forushani, M. E. H. Golshan, M. Shafie-khah, and J. P. S. Catalão. “Optimal coordination of Battery Energy Storages and Demand Response Programs with application to wind integration”. In: *2015 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*. 2015, pp. 1–6. DOI: 10.1109/SEGE.2015.7324594.

[192] J. Aghaei, A. Nikoobakht, M. Mardaneh, M. Shafie-khah, and J.P.S. Catalão. “Transmission switching, demand response and energy storage systems in an innovative integrated scheme for managing the uncertainty of wind power generation”. In: *International Journal of Electrical Power and Energy Systems* 98 (2018), pp. 72–84. DOI: 10.1016/j.ijepes.2017.11.044.

[193] W. El-Baz, P. Tzscheutschler, and U. Wagner. “Integration of energy markets in microgrids: A double-sided auction with device-oriented bidding strategies”. In: *Applied Energy* 241 (2019), pp. 625–639. DOI: 10.1016/j.apenergy.2019.02.049.

[194] F. Babonneau, M. Caramanis, and A. Haurie. “A linear programming model for power distribution with demand response and variable renewable energy”. In: *Applied Energy* 181 (2016), pp. 83–95. DOI: 10.1016/j.apenergy.2016.08.028.

[195] A. Esmat, J. Usaola, and M.Á. Moreno. “A decentralized local flexibility market considering the uncertainty of demand”. In: *Energies* 11.8 (2018). DOI: 10.3390/en11082078.

[196] G. He, Q. Chen, C. Kang, and Q. Xia. “Optimal Offering Strategy for Concentrating Solar Power Plants in Joint Energy, Reserve and Regulation Markets”. In: *IEEE Transactions on Sustainable Energy* 7.3 (2016), pp. 1245–1254. DOI: 10.1109/TSTE.2016.2533637.

[197] J.I. Otashu and M. Baldea. “Grid-level “battery” operation of chemical processes and demand-side participation in short-term electricity markets”. In: *Applied Energy* 220 (2018), pp. 562–575. DOI: 10.1016/j.apenergy.2018.03.034.

[198] P. Ströhle and C.M. Flath. “Local matching of flexible load in smart grids”. In: *European Journal of Operational Research* 253.3 (2016), pp. 811–824. DOI: 10.1016/j.ejor.2016.03.004.

[199] S. Mathieu, Q. Louveaux, D. Ernst, and B. Cornélusse. “DSIMA: A testbed for the quantitative analysis of interaction models within distribution networks”. In: *Sustainable Energy, Grids and Networks* 5 (2016), pp. 78–93. DOI: 10.1016/j.segan.2015.11.004.

[200] S. Mathieu, D. Ernst, and B. Cornéiusse. “Agent-based analysis of dynamic access ranges to the distribution network”. In: *2016 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. 2016, pp. 1–6. DOI: 10.1109/ISGTEurope.2016.7856191.

[201] Sebastien Mathieu, Frederic Olivier, Damien Ernst, and Bertrand CornElusse. “Effect of voltage constraints on the exchange of flexibility services in distribution networks”. In: *2018 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. 2018, pp. 1–5. DOI: 10.1109/ISGT.2018.8403328.

[202] M. K. Petersen, L. H. Hansen, J. Bendtsen, K. Edlund, and J. Stoustrup. “Market integration of Virtual Power Plants”. In: *52nd IEEE Conference on Decision and Control*. 2013, pp. 2319–2325. DOI: 10.1109/CDC.2013.6760227.

[203] Meimei Duan, Xuesong Shao, Gaoying Cui, Aixia Zheng, Gege Tong, Gang Wu, and Ciwei Gao. “Rolling dispatch of demand response resources considering renewable energy accommodation”. In: *2017 4th International Conference on Systems and Informatics (ICSAI)*. 2017, pp. 358–362. DOI: 10.1109/ICSAI.2017.8248318.

[204] Jiang Wu, Feng Gao, and Zhixi Kang. “A multi-agent system for households response to dynamic pricing”. In: *IEEE PES Innovative Smart Grid Technologies*. 2012, pp. 1–5. DOI: 10.1109/ISGT-Asia.2012.6303204.

[205] K. Paridari, D. Azuatalam, A.C. Chapman, G. Verbic, and L. Nordstrom. “A plug-and-play home energy management algorithm using optimization and machine learning techniques”. In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2018*. 2018. DOI: 10.1109/SmartGridComm.2018.8587418.

[206] Jianming Lian, Di Wu, Karanjit Kalsi, and Hua Chen. “Theoretical framework for integrating distributed energy resources into distribution systems”. In: *2017 IEEE Power Energy Society General Meeting*. 2017, pp. 1–5. DOI: 10.1109/PESGM.2017.8274331.

