Automated Generation of Models for Demand Side Flexibility Using Machine Learning: An Overview

KEVIN FÖRDERER, Karlsruhe Institute of Technology, Germany and FZI Research Center for Information Technology, Germany
VEIT HAGENMEYER, Karlsruhe Institute of Technology, Germany
HARTMUT SCHMECK, Karlsruhe Institute of Technology, Germany and FZI Research Center for Information Technology, Germany

Flexibility in consumption and production provided by distributed energy resources (DERs) is a key to the integration of renewable energy sources into the energy system. However, even for identical DERs, the flexibility can vary widely, based on local constraints and circumstances. Therefore, handcrafting models can be labor-intensive and automating the generation of models could help increasing the volume of controllable flexibility in smart grids. Depending on the underlying mechanism for controlling demand side flexibility, there are various ways how an automation can be achieved. In this paper, we discuss fundamental concepts relevant to the automated generation of models for demand side flexibility, give an overview of different approaches, and point out fundamental differences. The main focus lies on model generation by means of machine learning techniques.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Hardware → Smart grid**; • **Computing methodologies → Modeling methodologies**.

Additional Key Words and Phrases: Demand Side Management; Machine Learning; Automated Model Generation

ACM Reference Format:
Kevin Förderer, Veit Hagenmeyer, and Hartmut Schmeck. 2021. Automated Generation of Models for Demand Side Flexibility Using Machine Learning: An Overview. 1, 1 (November 2021), 13 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Renewable energy sources (RES) play a key role in the decarbonization of the energy system. However, they also pose new challenges. One major challenge is balancing the supply and demand of electricity, caused by the fluctuating nature of solar and wind power [44]. Another challenge is the integration of this decentral generation into distribution grids, as the additional power feed-in can cause grid congestions and voltage violations [30, 32]. In order to tackle these issues economically, without simply installing more and more grid capacity wherever RES are installed, the former demand side must partake in the effort of balancing electricity supply and demand, as well as alleviating grid congestions and voltage violations.

Measures for influencing energy demand are generally covered under the term demand side management (DSM) and include procedures for direct and indirect control of distributed energy resources (DERs) [36], i.e., storages, generators, and controllable loads. Control is exercised by some demand side manager (DSMgr). Depending on the regulatory framework and potential contractual agreements, examples for DSMgrs include distribution system operators (e.g., [11]), virtual power plant operators, and regional or district automated energy management systems (EMSs). In order to plan interventions and perform this control, a DSMgr needs to know their available courses of action, that is, models from which options to influence DERs can be derived. Furthermore, these models of demand side flexibility need to be increasingly detailed, the more accurate a DSMgr wants to plan and control. The demand side, on the other hand, is very heterogeneous, as individual providers of demand side flexibility operate different DERs combined in different ways under varying operational constraints. Consider different kinds of production sites, office buildings, or residential homes for example. Crafting new models by hand or configuring existing models for each site is indeed possible, but labor intensive and needs at least some expert knowledge. If, instead, the required models could be generated or parameterized automatically, the acquisition of customized models for the available flexibility would be simplified, potentially leading to more overall usable demand side flexibility.

Before we further outline the prospects of automatically generating such models, let us first briefly discuss the term flexibility. Demand side flexibility, or simply "flexibility" in the context of smart grids generally either refers to the capability of influencing the operation of DERs (in the sense of "being flexible") or a description of how DERs can be controlled (characterizing "how flexible" it is) [31, 37]. One such description is a set of feasible load schedules. A feasible load schedule is a schedule a DER or an ensemble of DERs can reproduce, while satisfying all operational constraints. Hence, the set of feasible load schedules details all choices a flexibility provider can offer to an external entity, such as a DSMgr. Understanding this set as flexibility is a simplification of the definitions found in [31, 37, 48, 56] and the same concept is applied in [2, 3, 41], for instance.

Making demand side flexibility readily available is a key to the integration of RES. With sufficient flexibility from energy storages and demand, located at suitable positions in the distribution grid, the curtailment of generation from RES can be reduced or even avoided

Authors' addresses: Kevin Förderer, kevin.foerderer@kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany and FZI Research Center for Information Technology, Karlsruhe, Germany; Veit Hagenmeyer, veit.hagenmeyer@kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany; Hartmut Schmeck, hartmut.schmeck@kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany and FZI Research Center for Information Technology, Karlsruhe, Germany.

XXXX-XXXX/2021/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

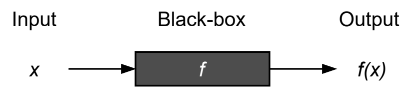

Fig. 1. A black-box model. The function f is learned with the goal of approximating given target values y for given inputs x.

by increasing demand and storing excess energy. Shortages, on the other hand, can be dampened by decreasing demand and releasing energy from storages. This coordinated control can be achieved or incentivized with different approaches, e.g., the aforementioned direct and indirect DSM measures. The automated generation of models could help in increasing the volume of controllable flexibility in a smart grid, especially when detailed descriptions are desired, as it would automate one of the most labor-intensive steps in integrating DERs into a coordination mechanism. Each model encodes the flexibility of a set of DERs and provides the necessary information for deriving operational choices. It may be generated by the associated flexibility provider themselves, i.e., the owner of the DERs, or some other, external party. The approach of using generated models is especially promising in combination with flexibility providers who employ similar methods. If, for instance, the local, automated EMS of a flexibility provider uses model-based reinforcement learning, it already has learned a model of the DERs' dynamics, which could be passed on.

There are different options for the automated generation of models, for instance preparing a comprehensive model by hand and automating the parameterization, that is, using a grey-box approach. Grey-box models are combinations of white-box and black-box models [17]. While white-box models transparently describe the dynamics of a system, black-box models approximate the system output from a given input [17]. In other words, with a white-box model it is possible to analyze and explain in detail how the output is generated from the input. A black-box model, on the other hand, has an unknown inner behavior so that only inputs and outputs are known for certain. Such a black-box is depicted in Figure 1. Black-box models are statistical models and learned from data [17]. In this paper we view flexibility providers and their DERs from a black-box perspective and focus on machine learning, as this allows for great versatility. Using supervised learning, the mapping f, i.e., the black-box model, is learned from given input and output data (x, y). The goal is to find an f such that the model output $f(x)$ approximates the true output y with minimal error, e.g., $\|f(x) - y\|$. Since the black-box models act as surrogates for the actual DERs or hand-crafted models, they are also referred to as surrogate models, in this paper.

With this paper, we aim to motivate and aid future research on the automated generation of models for the flexibility of DERs by means of machine learning techniques. There are various approaches for encoding flexibility in a learned model, i.e., many ways to generate such a model [16]. Based on our previous work and conducted review of general mechanisms for the exploitation of flexibility [16], we present and distinguish different modeling options, putting our previous results in a broader context and creating an overview of the different concepts and their associated use cases.

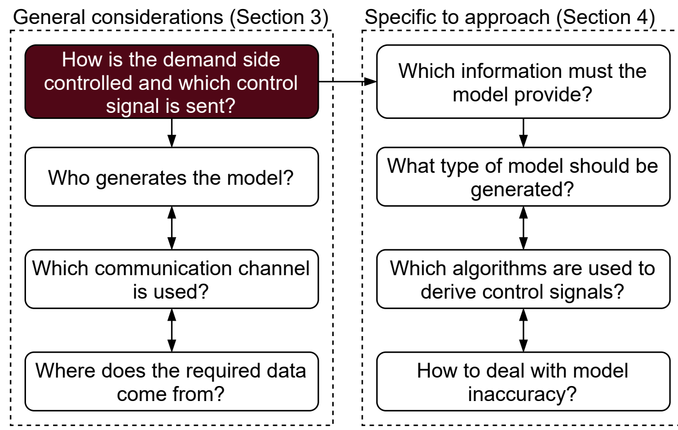

Fig. 2. Questions to be answered in the process of developing an approach for the automated generation of models for demand side flexibility.

Furthermore, in this paper, we outline and illustrate fundamental differences between the approaches and point out important aspects to be considered when conceiving new methods for the automated generation of models for demand side flexibility.