[207] Antonio De Paola, Filiberto Fele, David Angeli, and Goran Strbac. “Distributed Coordination of Price-Responsive Electric Loads: A Receding Horizon Approach”. In: *2018 IEEE Conference on Decision and Control (CDC)*. 2018, pp. 6033–6040. DOI: 10.1109/CDC.2018.8619024.

[208] A. Asadinejad and K. Tomsovic. “Optimal use of incentive and price based demand response to reduce costs and price volatility”. In: *Electric Power Systems Research* 144 (2017), pp. 215–223. DOI: 10.1016/j.epsr.2016.12.012.

[209] A. Dadkhah and B. Vahidi. “On the network economic, technical and reliability characteristics improvement through demand-response implementation considering consumers’ behaviour”. In: *IET Generation, Transmission and Distribution* 12.2 (2018), pp. 431–440. DOI: 10.1049/iet-gtd.2017.0554.

[210] Y. Wang, Y. Huang, Y. Wang, M. Zeng, H. Yu, F. Li, and F. Zhang. “Optimal scheduling of the RIES considering time-based demand response programs with energy price”. In: *Energy* 164 (2018), pp. 773–793. DOI: 10.1016/j.energy.2018.09.014.

[211] Jacopo Parvizi, John B. Jørgensen, and Henrik Madsen. "Robust Model Predictive Control with Scenarios for Aggregators in Grids with High Penetration of Renewable Energy Sources." In: *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. 2018, pp. 1–7. DOI: 10.1109/SmartGridComm.2018.8587603.

[212] G. Mantovani, G.T. Costanzo, M. Marinelli, and L. Ferrarini. "Experimental validation of energy resources integration in microgrids via distributed predictive control". In: *IEEE Transactions on Energy Conversion* 29.4 (2014), pp. 1018–1025. DOI: 10.1109/TEC.2014.2362887.

[213] Araz Ashouri, Paul Stadler, and François Maréchal. "Day-ahead promised load as alternative to real-time pricing". In: *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 2015, pp. 551–556. DOI: 10.1109/SmartGridComm.2015.7436358.

[214] Tri Kurniawan Wijaya, Dipyaman Banerjee, Tanuja Ganu, Dipanjan Chakraborty, Sourav Battacharya, Thanasis Papaioannou, Deva P. Seetharam, and Karl Aberer. "DRSim: A cyber physical simulator for Demand Response systems". In: *2013 IEEE International Conference on Smart Grid Communications (SmartGridComm)*. 2013, pp. 217–222. DOI: 10.1109/SmartGridComm.2013.6687960.

[215] M. Sun, P. Djapic, M. Aunedi, D. Pudjianto, and G. Strbac. "Benefits of smart control of hybrid heat pumps: An analysis of field trial data". In: *Applied Energy* 247 (2019), pp. 525–536. DOI: 10.1016/j.apenergy.2019.04.068.

[216] B. Felten and C. Weber. "The value(s) of flexible heat pumps – Assessment of technical and economic conditions". In: *Applied Energy* 228 (2018), pp. 1292–1319. DOI: 10.1016/j.apenergy.2018.06.031.

[217] Sandeep Chawda, Parul Mathuria, and Rohit Bhakar. "Dynamic Retail Pricing for Load Serving Entity Under Significant Renewable Energy Penetration". In: *2018 8th IEEE India International Conference on Power Electronics (IICPE)*. 2018, pp. 1–5. DOI: 10.1109/IICPE.2018.8709489.

[218] W. Lv, J. Wu, Z. Luo, M. Ding, X. Jiang, H. Li, and Q. Wang. "Load Aggregator-Based Integrated Demand Response for Residential Smart Energy Hubs". In: *Mathematical Problems in Engineering* 2019 (2019). DOI: 10.1155/2019/6925980.

[219] S. Yoon and E. Hwang. "Load Guided Signal-Based Two-Stage Charging Coordination of Plug-In Electric Vehicles for Smart Buildings". In: *IEEE Access* 7 (2019), pp. 144548–144560. DOI: 10.1109/ACCESS.2019.2945483.

[220] L. Han, T. Morstyn, and M. McCulloch. "Incentivizing Prosumer Coalitions With Energy Management Using Cooperative Game Theory". In: *IEEE Transactions on Power Systems* 34.1 (2019), pp. 303–313. DOI: 10.1109/TPWRS.2018.2858540.

[221] J. Niu, Z. Tian, Y. Lu, and H. Zhao. "Flexible dispatch of a building energy system using building thermal storage and battery energy storage". In: *Applied Energy* 243 (2019), pp. 274–287. DOI: 10.1016/j.apenergy.2019.03.187.

[222] Rosa Morales González, Madeleine Gibescu, Sjef Cobben, Martijn Bongaerts, Marcel de Nes-Koedam, and Wouter Vermeiden. "Medical Freezers as Flexible Load for Demand Response in a Business Park Microgrid with Local Solar Power Generation". In: *Smart Cities, Green Technologies and Intelligent Transport Systems*. Ed. by Brian Donnellan, Cornel Klein, Markus Helfert, and Oleg Gusikhin. Cham: Springer International Publishing, 2019, pp. 23–43. ISBN: 978-3-030-26633-2.

[223] Ling Wei and M. Yang. "A long-term energy purchasing model of distribution company in power market with DG and multi-types contracts". In: *2008 China International Conference on Electricity Distribution*. Dec. 2008, pp. 1–6. DOI: 10.1109/CICED.2008.5211731.

[224] R. Chen, H. Sun, Q. Guo, H. Jin, W. Wu, and B. Zhang. "Profit-seeking energy-intensive enterprises participating in power system scheduling: Model and mechanism". In: *Applied Energy* 158 (2015), pp. 263–274. DOI: 10.1016/j.apenergy.2015.08.018.

[225] E. Heydarian-Forushani, M.E.H. Golshan, and P. Siano. "Evaluating the benefits of coordinated emerging flexible resources in electricity markets". In: *Applied Energy* 199 (2017), pp. 142–154. DOI: 10.1016/j.apenergy.2017.04.062.

[226] S. Mitra, L. Sun, and I.E. Grossmann. "Optimal scheduling of industrial combined heat and power plants under time-sensitive electricity prices". In: *Energy* 54 (2013), pp. 194–211. DOI: 10.1016/j.energy.2013.02.030.

[227] Martin E. Wainstein, Roger Dargaville, and Adam Bumpus. "Social virtual energy networks: Exploring innovative business models of prosumer aggregation with virtual power plants". In: *2017 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. 2017, pp. 1–5. DOI: 10.1109/ISGT.2017.8086022.

[228] J. Kong, Z. Shen, M. Niu, T. Zhang, B. Liu, S. Hu, and H. Sun. "Optimized Economic Dispatch of Active Distribution Network with Electric Vehicle Group". In: *IOP Conference Series: Materials Science and Engineering*. Vol. 366. 1. 2018. DOI: 10.1088/1757-899X/366/1/012077.

[229] D. Croce, F. Giuliano, I. Tinnirello, A. Galatioto, M. Bonomolo, M. Beccali, and G. Zizzo. "Overgrid: A Fully Distributed Demand Response Architecture Based on Overlay Networks". In: *IEEE Transactions on Automation Science and Engineering* 14.2 (2017), pp. 471–481. DOI: 10.1109/TASE.2016.2621890.

[230] L. Fabietti, T.T. Gorecki, E. Namor, F. Sossan, M. Paolone, and C.N. Jones. "Enhancing the dispatchability of distribution networks through utility-scale batteries and flexible demand". In: *Energy and Buildings* 172 (2018), pp. 125–138. DOI: 10.1016/j.enbuild.2018.04.056.

[231] Y. Li, W. Yang, P. He, C. Chen, and X. Wang. "Design and management of a distributed hybrid energy system through smart contract and blockchain". In: *Applied Energy* 248 (2019), pp. 390–405. DOI: 10.1016/j.apenergy.2019.04.132.

[232] M.A. Bucher, S. Chatzivasileiadis, and G. Andersson. "Managing Flexibility in Multi-Area Power Systems". In: *IEEE Transactions on Power Systems* 31.2 (2016), pp. 1218–1226. DOI: 10.1109/TPWRS.2015.2413387.

[233] M. Juelsgaard, P. Andersen, and R. Wisniewski. "Distribution loss reduction by household consumption coordination in smart grids". In: *IEEE Transactions on Smart Grid* 5.4 (2014), pp. 2133–2144. DOI: 10.1109/TSG.2014.2312428.

[234] R. Verschae, H. Kawashima, T. Kato, and T. Matsuyama. "A distributed coordination framework for on-line scheduling and power demand balancing of households communities". In: *2014 European Control Conference (ECC)*. June 2014, pp. 1655–1662. DOI: 10.1109/ECC.2014.6862394.

[235] J. Hu, G. Yang, C. Ziras, and K. Kok. "Aggregator Operation in the Balancing Market Through Network-Constrained Transactive Energy". In: *IEEE Transactions on Power Systems* 34.5 (2019), pp. 4071–4080. DOI: 10.1109/TPWRS.2018.2874255.

[236] Daniel Esteban Morales Bondy, Giuseppe Tommaso Costanzo, Kai Heussen, and Henrik W. Bindner. "Performance assessment of aggregation control services for demand response". In: *IEEE PES Innovative Smart Grid Technologies, Europe*. 2014, pp. 1–6. DOI: 10.1109/ISGTEurope.2014.7028779.