Figure 2 outlines the fundamental questions discussed in this paper and shows how they are connected from a general perspective. Each question is associated to one step in the development of an approach for the automated generation of models for demand side flexibility and each answer influences the others. Starting point is the selection of an exploitation approach, which is closely tied to the specific use case and the specification of a control signal. The questions are separated into two groups: a group of more general considerations on the left and a group of considerations specific to an exploitation approach on the right. Interconnections between the individual questions of both groups do certainly exist and may be relevant in specific cases, but are not depicted. The paper's structure is based on this distinction. In Section 2, related work is presented, outlining the current state of research in the field. Afterwards, in Section 3, the general questions are discussed: we show how a DSMgr can exploit demand side flexibility, illustrate the consequences of who generates the model and discuss the challenges of acquiring data and detecting flexibility in observed data. Section 4 is dedicated to use case specific consideration. It provides an overview of modeling approaches using machine learning models and discusses important aspects. Finally, the paper is concluded by Section 5.

2 RELATED WORK

The wish to automate model generation processes is not exclusive to energy related topics. "Automated model generation (AMG)" is investigated in, but not limited to, the context of integrated circuit verification and testing [26, 55]. The generated model, e.g., in form of differential (algebraic) equations, replicates input-output characteristics of complex circuits [55].

The generation of simulation models with "flexible structures", in the sense of constructing the model structure dynamically during the generation process, is investigated in [26]. The generation process involves transforming the given data into a graph, specifying how individual components from an existing "domain model component library" are coupled, model instantiation, and model calibration [26]. Similar approaches are conceivable for the modeling of demand side flexibility. However, in this paper we focus on the training of machine learning models, instead of the utilization of predefined components.

Automated model generation is also applied in energy related fields. One example is given by [35], where a generic space heating model representing an entire building as a single heating zone is presented. The model has a fixed structure. In order to generate an actual model for a building, the missing parameters are estimated from the given input data. In [9] models for evaluating the performance of heating, ventilation, and air-conditioning in buildings are generated from data. The input data is given according to a custom data model. From this data, a component-based system description in the form of a network of components is derived. Model generation for district heating systems is explored in [19] and [18]. The proposed generation process starts with the import of a given graph, e.g., OpenStreetMap data. Subsequently, a heating network is created based on the graph and parameterized. The generation process can be influenced with a set of configurations, like the number of connected buildings and probabilities for houses to be connected to the network [19]. Another example for an automatically generated thermal grid model is presented in [49].

Regarding DSM and flexibility, automated model generation is closely related to the detection and quantification of (potential) flexibility, as both derive information about flexibility from given data. Flexibility may be subdivided into different categories (see also [16]), such as energy flexibility and time flexibility [33, 34]. While energy flexibility means adaptability of power or running time, time flexibility means the possibility to change the starting time [33, 34]. In [33, 34], the (potential) energy and time flexibility of a building of a chemical factory are derived from load consumption time series by means of motif discovery. Motifs in this context means reoccurring patterns in time [33, 38]. The results are statistics for the length, power intensity, as well as the potential starting times. Nevertheless, detecting and learning about flexibility in recorded data is a challenging task, as discussed in Section 3.5.

Encoding the feasible set of DERs into surrogate models for the purpose of communicating flexibility was first proposed in [6]. The authors trained support vector data descriptions (SVDDs) to distinguish between feasible and infeasible load schedules. The basic idea of SVDD is to enclose a given dataset with the smallest possible sphere within a so-called feature space [53]. Given an SVDD model of the feasible set and an arbitrary load schedule, the schedule can be classified by checking whether it lies within the sphere or not. If it lies outside, the schedule is infeasible. Projecting infeasible schedules onto or into the sphere and calculating the pre-image, that is, a potentially feasible schedule, within the data space is proposed in [7, 8] and thoroughly investigated in [3]. This SVDD-based "repair" mechanism has since been extended (e.g., [2, 43]). A similar approach by the same authors makes use of Chi-shapes instead, that is, polygons enclosing a dataset (similar to a convex hull) [4].

A classification-based approach to encode the feasible set is proposed in [41] and further investigated in [40, 42]. It makes use of a cascade of classifiers to determine whether a schedule is feasible or not. Each classifier evaluates a specific part of the load schedule and the schedule is deemed feasible if all classifiers label their part of the schedule as feasible [41]. Different types of classifiers may be used, including support vector machines, one-class support vector machines, and artificial neural networks (ANNs) [40, 41, 41].

Inspired by [6], we first proposed to model the flexibility of DERs by encoding it into ANNs in [13]. For this purpose, we conceived and outlined different "usage patterns", which could be used by a DSMgr in order to derive control signals from a given ANN. The patterns are evaluated in [12] and based on these results another, similar pattern is proposed and investigated in [15]. Finally, a more systematic approach to the topic is presented in [16], where opportunities for the utilization of ANN-based surrogate models are identified. Additionally, a more thorough evaluation of the idea of ANN-based surrogate modeling is presented.

Data-driven models can also be found in the field of model predictive control. In [1] a review and case study of the ANN-based optimization of heating, ventilation, and air conditioning system is presented. The utilization of automatically generated hybrid automata is proposed in [50] and evaluated for a heat pump and a boiler. In contrast to these works, which generate models for the local, automated building energy management, in this paper, we focus on models used by some external entity in order exploit demand side flexibility.

3 EXPLOITATION OF DEMAND SIDE FLEXIBILITY WITH AUTOMATICALLY GENERATED MODELS

The first step towards the utilization of automatically generated models for the exploitation of flexibility, i.e., the direct or indirect control of DERs, is the selection of an existing or conception of a new exploitation approach. A categorization of different approaches found in the literature is presented in the following.

3.1 Approaches for the Exploitation of Demand Side Flexibility

A variety of different approaches for the exploitation of flexibility can be identified, using different characteristics [16, 28, 31, 37]. In this paper, we make use of the classification proposed in [16], which is based on [37]. In summary, five different categories of approaches for the exploitation of flexibility, also named "patterns", are distinguished [16]:

Direct exploitation: DERs are directly controlled by some external entity. In order to be able to do so, the external entity uses a predefined interface and may even be able to request and collect information relevant to the DERs operation. As an example, consider an aggregator who controls individual DERs using custom interfaces for different DERs and flexibility providers.

Exploitation of abstracted flexibility: DERs are controlled with the help of highly abstracted models, meaning the models are not specific to a fixed DER type, i.e., a generator or a storage, but instead suitable to describe multiple types of DERs or an entire ensemble of DERs. One example from this category would be the curtailment of generation and shedding of load with a reduction or on/off signal. Another example is the activation of automatic Frequency Restoration Reserves, which is implemented via a setpoints.

Market-based exploitation: How DERs are operated is decided by some market mechanism. Flexibility providers submit bids or requests to a market and receive the result after market clearing. Examples are energy markets and balancing energy markets.

Indirect Exploitation: Flexibility providers receive a signal incentivizing them to change their behavior and reschedule their DERs. Incentives are usually provided in the form of price signals. Dynamic tariffs are an example for an indirect exploitation approach.

State information-based exploitation: DERs are operated based on some state information observed or received by the flexibility provider. This exploitation pattern is usually employed in decentralized or distributed approaches. One exemplary application is the activation of the Frequency Containment Reserve, which is coupled to the observed power grid frequency.

The examples of Frequency Containment Reserve and automatic Frequency Restoration Reserve show that multiple patterns can be combined, as the monetary compensation for providing balancing power is determined with the help of an auction (market-based exploitation). Based on this classification, we conducted an extensive literature review, which is presented in [16], in order to identify common modeling approaches for the exploitation of demand side flexibility. It is important to note that the approaches are not limited to the automated generation of models. The result, which was compiled from a total of 173 analyzed publications, is depicted in Figure 3.

It shows the different classes of exploitation approaches (blue boxes), the associated modeling approaches within this class (white boxes), as well as different communication schemes, and relates them in terms of their abstractness. Abstractness (x-axis) was judged, based on the following two statements:

- "Abstraction is a process of generalization, removing restrictions, eliminating detail, removing inessential information (such as the algorithmic details)." [54]
- "Abstract specifications have 'more potential implementations', moving to a lower level means restricting the number of potential implementations." [54]

On the y-axis, the approaches are grouped according to their genericity: from specific to a single DER or type of DER to generic approaches applicable to any DER.