[237] Giuseppe Tommaso Costanzo, Oliver Gehrke, Daniel Esteban Morales Bondy, Fabrizio Sossan, Henrik W. Bindner, Jacopo Parvizi, and Henrik Madsen. "A Coordination Scheme for Distributed Model Predictive Control: Integration of Flexible DERs". English. In: *Proceedings of IEEE Innovative Smart Grid Technologies Europe 2013*. 2013 4th IEEE PES Innovative Smart Grid Technologies Europe, 2013 ISGT Europe ; Conference date: 06-10-2013 Through 09-10-2013. United States: IEEE, 2013. DOI: 10.1109/ISGTEurope.2013.6695474.

[238] M. Kahl, C. Freye, and T. Leibfried. "A Cooperative Multi-Area Optimization with Renewable Generation and Storage Devices". In: *IEEE Transactions on Power Systems* 30.5 (2015), pp. 2386–2395. DOI: 10.1109/TPWRS.2014.2363762.

[239] Bishnu P. Bhattarai, Iker Diaz De Zerio Mendaza, Birgitte Bak-Jensen, Jayakrishnan R. Pillai, Nava R. Karki, Jake P. Gentle, and Kurt S. Myers. "Active control of thermostatic loads for economic and technical support to distribution grids". In: *2016 IEEE Power and Energy Society General Meeting (PESGM)*. 2016, pp. 1–5. DOI: 10.1109/PESGM.2016.7741735.

[240] I.D. De Cerio Mendaza, I.G. Szczesny, J.R. Pillai, and B. Bak-Jensen. "Demand Response Control in Low Voltage Grids for Technical and Commercial Aggregation Services". In: *IEEE Transactions on Smart Grid* 7.6 (2016), pp. 2771–2780. DOI: 10.1109/TSG.2015.2465837.

[241] B. Huang, L. Liu, H. Zhang, Y. Li, and Q. Sun. "Distributed Optimal Economic Dispatch for Microgrids Considering Communication Delays". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.8 (2019), pp. 1634–1642. DOI: 10.1109/TSMC.2019.2900722.

[242] G. Lou, W. Gu, W. Sheng, X. Song, and F. Gao. "Distributed model predictive secondary voltage control of islanded microgrids with feedback linearization". In: *IEEE Access* 6 (2018), pp. 50169–50178. DOI: 10.1109/ACCESS.2018.2869280.

[243] Paul Stadler, Araz Ashouri, and François Maréchal. "Distributed model predictive control of energy systems in microgrids". In: *2016 Annual IEEE Systems Conference (SysCon)*. 2016, pp. 1–6. DOI: 10.1109/SYSCON.2016.7490607.

[244] Xu Liu, Hongtao Wang, Qinyong Zhou, and Bin Hu. "Multi-area scale electric power and energy balance with high wind power penetration based STACKELBERG decision theory". In: *IEEE PES Innovative Smart Grid Technologies*. 2012, pp. 1–6. DOI: 10.1109/ISGT-Asia.2012.6303213.

[245] Y. Du, J. Wu, S. Li, C. Long, and I.Ch. Paschalidis. "Distributed MPC for coordinated energy efficiency utilization in microgrid systems". In: *IEEE Transactions on Smart Grid* 10.2 (2019), pp. 1781–1790. DOI: 10.1109/TSG.2017.2777975.

[246] Yaping Li, Juan Zuo, Wenbo Mao, and Xiaorui Guo. "The flexibility of Thermostatically Controlled Loads for regulation". In: *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*. 2016, pp. 305–309. DOI: 10.1109/APPEEC.2016.7779517.

[247] H. Hui, Y. Ding, Y. Song, and S. Rahman. "Modeling and control of flexible loads for frequency regulation services considering compensation of communication latency and detection error". In: *Applied Energy* 250 (2019), pp. 161–174. DOI: 10.1016/j.apenergy.2019.04.191.

[248] Konstantin Turitsyn, Petr Šulc, Scott Backhaus, and Michael Chertkov. "Distributed control of reactive power flow in a radial distribution circuit with high photovoltaic penetration". In: *IEEE PES General Meeting*. 2010, pp. 1–6. DOI: 10.1109/PES.2010.5589663.

[249] G. Ensermu, A. Bhattacharya, and N. Panigrahy. "Real-Time Simulation of Smart DC Microgrid with Decentralized Control System Under Source Disturbances". In: *Arabian Journal for Science and Engineering* 44.8 (2019), pp. 7173–7185. DOI: 10.1007/s13369-019-03842-6.

[250] N. Voropai, C. Rehtanz, S. Kippelt, N. Tomin, U. Haeger, D. Efimov, V. Kurbatsky, and I. Kolosok. “The Development of a Joint Modelling Framework for Operational Flexibility in Power Systems”. In: *2019 16th Conference on Electrical Machines, Drives and Power Systems (ELMA)*. June 2019, pp. 1–6. DOI: 10.1109/ELMA.2019.8771685.