Please note that Figure 3 does by no means capture every possible exploitation mechanism. Instead, it depicts and classifies the approaches commonly found in the literature. Moreover, it is possible to devise exploitation mechanisms that do not fit into this framework, simply by creating novel combinations of the depicted elements. For a detailed discussion of each element, we refer to [16]. A short summary of the approaches is provided in the following [16]:

Precise model: Precise models are developed for each DER or type of DER in order to closely describe their dynamics.

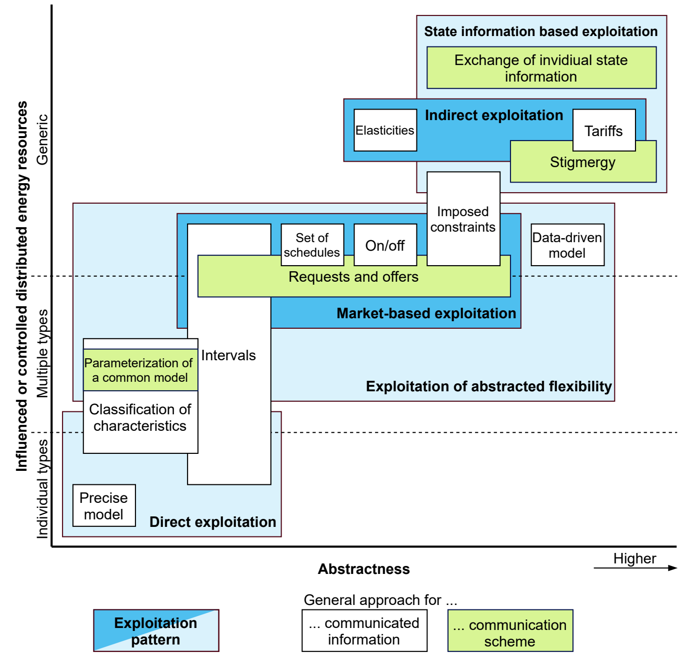

Fig. 3. Common modeling approaches for each exploitation pattern. Source: [16].

Classification of characteristics: DERs are assigned characteristics, such as "shiftable", "curtailable", or "interruptible", and are modelled accordingly. Therefore, different DERs or types of DERs may share identical models.

Parameterization of a common model: A communication scheme in which flexibility providers describe their flexibility to an external entity by determining and sending the parameters of a predefined and commonly known model.

Intervals: Models are built around or from intervals, e.g., intervals for power, energy, and ramping capacity.

Set of schedules: DERs and their flexibility are described by a list of feasible load schedules. External entities simply choose the most beneficial entry.

On/off: DERs can be stopped and started.

Imposed constraints: External entities restrict the operation of DERs by setting operational constraints, e.g., a feed-in limit. Usually only one limitation is active at a time for either flow direction, that is, feed-in or consumption.

Requests and offers: A communication scheme in which flexibility providers explicitly offer flexibility, e.g., on a market, and/or external entities request the utilization of flexibility, for instance based on a contract.

Data driven: Flexibility is exploited in a data-driven way, for instance by learning a model and then using it to determine control signals. This is one of the key approaches for the automated generation of models.

Elasticities: Price-elasticities can be used to predict the reaction of a flexibility provider to price changes. With the help of these predictions, different price signals can be evaluated before selecting one.

Tariffs: Dynamic tariffs allow the adaptation of prices in order to influence the demand side and exploit its flexibility indirectly [37]. In the literature, the existence of dynamic tariffs is frequently assumed, but rather rarely the investigated subject.

Stigmergy: Stigmergy is a coordination approach making use of anonymous and only indirect communication via the manipulation of a shared environment [47]. In case of the Frequency Containment Reserve, the electrical grid is the shared environment and the observed information is the grid frequency. Each reserve provider manipulates the frequency by a tiny fraction via their load, but it is not possible for a reserve provider to tell what others have contributed.

Exchange of individual state information: A communication scheme in which flexibility providers share their state information, such as their planned load schedule. Other flexibility providers can use this information and adapt their own schedules in order to pursue a collective goal. In a distributed approach, flexibility providers are external entities to each other.

These categories and general approaches are vital for the definition of procedures for the automated generation of models for demand side flexibility. A selection of general concepts is presented in Section 4. For now, let us assume, we know the pattern the external entity uses to exploit flexibility, what kind of model should be generated, and which kind of data is required. The next step is then model generation, which poses the questions of who should generate the model, that is, the flexibility provider or the external entity, and from where the necessary data should be sourced.

3.2 External Entities Generate Models from Observed Data

In case models should be generated by a DSMgr or any other external entity, they must acquire the necessary data, before they are able to do so. Data can be acquired in different ways: firstly, it may be observed with the help of a (advanced) metering infrastructure. The infrastructure measures energy usage and can provide this information to the external entity, who stores the received individual measurements or load profiles. This data may, however, include inflexible power flows, which are not related to DERs. Secondly, the flexibility provider could periodically send the latest dynamic DER data to the external entity. Examples for dynamic data from DERs are state information, such as the state of charge of a storage or the operation status of a heat pump. The downside of this approach is that the flexibility provider is at all times completely transparent to the external entity. Lastly, sets of data may explicitly be requested from the flexibility provider. If the exact date of measuring the data does not matter in the intended exploitation approach, the data could be partitioned, the partitions shuffled, and date information removed, in order to obfuscate at least some behavioral patterns of the flexibility providers DERs.

The process of model generation by an external entity and subsequent flexibility exploitation is depicted in Figure 4. After acquiring a sufficient amount of data, using one or multiple of the options outlined above, the external entity generates the model. Which amount of data is sufficient, strongly depends on the employed machine learning model (see Section 4). The generated model is then stored for later use. Whenever DERs need to be controlled, that is, periodically or in case a specified event is detected, the model is loaded and used to derive control signals. Any type of signal, such as tariffs, load schedules, or operational constraints, could be derived given a suitable model (see Section 4). The signal, i.e., the "flexibility choice", is then sent to the EMS of the flexibility provider, which replans and controls the DERs accordingly. A feedback mechanism may optionally be implemented in order to detect discrepancies between model output and actual flexibility. Discrepancies arise since the models only approximate the real flexibility. Depending on the quality of the generated model, this approximation may be very rough. As an example, consider an EMS receiving a load schedule from an external entity. The EMS checks whether it can reproduce the schedule by means of rescheduling the local DERs and sends back the closest schedule it can achieve. The feedback can be collected to improve the respective model and may trigger a replanning process, in case the deviation is too large for the intended use.

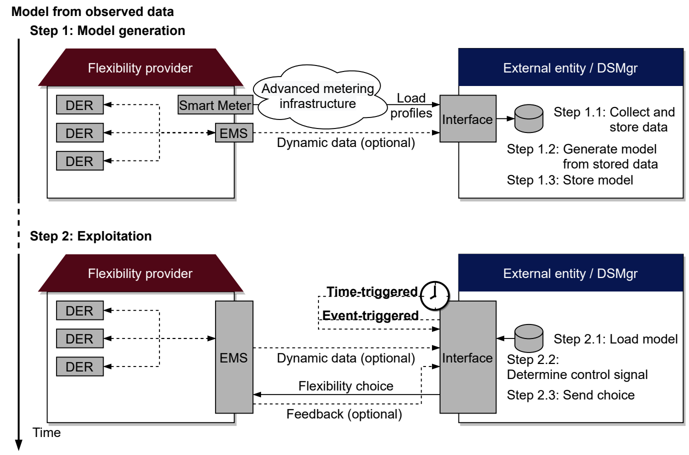

Fig. 4. An external entity collects data and generates a model (**Step 1**), which is then used to derive control signals (**Step 2**). Based on [16].

3.3 Flexibility Providers Generate Models

Models may also be generated by the flexibility provider and sent to the external entity. A major advantage is the reduced amount of communicated data, as only the generated model needs to be transferred, instead of the training data. Hence, this decentralized model generation enables the utilization of additionally generated, artificial data in order to improve model accuracy. There are also privacy related implications: on the one hand, privacy is improved in comparison to the centralized model generation, as the training data is not directly available to the external entity. On the other hand, the communicated model may be exploited to draw conclusions about the dynamics and constraints of the DERs, and thus the behavior and habits of the flexibility provider.

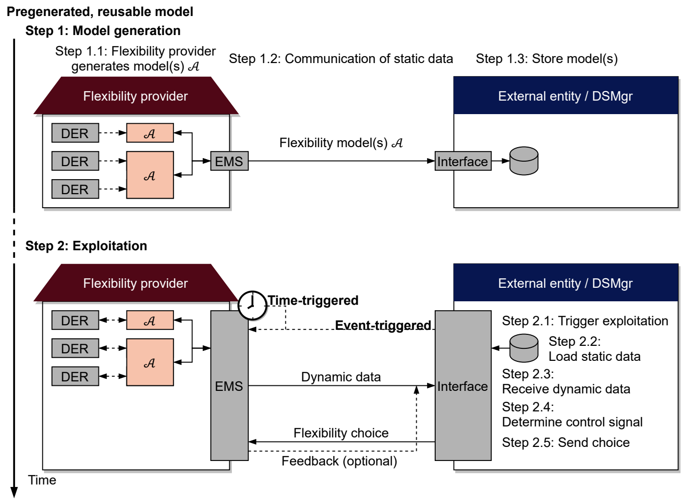

Fig. 5. The automated EMS of a flexibility provider generates reusable models $\mathcal{A}$ of the controlled DERs and communicates them to an external entity, where they are stored for later use (**Step 1**). Whenever required, control signals, that is, "flexibility choices", are derived with the help of the stored model and the received dynamic state variables (**Step 2**). Based on [16].

The automated generation of models for flexibility is an especially promising concept if it is employed on multiple hierarchical levels and makes use of similar models on each level. As an example, consider a flexibility provider who optimizes the operation of DERs with the help of model-based reinforcement learning. This means that the flexibility provider has already trained a model of the DER dynamics (see [51]). Such a model could be passed to an external entity, that is, upward in the control hierarchy, and additional information or models may be provided to further facilitate signal generation.

A decentralized model generation can be implemented in two different ways: firstly, we can utilize reusable models, which only need to be generated whenever there are changes to the DERs and their operational constraints. For a model to be reusable, we must be able to pass state variables, that is, dynamic data determining the actual flexibility within the frame of theoretical flexibility given by the model and its (static) parameters. Take a battery energy storage system, for example. The amount of energy that can be charged or discharged is determined by technical properties and its current state, e.g., its nominal capacity and the state of charge. The nominal capacity is a parameter and can be communicated upfront, as it remains fixed, but the state of charge needs to be transmitted timely, whenever flexibility needs to be assessed.

Figure 5 depicts the utilization of reusable models. Models are generated by the EMS of the flexibility provider in such a way that they remain valid as long as the underlying parameters remain unchanged. As stated before, this can be achieved by generating models in which state information can be incorporated dynamically. Consider an ANN-based model which takes the state variables as an input, for instance. The resulting models are sent to the external

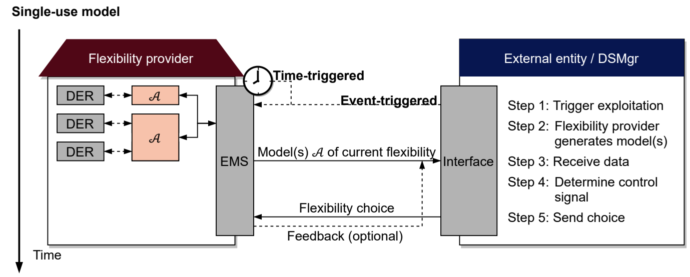

Fig. 6. The automated EMS of a flexibility provider generates state-specific models $\mathcal{A}$ of the controlled DERs and communicates them to an external entity. The models are immediately used to derive control signals. Based on [16].

entity and stored for later use. In case the DER dynamics change, e.g., when another DER is installed, the process is repeated.

Whenever flexibility is to be exploited, that is, when a given event is detected or periodically, the stored models are loaded and the flexibility providers are requested to send their current state information. Using this data, control signals are generated and sent to the flexibility providers' EMSs, which in turn may provide feedback, analogous to Figure 4 and the external model generation.

If the generated model encodes the current state of the DERs, it is only valid in this exact state. In this sense, the model can only be used once, as the next time flexibility needs to be assessed, the DERs will likely be in another state. Therefore, new models need to be generated and communicated for each single exploitation attempt. An example for such a model is the set of feasible schedules encoded into an SVDD (see [3]). The complete exploitation process is depicted in Figure 6. Since all dynamic information is already encoded in the model, it does not need to be transmitted separately. Furthermore, there is no need to store the model, as every time a new one is generated.

In comparison, single-use und reusable models have different advantages and disadvantages: while a reusable model only needs to be generated once, it must be able to capture more complex dynamics, since it needs to be valid for a wide range of different states. If the model is learned from data, this means that training data from many different states is needed. As a consequence, the training process takes longer and needs more computational resources, since the relationship of state and flexibility must be captured. This challenge may also be reflected in model quality. In general, the reusable model is likely to perform worse in a given state than a single-use model trained to describe exactly this state (compare [16] and the results of [16] and [3]). However, since the reusable model is not immediately needed by the external entity, there is time for a thorough training. Vice versa, the generation of the single-use model must be fast (seconds to minutes), to be applicable on short notice. But, since only a description of the current flexibility is needed, simpler models and therefore fewer computational resources can suffice for generating each individual single-use model. As an example, compare again SVDDs generated from a few thousand samples [3] and ANNs generated from hundred thousands of samples over the course of many training epochs [16].

3.4 Communication Channel

Receiving dynamic data from the EMS, sending control signals, and receiving feedback requires communication between EMS and the external entity. The external entity provides the necessary interfaces for the exchange of information. Technical limitations regarding the communication channel may be posed by the amount of data to be communicated, data transfer rates, reliability, and the mere existence of the communication infrastructure in remote areas. While the data may be communicated via the Internet, communication channels isolated from the Internet should be considered. The German Smart Meter Gateway infrastructure, for example, is able to provide a secure channel for such communication (e.g. [14, 29]).

3.5 Detecting Demand Side Flexibility in Data

If one were to handcraft custom models for individual flexibility providers, one would usually examine the associated DERs, inspect their environment, and talk to operators and users in order to determine constraints and record their preferences. Automating these steps requires collecting the same information from given data.

In the most basic case, only historic load profiles are given, from which we need to estimate parameters (grey-box approach) or learn a mode from scratch. Imagine we want to quantify how flexible an office building can be operated, solely by looking at observed load profiles. When there is feed-in and it follows a curve with a peak around noon, we can plausibly assume that there is a photovoltaic system present, but does the absence of such a feed-in curve mean that there is no photovoltaic system? The answer is: not necessarily. The installed generation capacity could simply be too small to outweigh the demands during office hours. There could also be a battery energy storage system, preventing feed-in and storing energy for later use. Only if there are days, in which we see the characteristic curve from solar generation in the load profile, we can conclude that there probably is a photovoltaic system. Then, using this information we can try to find evidence for batteries. This example illustrates the two core challenges of learning flexibility from load profiles alone:

(1) In order to detect flexibility, it must show in the data. If DERs are always operated in the same way in a given situation, we cannot know for sure whether there are alternative operational choices or not.
(2) Load profiles alone can only provide clues of the actual flexibility. Further assumptions, e.g., made by scanning for characteristic sequences, or information, for instance given by some master data register, are needed.

The first challenge is related to the exploration-exploitation dilemma found in reinforcement learning (compare [51]): an agent has to explore alternatives to the already known, well performing operational choices in order to discover new, potentially better options. Translated to the operation of DERs, this means that the normal, daily operation needs to be disturbed in some way from time to time, in order to force the selection of load schedules unseen before. The resulting load profiles then provide information about the available flexibility, when compared to periods without disturbances, especially if details about the disturbances are known. If, for instance, charging or discharging the previously considered battery energy storage system is restricted on some days, there will be days in which electricity generation from solar power is not concealed. Now, if we additionally know that charging was restricted, the difference in consumption gives us a clue about the flexibility. Please note that it is not necessary to know about the storage itself, it suffices to know that there was a restriction.

Knowingly causing disturbances in DER operation effectively generates additional information, which brings us to the second challenge: we need further assumptions or information to confidently draw conclusions about the available flexibility. For instance, one could try to disaggregate the collected load profiles or try to identify characteristic patterns (compare [33]) and conduct further analyses based on the results. Looking at the (imaginary) load profiles from our introductory example and reasoning that there likely is a photovoltaic system and a battery energy storage system equates to making assumptions. Based on these assumptions we can try to quantify the flexibility offered by the battery, but we cannot know for sure that the battery actually exists. In other words, estimates are generated on the basis of estimates. In order to make more confident statements, more definitive information is needed. Basic DER information could alternatively be given by hand or automatically collected from some master data register like the Marktstammdatenregister in Germany, an official register for power plants and battery energy storages.