[251] Hermen Toersche. “Effective and efficient coordination of flexibility in smart grids”. Undefined. PhD thesis. Netherlands: University of Twente, Oct. 2016. ISBN: 978-90-365-4197-8. DOI: 10.3990/1.9789036541978.

[252] A. Mnatsakanyan and S. W. Kennedy. “A Novel Demand Response Model with an Application for a Virtual Power Plant”. In: *IEEE Transactions on Smart Grid* 6.1 (Jan. 2015), pp. 230–237. ISSN: 1949-3061. DOI: 10.1109/TSG.2014.2339213.

[253] Kilian Geschermann, Katharina Volk, Christian Lakenbrink, and Martin Konermann. “PV integration with flexible generation and consumption units – evaluation of a quota-based grid traffic light approach in a field test”. In: *Digital Proceedings of the 8th International Workshop on Integration of Solar Power into Power Systems*. Oct. 2018. ISBN: 978-3-9820080-0-4.

[254] Kaibin Bao, Kanan Ibrahimov, Martin Wagner, and Hartmut Schmeck. “Enhancing neural non-intrusive load monitoring with generative adversarial networks”. In: *Energy Informatics* 1.1 (Oct. 2018), p. 18. ISSN: 2520-8942. DOI: 10.1186/s42162-018-0038-y.

[255] Z. A. Khan, D. Jayaweera, and H. Gunduz. “Smart meter data taxonomy for demand side management in smart grids”. In: *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*. Oct. 2016, pp. 1–8. DOI: 10.1109/PMAPS.2016.7764143.

[256] Anke Weidlich and Ahsan Zaidi. “Operational Flexibility of Small-Scale Electricity-Coupled Heat Generating Units”. In: *Technology and Economics of Smart Grids and Sustainable Energy* 4.1 (2019), p. 8. ISSN: 2199-4706. DOI: 10.1007/s40866-019-0064-2.

[257] Erik Heilmann, Nikolai Klempp, and Heike Wetzel. *Market design of regional flexibility markets: A classification metric for flexibility products and its application to German prototypical flexibility markets*. eng. MAGKS Joint Discussion Paper Series in Economics 02-2020. Marburg, 2020.

[258] S. Koeppl, C. Lang, A. Bogensperger, T. Estermann, and A. Zeiselmair. “Altdorfer Flexmarkt - Decentral flexibility for distribution networks”. In: *International ETG-Congress 2019; ETG Symposium*. May 2019, pp. 1–6.

[259] Andreas Zeiselmair, Simon Köppl, Thomas Estermann, Nico Lehmann, Emil Kraft, and Nikolai Klempp. “Netzdienlicher Handel als Element des zellulären Energiesystems am Beispiel des Altdorfer Flexmarkts (ALF)”. In: *11. Internationale Energiewirtschaftstagung an der TU Wien, Wien, Österreich, 13.-15. Februar 2019* (2019).

[260] M. Zade, Y. Incedag, W. El-Baz, P. Tzscheutschler, and U. Wagner. “Prosumer Integration in Flexibility Markets: A Bid Development and Pricing Model”. In: *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*. Oct. 2018, pp. 1–9. DOI: 10.1109/EI2.2018.8582022.

[261] Sebastian Kochanneck. “Systemdienstleistungserbringung durch intelligente Gebäude”. German. PhD thesis. Karlsruher Institut für Technologie (KIT), 2019. 308 pp. ISBN: 978-3-7315-0897-7. DOI: 10.5445/KSP/1000090879.

[262] Slawomir Koziel and Leifur Leifsson. *Surrogate-based modeling and optimization*. New York: Springer, 2013.

[263] Raul Yondo, Kamil Bobrowski, Esther Andrés, and Eusebio Valero. “A Review of Surrogate Modeling Techniques for Aerodynamic Analysis and Optimization: Current Limitations and Future Challenges in Industry”. In: *Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences*. Cham: Springer International Publishing, 2019, pp. 19–33. ISBN: 978-3-319-89988-6. DOI: 10.1007/978-3-319-89988-6_2.

[264] Yuichi Kuya, Kenji Takeda, Xin Zhang, and Alexander I. J. Forrester. “Multifidelity Surrogate Modeling of Experimental and Computational Aerodynamic Data Sets”. In: *AIAA Journal* 49.2 (2011), pp. 289–298. DOI: 10.2514/1.J050384.

[265] Jan Müller, Matthias März, Ingo Mauser, and Hartmut Schmeck. “Optimization of Operation and Control Strategies for Battery Energy Storage Systems by Evolutionary Algorithms”. In: *Applications of Evolutionary Computation*. Ed. by Giovanni Squillero and Paolo Burelli. Cham: Springer International Publishing, 2016, pp. 507–522. ISBN: 978-3-319-31204-0.

[266] Alexander W. Churchill, Siddharth Sigtia, and Chrisantha Fernando. “Learning to Generate Genotypes with Neural Networks”. In: *CoRR* abs/1604.04153 (2016).