Overall, the more information is available, the more certain and accurate conclusions can be drawn. Information cannot only be generated during model generation, but also when the model is used to control the DERs of a flexibility provider. Whenever a flexibility provider is not able to implement the schedule changes derived from the learned model, we gain new data, which we could use to improve our model. Additionally, since it is clear that the models will only provide estimates, the utilization of stochastic models should be considered. If, for instance, probability distributions or likelihoods are given, a DSMgr could compute confidence intervals when planning how DERs should be controlled.

3.6 Sources for Data

To generate a model from data, data is needed in the first place. Which data is required strongly depends on the intended use case (see Section 4). In case of the external model generation, data may be collected locally by the EMS and sent to the external entity periodically ("dynamic data" in Figure 4). From a general perspective, there are multiple possible ways to collect data:

Observation during operation: Automated EMSs manage energy flows in some optimized way. To do this, the EMS must observe sensor inputs and collect state information from DERs. In combination with the control signals issued by the EMS itself or the observed reaction of the DERs, a model of the DERs' dynamics can be trained. However, the issue of determining flexibility from observed data, as discussed in Section 3.5, persists. Recording not only the DERs reaction to the optimized controls, but also feasible, alternative options, is one possible solution to this problem. Such alternative options, for instance in the form of a set of feasible actions, could be provided by some future, standardized DER interface. Depending on the exact use case and modeling approach, different data is required.

Querying from DER: If such a standardized interface allows passing a presumed state as an argument, the necessary information may be collected directly from the DER independent of the true state and actual operation.

Fake data points: One option to introduce flexibility into the dataset is adding artificial data points. The creation of fake data for the sake of training better models is called dataset augmentation [22]. Depending on the modeling approach, there can be many different options for the generation of fake data. The major challenge is creating valid fake data, or else the learned model is at least partially invalid, that is, it does not represent the flexibility correctly.

Extraction from existing models: In order to control and optimize local energy flows, the automated EMS makes use of optimization algorithms and models. It would be possible to extract information on operational constraints, e.g., maximum and minimum power in a given state, from these existing models. If an approach for the automated generation of surrogate models was established, there could be a standardized interface for extracting all required information from optimization models.

Simulation models can be another source of data (e.g. [3, 12, 16]). We trained surrogate models from newly developed simulation models in our previous work [12, 16] to demonstrate the viability of the concept of ANN-based flexibility models. However, manually creating a simulation model solely for generating a surrogate model defies the purpose of surrogate modeling in this context.

4 DIFFERENT USE CASES AND MODELING OPTIONS

With the general considerations from the previous section in mind, let us now look at different options for generating descriptions of demand side flexibility. From Figure 3 in Section 3 we can derive three basic options:

Generating a mathematical model: An algorithm or artificial intelligence able to generate mathematical models (e.g., genetic and evolutionary programming [52], "evolvable mathematical models" [24], automatic generation of hybrid automata [50], or some regression algorithm) could be used to implement modeling approaches from the direct exploitation category in Figure 3. If the model is generated by the flexibility provider, the resulting mathematical model needs a common interface, that is, specific variables that are known, manipulated, or observed by the external entity. Without these, the external entity would not able to make use of the model.

Parameterizing a common model: If the model structure is predefined and fixed, specific models are generated by estimating the missing parameters, which is a process that may be automated. This approach is mostly suitable for the abstracted flexibility category in Figure 3, where models with a higher degree of abstraction are used. There also may be multiple model structures to choose from, e.g., a battery, a bakery, and a bucket [45]. The selection of the best "template" could be achieved with the help of clustering methods.

Generating a (data-driven) surrogate model: For this, the goal is to generate a surrogate model, i.e., a black-box model, which in its inputs and outputs provides all the information needed to derive and select control signals. Control signals could be any signal, including load schedules, load deltas, and dynamic tariffs. Which signal is needed, how often a signal is sent, which time horizon it covers, and many other parameters depend on the exact exploitation approach. We refer again to Figure 3 for an overview of commonly found approaches and their associated signals. An overview of surrogate modeling approaches is provided in the next section.

Depending on the employed algorithms and whenever the model is generated step by step, e.g., evolution or gradient descent in the case of ANNs, the duration of model generation may be shortened for any of these options by starting from a pregenerated solution instead from scratch. Improving upon a prior solution can also help to improve model quality (see [22]), but may as well lead into a local optimum, potentially far away from the global optimum which would only be reached from a different starting point. In the context of ANNs, starting with pregenerated models is known as transfer learning and pretraining (see [22] for an introduction).

In the following we will focus on the third option for the generation of models, that is, the generation of surrogate models, which is usually achieved with the help of machine learning. With this option, demand side flexibility is encoded into a machine learning model, such as an ANN or support vector machine.

4.1 Surrogate Models for Demand Side Flexibility

Generally speaking, one could try to implement any approach depicted in Figure 3 with the help of surrogate models. Remember that from the black-box perspective the only thing we know about a given system is a set of system inputs and their associated system outputs. Hence, the main precondition is the possibility to derive adequate control signals from the considered model inputs and outputs, which in turn is only possible if the generated models are sufficiently accurate. Training accurate models requires an adequate amount of data, meaningful data (see also Section 3.5 on the issue of detecting flexibility), and a relationship between model inputs and outputs. Furthermore, the external entity using the model must be able to make sense of at least some of the data and know how the model needs to be used.

In [16], we derived and compiled a list of prospective surrogate modeling approaches for ANN-based surrogates by looking at the various types of models found in the literature and contemplating different input and output combinations. A generalization of this list, including many non-ANN examples, is given by Table 1. Table 1 is not exhaustive and many variations of the listed methods can exist. Furthermore, despite our best efforts, for some approaches no suitable examples could be identified in the literature, as the algorithmic generation and utilization of surrogate models as models

Table 1. Outline of possible surrogate modeling approaches. The generated model provides predictions of the necessary information. Based on: [16].

Signal	Method	Example
Schedule	Repairing infeasible schedule	[3, 4, 8, 12, 43]
Schedule	Generation from abstract representation	[12]
Schedule	Prediction of state and action trajectory	[10, 15, 16]
Schedule	Classifying feasibility of schedule fragments	[40–42]
Schedule	Classifying feasibility of entire schedule	[6, 12]
Schedule	Prediction of costs	-
Tariff	Predicting resulting load (schedule)	[12], [27]*
Load delta	Predicting how long load is changed	[21]*
Constraint	Predicting how long constraint is satisfied	[21]*
Other	Predicting resulting load (schedule)	-

*related

for demand side flexibility is still a niche approach (compare [16]) and existing, similar models are often hand-crafted or used in other contexts (see for example [25], where load forecasting in general is investigated).

The "Signal" column lists the type of signal the external entity sends to the providers of flexibility. In order to control flexibility providers and their DERs via load schedules, one must be able to identify feasible load schedules within the generally much larger space of all load schedules. Since we define flexibility as the set of all feasible load schedules, any surrogate model that allows us to identify feasible schedules can be seen as a model for the flexibility of DERs [12, 13, 16]. Table 1 lists six potential methods for identifying feasible schedules. For the sake of brevity, we only outline the different methods for signal generation. More elaborate explanations and illustrations of the generation processes are provided in [16].

Repairing infeasible schedule: Given a mapping from infeasible load schedules to (close-by) feasible load schedules, one cannot only identify feasible load schedules by passing any random input, but also search for desirable schedules by systematically checking different inputs. This concept is used by the SVDD approach [3], for instance. It can easily be integrated into optimization heuristics, such as particle swarm optimization, by searching within the (unconstrained) space of all schedules and performing a schedule repair just before evaluating the target function value [3].