[267] Mischa Ahrens and Hartmut Schmeck. “Generation of Time-of-Use Tariffs for Demand Side Management Using Artificial Neural Networks”. In: *Proceedings of the Ninth International Conference on Future Energy Systems*. e-Energy ’18. Karlsruhe, Germany: Association for Computing Machinery, 2018, pp. 396–398. ISBN: 9781450357678. DOI: 10.1145/3208903.3212037.

[268] Tesla, Inc. *Powerwall 2 AC Datasheet*. https://www.tesla.com/sites/default/files/pdfs/powerwall/Powerwall%202_AC_Datasheet_en_northamerica.pdf. Accessed on: 2020/11/25. 2019.

[269] Jan Philipp Schmidt, André Weber, and Ellen Ivers-Tiffée. “A novel and fast method of characterizing the self-discharge behavior of lithium-ion cells using a pulse-measurement technique”. In: *Journal of Power Sources* 274 (2015), pp. 1231–1238. ISSN: 0378-7753. DOI: https://doi.org/10.1016/j.jpowsour.2014.10.163.

[270] European Union. *Commission Delegated Regulation (EU) No 812/2013 of 18 February 2013 supplementing Directive 2010/30/EU of the European Parliament and of the Council with regard to the energy labelling of water heaters, hot water storage tanks and packages of water heater and solar device Text with EEA relevance*. 2013.

[271] Eoghan McKenna and Murray Thomson. “High-resolution stochastic integrated thermal–electrical domestic demand model”. In: *Applied Energy* 165.Supplement C (2016), pp. 445–461. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2015.12.089.

[272] SenerTec Kraft-Wärme-Energiesysteme GmbH. *Technische Daten - Dachs*. Art. Nr.: 06.05/4798.092.007. 2005.

[273] DIN Deutsches Institut für Normung e.V. *Heating boilers – Energy assessment of hot water storage systems; German version EN 15332:2007*. Beuth Verlag GmbH, 2008.

[274] Mercedes-Benz Cars UK Limited. *» smart EQ fortwo and forfour. The price list*. https://tools.mercedes-benz.co.uk/current/smart/pricelists/smart-electric-drive.pdf. Accessed on: 2020/11/25. 2020.

[275] BMW AG. *Specifications. The new BMWi3*. https://www.press.bmwgroup.com/global/article/attachment/T0280411EN/406749. Accessed on: 2020/11/25. 2017.

[276] BMW AG. *Technical specifications. BMW i3(120 Ah)*. https://www.press.bmwgroup.com/global/article/attachment/T0284828EN/415571. Accessed on: 2020/11/25. 2018.

[277] Volkswagen Group UK Ltd. *The new SEAT Mii electric. Model, pricing and specification*. https://web1.seat.co.uk/sites/default/files/seat_ra_file/mii-electric-pricelist-brochure-may-2020.pdf. Accessed on: 2020/11/25. 2020.

[278] Renault s.a.s. *New Renault ZOE*. https://cdn.group.renault.com/ren/ie/transversal-assets/brochures/zoe-brochure-may.pdf. Accessed on: 2020/11/25. 2020.

[279] Tesla, Inc. *Model S Premium Electric Sedan*. https://www.tesla.com/sites/default/files/tesla-model-s.pdf. Accessed on: 2020/11/25.

[280] BAE Batterien GmbH. *Technische Spezifikation für BAE SunDepot*. https://www.bae-berlin.de/images/pdf/zubehoer/BAE_SUNDEPOT_dt_2013.09.pdf. Accessed on: 2020/11/25.

[281] Constantine Spanos, Damon E. Turney, and Vasilis Fthenakis. "Life-cycle analysis of flow-assisted nickel zinc-, manganese dioxide-, and valve-regulated lead-acid batteries designed for demand-charge reduction". In: *Renewable and Sustainable Energy Reviews* 43 (2015), pp. 478–494. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2014.10.072.

[282] BAE Batterien GmbH. *Technische Spezifikation für BAE SECURA PVV BLOCK solar*.

[283] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. "Neural Architecture Search: A Survey". In: *Journal of Machine Learning Research* 20.55 (2019), pp. 1–21.

[284] Emin Orhan and Xaq Pitkow. "Skip Connections Eliminate Singularities". In: *International Conference on Learning Representations*. 2018.

[285] Yasutaka Furusho and Kazushi Ikeda. "Theoretical analysis of skip connections and batch normalization from generalization and optimization perspectives". In: *APSIPA Transactions on Signal and Information Processing* 9 (2020), e9. DOI: 10.1017/ATSIP.2020.7.