Generation from abstract representation: A surrogate model may provide a mapping from some abstract (potentially random) representation to a load schedule, similar to the mapping from genotype to phenotype in evolutionary algorithms. As an example for a generative model, consider a generative adversarial network (GAN) [23]. A GAN maps random input noise to some artificial data point, such as an image [23]. Since any data space is valid, one may train a GAN to generate feasible load schedules. Furthermore, GANs can be trained to consider additional input variables [39], e.g., dynamic state variables, which makes the model reusable. With such a GAN at hand, the external entity could pass (random) inputs until a desired load schedule has been returned or until a given time or computational budget is exceeded and choose the best generated schedule.

Prediction of state and action trajectory: With a model of the state space it is possible to predict how the states of a flexibility provider's systems evolve. Starting from a given initial state, the external entity repeatedly determines a desirable power level (action) and the resulting subsequent states. Doing so, a schedule is generated power level by power level. Whether a power level (action) is feasible in a given state or not, can be detected with the help of an additional classifier [15, 16].

Classifying feasibility of schedule fragments: Classifiers for fragments of a load schedule are another option to identify feasible load schedules. A schedule is deemed feasible if every fragment is judged as feasible. One option to implement this method would be to train a list of classifiers, the first one judging only the first value, the second one judging the first two values, the third one judging the first three values, and so on. An external entity could then select value after value, always using the next classifier in line (see also [16]). Another option would be the utilization of the "cascade classifier" [40].

Classifying feasibility of entire schedule: A classifier able to distinguish feasible and infeasible load schedules allows us to identify feasible load schedules. However, randomly generating inputs most likely generates infeasible solutions, as the vast majority of possible schedules is infeasible. Additional information and systematic search methods are required to make use of this method. For ANN-based classifiers, for instance, one could try to use backpropagation to find input vectors, i.e., schedules, with a high likelihood of being feasible, that is, inputs yielding a high output value [16].

Prediction of costs: This method is closely related to the classification of an entire schedule. While the classifier rates whether a given schedule is likely feasible or not, with ratings from 0% to 100%, the cost prediction yields some value from the set of real numbers. In order to distinguish feasible and infeasible schedules, the model must be trained to penalize infeasibility with very high costs. By searching schedules with a cost below a given schedule, feasible schedules can be identified. As an added benefit, the estimated cost of implementing the schedule is known.

The length of the load schedules can be selected arbitrarily before model generation. Depending on the schedule generation method and the employed models, it may even be possible to change the length dynamically, as needed. One example for a method allowing dynamic schedule lengths is the *prediction of state and action trajectory*, in case the trajectory is put together one time step at a time. Then, the schedule generation can be stopped once the desired length has been reached. The length of the individual time steps, on the other hand, is generally fixed.

The remaining methods listed in Table 1 yield different types of signals and are not directly comparable. Furthermore, they are generally not suitable for the detailed control of DERs, as only incentives and operational boundaries are passed to the flexibility providers.

Prediction of load (schedule) resulting from tariff: A model predicting the resulting load schedule for a tariff or the resulting power level for a short term monetary incentive can be used to assess price signals. By comparing the expected reactions to price changes, the external entity can evaluate different tariffs and make more informed choices. However, depending on the modeled DERs, the changes between different tariffs may be rather small, which is one reason for the good results we achieved in [12]. In [27] the reward received by the external entity due to the changed schedule is predicted, instead of the load schedule itself.

Predicting how long load is changed: Short-term power changes can be evaluated by predicting if and how long the different changes can be implemented. For each flexibility provider, the desired change in Watts is passed to the surrogate model, which then provides the necessary predictions. Additional models or model outputs may be used to predict catch-up effects caused by each power change. In [21] a variation of this method is proposed, where the external entity computes how long a load can be interrupted.

Predicting how long constraint is satisfied: Limiting the (total) power drawn from or fed into the grid, for instance by issuing quotas of the maximum feed-in and consumption [11], is another possible approach to DSM. Here, the surrogate model predicts how long a given constraint can be satisfied by the flexibility provider. Again, additional models or model outputs could predict catch-up effects.

Prediction of load (schedule) resulting from signal: If flexibility is guided by the state of its environment, that is, flexibility providers observe the environment, interpret the state and react accordingly, one could try to manipulate this state in order to deceive flexibility providers. In this case, the surrogate model would predict the reaction to some defined, arbitrary signal. Please note that this case essentially is an abstraction of the tariff-based prediction.

The different presented methods for signal generation may be implemented with various types of (machine learning) models and in different ways.

4.2 Considerations for the Utilization of Surrogate Models

Different machine learning models come with different advantages and disadvantages. For instance, we can easily create conditional models with ANNs, where the output depends on some additional input variables (e.g., [12, 39]). On the downside, ANNs are comparatively expensive to train, as a large amount of data is needed [22]. In this section, we outline fundamental considerations important to choosing a model type and implementing a signal generation method.

Computational effort: The computational effort required to generate/train and use a model can vary widely between different types of models. In the first step, a model must be generated either by the flexibility provider or the external entity. While some models can be generated rather fast, such as a simple regression model, others need extensive training, e.g., some deep ANN. Higher computational effort may be mitigated by the generation of reusable models. As explained before (see Section 3.3), if a surrogate model can process and make use of dynamic state information, the model only needs to be generated once, every time the DERs or other underlying dynamics or constraints change. Single-use models, on the contrary, must be generated every single time flexibility is needed and the model generation process must be sufficiently fast for the given use case. Hence, even though initially more computation resources are required, a reusable model may need fewer resources in the long run. Additional computational resources are needed by the external entity in order to generate control signals and by the flexibility provider in order to implement the signal. However, these resource requirements heavily depend on the exploitation approach and intended signal generation method.

Model size: Model size is not only relevant for the external entity storing the model, but also for the transmission step and generation process. Different types of models need different amounts of data to describe them. For example, a support vector machine is described by a set of support vectors and some parameters. Even more compact is a simple linear regression model, as it only needs a few parameters. In contrast, the description of an ANN comprises its topology, neuron weights, activation functions, and further parameters. Especially in case of deep ANNs, the number of weights may grow very large, yielding large models requiring many megabytes of data.

Quality of approximation: A surrogate model in general only provides approximations of the true flexibility. How well this true flexibility is approximated depends on many different parameters, including the general capabilities of the selected type of model, successful training, and the data used during the training process (see Section 3.5). While some types of models may already yield good results from a relatively small amount of data, others may need much more. Furthermore, approximation is a more difficult task for reusable models than for single-use models, since additional inputs must be interpreted (see also Section 3.3).

Optimization: In general, an external entity is not interested in identifying just any feasible control signal, but instead wants to find signals beneficial to their overall goals. While it is possible to generate a set of random feasible load schedules or other possible signals for all flexibility providers and then select the best combination, a more systematic and targeted search approach is desirable. In case of the schedule repair method, one may try to identify the best possible schedule, ignoring all constraints, and then use the surrogate model to find a (hopefully) close-by feasible schedule. Such an approach is evaluated in [3]. When generating load schedules from abstract representations and similar representations (genotypes) lead to similar load schedules (phenotypes), the external entity may be able to optimize the schedules with

Table 2. Reusability of surrogate models in the literature.

Reusable	Signal generation	Model type	Example
Yes	Various methods	ANN	[12, 15, 16]
No	Classification	SVDD	[7]
No	Repair	SVDD	[3, 8, 43]
No	Classif. of fragments	Different classifiers	[40–42]
No	Repair	Chi-Shapes	[4]

the help of an evolutionary algorithm. Similar inputs producing similar outputs is the case for GANs [46], for instance. Next, for the state and action trajectory method, one may be able to apply algorithms from reinforcement learning, such as the Monte Carlo tree search [16]. How signals may be optimized strongly depends on the capabilities of the selected model type. An SVDD, for instance, describes data in the form of a sphere. It can be used as a classifier by checking if a point (schedule) lies outside the sphere and also for projecting points from outside onto or into the sphere (set of feasible schedules) (see [3]). ANNs, in comparison, are more versatile, but lack such a geometrically intuitive "repair" method. In [16], we suggest approaches for optimizing signal selection when using ANN-based surrogate models for the different signal generation methods listed in Table 1.

Scalability: A DSMgr generally manages a larger number of flexibility providers at the same time, whether their goal is to resolve congestions in the electricity grid or to plan and control the operation of a virtual power plant. For the DSMgr, that is, the external entity, this means that they must be able to process multiple surrogate models in parallel. Hence, efficient and scalable algorithms for signal generation are crucial, especially when signals must be determined fast. The aggregation of DERs or flexibility providers is one possible way to improve the scalability of an exploitation mechanism, as fewer signals need to be generated. Disaggregation of the signal may be performed on a subordinate level of the control hierarchy. How aggregation can be achieved (e.g., [5, 10]) and how existing surrogate models can be aggregated into a new combined surrogate model (e.g., [21]) depends on the model type.