[286] Jörg Bremer and Michael Sonnenschein. "Constraint-handling for Optimization with Support Vector Surrogate Models - A Novel Decoder Approach". In: *ICAART 2013 - Proceedings of the 5th International Conference on Agents and Artificial Intelligence, Volume 2, Barcelona, Spain, 15-18 February, 2013*. Ed. by Joaquim Filipe and Ana L. N. Fred. Setúbal, Portugal: SciTePress, 2013, pp. 91–100.

[287] Jörg Bremer and Sebastian Lehnhoff. "Controlled Self-organization for Steering Local Multi-objective Optimization in Virtual Power Plants". In: *Highlights in Practical Applications of Agents, Multi-Agent Systems, and Trust-worthiness. The PAAMS Collection*. Ed. by Fernando De La Prieta, Philippe Mathieu, Jaime Andrés Rincón Arango, Alia El Bolock, Elena Del Val, Jaume Jordán Prunera, João Carneiro, Rubén Fuentes, Fernando Lopes, and Vicente Julian. Cham: Springer International Publishing, 2020, pp. 314–325. ISBN: 978-3-030-51999-5.

[288] David Wölfle, Arun Vishwanath, and Hartmut Schmeck. "A Guide for the Design of Benchmark Environments for Building Energy Optimization". In: *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. BuildSys '20. Virtual Event, Japan: Association for Computing Machinery, 2020, pp. 220–229. ISBN: 9781450380614. DOI: 10.1145/3408308.3427614.

[289] José R. Vázquez-Canteli, Jérôme Kämpf, Gregor Henze, and Zoltan Nagy. "CityLearn v1.0: An OpenAI Gym Environment for Demand Response with Deep Reinforcement Learning". In: *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. BuildSys '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 356–357. ISBN: 9781450370059. DOI: 10.1145/3360322.3360998.

[290] Vahid Beiranvand, Warren Hare, and Yves Lucet. "Best practices for comparing optimization algorithms". In: *Optimization and Engineering* 18.4 (Dec. 2017), pp. 815–848. ISSN: 1573-2924. DOI: 10.1007/s11081-017-9366-1.

[291] Marie Coffin and Matthew J. Saltzman. "Statistical Analysis of Computational Tests of Algorithms and Heuristics". In: *INFORMS Journal on Computing* 12.1 (2000), pp. 24–44. DOI: 10.1287/ijoc.12.1.24.11899.

[292] Frank Hutter, Holger H. Hoos, Kevin Leyton-Brown, and Thomas Stützle. "ParamILS: An Automatic Algorithm Configuration Framework". In: *Journal of Artificial Intelligence Research* 36 (2009), pp. 267–306. DOI: https://doi.org/10.1613/jair.2861.

[293] D. H. Wolpert and W. G. Macready. "No free lunch theorems for optimization". In: *IEEE Transactions on Evolutionary Computation* 1.1 (1997), pp. 67–82. DOI: 10.1109/4235.585893.

[294] John Silberholz, Bruce Golden, Swati Gupta, and Xingyin Wang. "Computational Comparison of Metaheuristics". In: *Handbook of Metaheuristics*. Cham: Springer International Publishing, 2019, pp. 581–604. ISBN: 978-3-319-91086-4. DOI: 10.1007/978-3-319-91086-4_18.

[295] M. Montaz Ali, Charoenchai Khompatraporn, and Zelda B. Zabinsky. "A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems". In: *Journal of Global Optimization* 31.4 (Apr. 2005), pp. 635–672. ISSN: 1573-2916. DOI: 10.1007/s10898-004-9972-2.

[296] Pierre Hansen, Nenad Mladenović, Jack Brimberg, and José A. Moreno Pérez. "Variable Neighborhood Search". In: *Handbook of Metaheuristics*. Cham: Springer International Publishing, 2019, pp. 57–97. ISBN: 978-3-319-91086-4. DOI: 10.1007/978-3-319-91086-4_3.

[297] José Antonio Parejo, Antonio Ruiz-Cortés, Sebastián Lozano, and Pablo Fernandez. "Metaheuristic optimization frameworks: a survey and benchmarking". In: *Soft Computing* 16.3 (Mar. 2012), pp. 527–561. ISSN: 1433-7479. DOI: 10.1007/s00500-011-0754-8.

[298] Matthew Andres Moreno, Wolfgang Banzhaf, and Charles Ofria. "Learning an Evolvable Genotype-Phenotype Mapping". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '18. Kyoto, Japan: Association for Computing Machinery, 2018, pp. 983–990. ISBN: 9781450356183. DOI: 10.1145/3205455.3205597.

[299] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. "Safe Reinforcement Learning via Shielding". In: *CoRR* abs/1708.08611 (2017).

[300] J. J. Durillo, A. J. Nebro, C. A. C. Coello, J. Garcia-Nieto, F. Luna, and E. Alba. "A Study of Multiobjective Metaheuristics When Solving Parameter Scalable Problems". In: *IEEE Transactions on Evolutionary Computation* 14.4 (2010), pp. 618–635. DOI: 10.1109/TEVC.2009.2034647.