As already pointed out in Section 3, some surrogate models are reusable while others are not, which more or less has an impact on every aspect discussed in this section. In Table 2, we list different types of machine learning models, associated examples, and point out whether the models are generated in a reusable way or not. Reusability is by no means limited to ANNs. For example, any regression model taking the DERs' states as an input could be reusable. However, the only surrogate modeling approaches with reusable models we could identify in the literature are ANN-based.

4.3 Dealing with Inaccuracy

As stated before, surrogate models for demand side flexibility only provide approximations of the underlying, true flexibility. Depending on the quality of a model, the majority of the derived signals may actually be infeasible. The only way for an external entity to know for sure whether a specific signal is feasible or not, is to ask the associated flexibility provider. This brings us to the question of how an external entity should deal with this issue. In the following we present building blocks for a possible solution:

Making use of a feedback mechanism: Figures 4, 5 and 6 all indicate the possibility of the flexibility provider sending feedback to the external entity. Such a feedback could, for instance, be the load schedule planned by the EMS as a reaction to the received signal. The external entity can then use this feedback to evaluate whether the predictions were accurate and whether corrective actions are needed.
Different kinds of corrective actions are possible, depending on the type of model. In case of reusable models, the feedback data may be used to improve the models by conducting further training steps. In doing so, it is also possible to deal with gradual or incremental concept drifts, as the model can gradually be updated. Concept drift refers to a changing relation between input data and target variable over time [20]. If an abrupt concept drift is detected, for instance caused by the installation of new DERs, the external entity may alternatively dismiss the current model and restart the model generation from scratch. For single-use models, concept drift is irrelevant, as the model is only expected to be valid for a very limited time.

Making use of statistical information: Some types of machine learning models are able to provide us with statistical information, such as the distribution of a predicted value or estimated probabilities. This information may for instance be used to predict confidence intervals or avoid choices for which the surrogate model predicts high uncertainty. None of the examples listed in Table 1 makes truly use of such a statistical approach. The only related application would be the limitation of the set of feasible schedules by adapting classification thresholds as discussed in the following.

Limiting the flexibility: In cases where the feasibility of a signal is more important than making available every last bit of flexibility, the artificial limitation of the encoded flexibility is a viable and effective option (see [15, 16]). This limitation can be achieved in multiple ways: firstly, by encoding only a subset of the true flexibility and thereby constricting the surrogate model itself [16]. Such a limitation can be incorporated by applying artificial constraints to the data used during model generation. As an example, think of preventing power levels near the nominal power of a DER. If later, in the application of the model, too large power levels are predicted, this artificial buffer can help to keep the prediction inside the true boundaries. A second possible way is to manipulate model inputs [15] in order to shift them closer to their boundaries, which again creates an artificial buffer. However, this option must be used with caution, as the opposite boundary is shifted away. Finally, in cases where classifiers are used, one may be able to tweak false positive and false negative rates. If the classifier returns a rating between 0 and 1 indicating the confidence of a match, e.g., a load schedule being feasible, one may increase the threshold in order to decrease the false positive rate. As a result, the false negative rate will generally increase, since now a more limited flexibility is described. Low false positive rates at the cost of higher false negative rates may also be achieved by knowingly overfitting the model to the training data [16]. Whether any artificial limitations are needed or not depends on the exact use case. However, we suggest to look into the alternatives first before applying such limitations, since prediction errors may cancel each other out when multiple flexibility providers are jointly controlled.

5 CONCLUSIONS

In this paper, we provide an overview how models for the flexibility of DERs can be generated in an automated process and subsequently be utilized by external entities, such as DSMgrs. Either flexibility providers or external entities may generate the necessary models and each of both options has its own benefits. When flexibility providers generate models, in general, less data needs to be transmitted and more accurate models may be generated, but more computational resources are needed by the flexibility provider. The models are generated from data, but detecting flexibility in historical data is a challenging task, which requires assumptions or additional information. Aside from observations, possible sources for information include fake data, already existing models, and standardized DER interfaces.

There are multiple options for the automated generation of models for demand side flexibility and even more different types of surrogate models, which can be used to generate control signals. Which approach is the most fitting one, depends on the specific use case, therefore only an overview of different approaches is provided here. Regardless of the approach, we point out aspects to consider when developing a surrogate-based approach for the exploitation of flexibility. Most importantly, surrogate modeling only provides approximations of the true, underlying flexibility. The magnitude and possible consequences of erroneous control signals should always be kept in mind. Utilizing statistical information, provided by the surrogate models, and integrating a feedback mechanism are two possible ways to deal with inaccurate models.

Overall, the automated generation of surrogate models for demand side flexibility could simplify the integration of DERs, as one of the most labor-intensive integration steps is automated. Moreover, the amount of exploitable demand side flexibility could thereby be increased, which in turn would facilitate the integration of RES into the energy system. There is a wide range of aspects to consider when developing such an approach.

ACKNOWLEDGMENTS

We gratefully acknowledge the financial support from the Federal Ministry for Economic Affairs and Energy (BMWi) for the project C/sells (funding no. 03SIN121). Furthermore, the present work was partially supported by the Helmholtz Association under the program "Energy System Design" in the Research Field Energy.

REFERENCES

[1] Abdul Afram, Farrokh Janabi-Sharifi, Alan S. Fung, and Kaamran Raahemifar. 2017. Artificial Neural Network (ANN) Based Model Predictive Control (MPC) and Optimization of HVAC Systems: A State of the Art Review and Case Study of a Residential HVAC System. *Energy and Buildings* 141 (April 2017), 96–113. https://doi.org/10.1016/j.enbuild.2017.02.012

[2] Ricardo J. Bessa, David Rua, Cláudia Abreu, Paulo Machado, José R. Andrade, Rui Pinto, Carla Gonçalves, and Marisa Reis. 2018. Data Economy for Prosumers in a Smart Grid Ecosystem. In *Proceedings of the Ninth International Conference on Future Energy Systems (E-Energy '18)*. Association for Computing Machinery, New York, NY, USA, 622–630. https://doi.org/10.1145/3208903.3210282

[3] Jörg Bremer. 2015. *Constraint-Handling mit Supportvektor-Dekodern in der verteilten Optimierung*. Ph. D. Dissertation. Carl von Ossietzky Universität Oldenburg, Ammerländer Heerstraße 114-118, 26129 Oldenburg.

[4] Joerg Bremer and Sebastian Lehnhoff. 2018. A Cascading Chi-Shapes Based Decoder for Constraint-Handling in Distributed Energy Management:. In *Proceedings of the 10th International Joint Conference on Computational Intelligence*. SCITEPRESS - Science and Technology Publications, Seville, Spain, 184–191. https://doi.org/10.5220/0006926101840191

[5] Joerg Bremer and Sebastian Lehnhoff. 2018. Unfolding Ensemble Training Sets for Improved Support Vector Decoders in Energy Management:. In *Proceedings of the 10th International Conference on Agents and Artificial Intelligence*. SCITEPRESS - Science and Technology Publications, Funchal, Madeira, Portugal, 322–329. https://doi.org/10.5220/0006543503220329

[6] Jörg Bremer, Barbara Rapp, and Michael Sonnenschein. 2010. Support Vector Based Encoding of Distributed Energy Resources' Feasible Load Spaces. In *2010 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT Europe)*. 1–8. https://doi.org/10.1109/ISGTEUROPE.2010.5638940

[7] Jörg Bremer, Barbara Rapp, and Michael Sonnenschein. 2011. Encoding Distributed Search Spaces for Virtual Power Plants. In *2011 IEEE Symposium on Computational Intelligence Applications In Smart Grid (CIASG)*. 1–8. https://doi.org/10.1109/CIASG.2011.5953329

[8] Jörg Bremer and Michael Sonnenschein. 2012. A Distributed Greedy Algorithm for Constraint-Based Scheduling of Energy Resources. In *2012 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 109–115.

[9] Aurélien Brès. 2018. *Automated Model Generation for Integrated Building and HVAC Performance Simulation*. Thesis. Wien.