[301] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. arXiv:1312.6114 [stat.ML]. 2013.

[302] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. Curran Associates, Inc., 2014, pp. 2672–2680.

[303] Alec Radford, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks". In: *arXiv preprint arXiv:1511.06434* (2015).

[304] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. "Learning Structured Output Representation using Deep Conditional Generative Models". In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, Inc., 2015, pp. 3483–3491.

[305] Mehdi Mirza and Simon Osindero. "Conditional generative adversarial nets. arXiv 2014". In: *arXiv preprint arXiv:1411.1784* (2014).

Previous Publications

[306] Nils Kroener, Kevin Förderer, Manuel Lösch, and Hartmut Schmeck. "State-of-the-Art Integration of Decentralized Energy Management Systems into the German Smart Meter Gateway Infrastructure". In: *Applied Sciences* 10.11 (2020). ISSN: 2076-3417. DOI: 10.3390/app10113665.

[307] Kevin Förderer, Manuel Lösch, Ralf Növer, Marilen Ronczka, and Hartmut Schmeck. "Smart Meter Gateways: Options for a BSI-Compliant Integration of Energy Management Systems". In: *Applied Sciences* 9.8 (2019). ISSN: 2076-3417. DOI: 10.3390/app9081634.

[308] Kevin Förderer and Hartmut Schmeck. "State-based load profile generation for modeling energetic flexibility". In: *Energy Informatics* 2.1 (2019), p. 18. ISSN: 2520-8942. DOI: 10.1186/s42162-019-0077-z.

[309] Nico Lehmann, Emil Kraft, Clemens Duepmeier, Ingo Mauser, Kevin Förderer, and Dominique Sauer. "Definition von Flexibilität in einem zellulär geprägten Energiesystem". In: *Zukünftige Stromnetze, Berlin, 30.-31.01.2019*. Jan. 2019. DOI: 10.5445/IR/1000090455.

[310] Katharina Volk, Christian Lakenbrink, Linda Rupp, Simon Eberlein, Joe Imfeld, Florian Wellmann, Daniel Contreras, Pascal Wiest, Peter Stolle, Sebastian Kochanneck, Daniel Zech, Daniel Groß, Steffen Pfendtner, and Kevin Förderer. *Grid-control - Advanced Decentral Grid Control : gemeinsamer Abschlussbericht zum Verbundvorhaben*. Tech. rep. Suttgart, 2019. DOI: 10.2314/KXP:1677497149.

[311] David Wölfle, Kevin Förderer, and Hartmut Schmeck. "A Concept for Standardized Benchmarks for the Evaluation of Control Strategies for Building Energy Management". In: *Abstracts from the 8th DACH+ Conference on Energy Informatics*. Vol. 2. Energy Informatics 2. Sept. 2019, pp. 9–12. DOI: 10.1186/s42162-019-0098-7.

[312] Kevin Förderer, Mischa Ahrens, Kaibin Bao, Ingo Mauser, and Hartmut Schmeck. "Modeling flexibility using artificial neural networks". In: *Energy Informatics* 1.1 (Oct. 2018), p. 21. ISSN: 2520-8942. DOI: 10.1186/s42162-018-0024-4.

[313] Kevin Förderer, Mischa Ahrens, Kaibin Bao, Ingo Mauser, and Hartmut Schmeck. "Towards the Modeling of Flexibility Using Artificial Neural Networks in Energy Management and Smart Grids: Note". In: *Proceedings of the Ninth International Conference on Future Energy Systems*. e-Energy '18. Karlsruhe, Germany: ACM, 2018, pp. 85–90. ISBN: 978-1-4503-5767-8. DOI: 10.1145/3208903.3208915.

[314] Kevin Förderer and Hartmut Schmeck. "Demo abstract: a building energy management system in the context of the smart grid traffic light concept". In: *Computer Science - Research and Development* 33.1 (Feb. 2018), pp. 269–270. ISSN: 1865-2042. DOI: 10.1007/s00450-017-0376-1.

[315] Katharina Volk, Christian Lakenbrink, Nina Hatje, Peter Stolle, Filippos Sivorotka, and Kevin Förderer. "Evaluation of power flow prognosis methods for congestion management in low voltage grids". In: *Digital Proceedings of the 8th International Workshop on Integration of Solar Power into Power Systems*. Oct. 2018. ISBN: 978-3-9820080-0-4.

[316] Ingo Mauser, Jan Müller, Kevin Förderer, and Hartmut Schmeck. "Definition, Modeling, and Communication of Flexibility in Smart Buildings and Smart Grids". In: *ETG-Fb. 155: International ETG Congress 2017* (Bonn, Germany). VDE, Nov. 2017, pp. 605–610. ISBN: 978-3-8007-4505-0.