[10] Indrasis Chakraborty, Sai Pushpak Nandanoori, and Soumya Kundu. 2018. Virtual Battery Parameter Identification Using Transfer Learning Based Stacked Autoencoder. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 1269–1274. https://doi.org/10.1109/ICMLA.2018.00206

[11] Carmen Exner, Marc-Aurel Frankenbach, Julia Koenig, and Nico Huebner. 2021. Grid Operation of the Future Using the Example of flexQgrid. In *ETG Congress 2021*. 1–5.

[12] Kevin Förderer, Mischa Ahrens, Kaibin Bao, Ingo Mauser, and Hartmut Schmeck. 2018. Modeling Flexibility Using Artificial Neural Networks. *Energy Informatics* 1, 1 (Oct. 2018), 21. https://doi.org/10.1186/s42162-018-0024-4

[13] Kevin Förderer, Mischa Ahrens, Kaibin Bao, Ingo Mauser, and Hartmut Schmeck. 2018. 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)*. Association for Computing Machinery, New York, NY, USA, 85–90. https://doi.org/10.1145/3208903.3208915

[14] Kevin Förderer, Manuel Lösch, Ralf Növer, Marilen Ronczka, and Hartmut Schmeck. 2019. Smart Meter Gateways: Options for a BSI-Compliant Integration of Energy Management Systems. *Applied Sciences* 9, 8 (Jan. 2019), 1634. https://doi.org/10.3390/app9081634

[15] Kevin Förderer and Hartmut Schmeck. 2019. State-Based Load Profile Generation for Modeling Energetic Flexibility. *Energy Informatics* 2, 1 (Sept. 2019), 18. https://doi.org/10.1186/s42162-019-0077-z

[16] Kevin Michael Förderer. 2021. *Modeling and Communicating Flexibility in Smart Grids Using Artificial Neural Networks as Surrogate Models*. Ph. D. Dissertation. Karlsruher Institut für Technologie (KIT). https://doi.org/10.5445/IR/1000136921

[17] Aurélie Foucquier, Sylvain Robert, Frédéric Suard, Louis Stéphan, and Arnaud Jay. 2013. State of the Art in Building Modelling and Energy Performances Prediction: A Review. *Renewable and Sustainable Energy Reviews* 23 (July 2013), 272–288. https://doi.org/10.1016/j.rser.2013.03.004

[18] Marcus Fuchs. 2018. *Graph framework for automated urban energy system modeling*. Ph. D. Dissertation. RWTH Aachen University, Aachen. https://doi.org/10.18154/RWTH-2018-224612

[19] Marcus Fuchs and Dirk Muller. 2018. Automated Design and Model Generation for a District Heating Network from OpenStreetMap Data. In *Proceedings of the 15th IBPSA Conference*. San Francisco, CA, USA, 2050–2059.

[20] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A Survey on Concept Drift Adaptation. *Comput. Surveys* 46, 4 (March 2014), 44:1–44:37. https://doi.org/10.1145/2523813

[21] C. U. I. Gaoying, Y. I. Yongxian, C. A. O. Xiaojun, and G. A. O. Ciwei. 2017. Dynamic Aggregation Model Considering Load Fluctuation and Its Participation in Economic Dispatch of Power System. *IOP Conference Series: Materials Science and Engineering* 199 (May 2017), 012075. https://doi.org/10.1088/1757-899X/199/

1/012075
[22] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.
[23] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Networks. *arXiv:1406.2661 [cs, stat]* (June 2014). arXiv:1406.2661 [cs, stat]
[24] Paul Grouchy. 2014. *Evolvable Mathematical Models: A New Artificial Intelligence Paradigm*. Ph. D. Dissertation. University of Toronto.
[25] Yusen He, Jiahao Deng, and Huajin Li. 2017. Short-Term Power Load Forecasting with Deep Belief Network and Copula Models. In *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Vol. 1. 191–194. https://doi.org/10.1109/IHMSC.2017.50
[26] Y. Huang. 2013. *Automated Simulation Model Generation*. Ph. D. Dissertation. Technische Universiteit Delft.
[27] Deqian Kong, Xiangyu Kong, Jie Xiao, Jian Zhang, Siwei Li, and Liang Yue. 2019. Dynamic Pricing of Demand Response Based on Elasticity Transfer and Reinforcement Learning. In *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*. 1–5. https://doi.org/10.1109/ICEMS.2019.8921683
[28] Anna Magdalena Kosek, Giuseppe Tommaso Costanzo, Henrik W. Bindner, and Oliver Gehrke. 2013. An Overview of Demand Side Management Control Schemes for Buildings in Smart Grids. In *2013 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*. 1–9. https://doi.org/10.1109/SEGE.2013.6707934
[29] Nils Kroener, Kevin Förderer, Manuel Lösch, and Hartmut Schmeck. 2020. State-of-the-Art Integration of Decentralized Energy Management Systems into the German Smart Meter Gateway Infrastructure. *Applied Sciences* 10, 11 (Jan. 2020), 3665. https://doi.org/10.3390/app10113665
[30] Georgios C. Kryonidis, Kyriaki-Nefeli D. Malamaki, Spyros I. Gkavanoudis, Konstantinos O. Oureilidis, Eleftherios O. Kontis, Juan Manuel Mauricio, José María Maza-Ortega, and Charis S. Demoulias. 2021. Distributed Reactive Power Control Scheme for the Voltage Regulation of Unbalanced LV Grids. *IEEE Transactions on Sustainable Energy* 12, 2 (April 2021), 1301–1310. https://doi.org/10.1109/TSTE.2020.3042855
[31] Nico Lehmann, Emil Kraft, Clemens Duepmeier, Ingo Mauser, Kevin Förderer, and Dominique Sauer. 2019. Definition von Flexibilität in einem zellulär geprägten Energiesystem. In *Zukünftige Stromnetze 2019 : 30./31. Januar 2019 in Berlin*. 459.
[32] Martin Lödl, Rolf Witzmann, and Michael Metzger. 2011. Operation Strategies of Energy Storages with Forecast Methods in Low-Voltage Grids with a High Degree of Decentralized Generation. In *2011 IEEE Electrical Power and Energy Conference*. 52–56. https://doi.org/10.1109/EPEC.2011.6070252
[33] Nicole Ludwig, Simon Waczowicz, Ralf Mikut, and Veit Hagenmeyer. 2017. Mining Flexibility Patterns in Energy Time - Series from Industrial Processes. *Proceedings. 27. Workshop Computational Intelligence, Dortmund, 23. - 24. November 2017. Hrsg.: F. Hoffmann* (2017), 13.
[34] Nicole Nadine Ludwig. 2020. Data-Driven Methods for Demand-Side Flexibility in Energy Systems. https://doi.org/10.5445/IR/1000123985
[35] Lukas Lundström, Jan Akander, and Jesús Zambrano. 2019. Development of a Space Heating Model Suitable for the Automated Model Generation of Existing Multifamily Buildings—A Case Study in Nordic Climate. *Energies* 12, 3 (01 2019), 485. https://doi.org/10.3390/en12030485
[36] Ingo Mauser. 2017. *Multi-modal Building Energy Management*. Ph. D. Dissertation. Karlsruher Institut für Technologie (KIT). https://doi.org/10.5445/IR/1000070625
[37] Ingo Mauser, Jan Mueller, Kevin Foerderer, and Hartmut Schmeck. 2017. Definition, Modeling, and Communication of Flexibility in Smart Buildings and Smart Grid. In *International ETG Congress 2017 (ETG-Fachberichte)*. VDE Verlag, 605–610.
[38] Clayton Miller, Zoltán Nagy, and Arno Schlueter. 2018. A Review of Unsupervised Statistical Learning and Visual Analytics Techniques Applied to Performance Analysis of Non-Residential Buildings. *Renewable and Sustainable Energy Reviews* 81 (Jan. 2018), 1365–1377. https://doi.org/10.1016/j.rser.2017.05.124
[39] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]* (Nov. 2014). arXiv:1411.1784 [cs, stat]
[40] Judith Neugebauer, Jörg Bremer, Christian Hinrichs, Oliver Kramer, and Michael Sonnenschein. 2016. Generalized Cascade Classification Model with Customized Transformation Based Ensembles. In *2016 International Joint Conference on Neural Networks (IJCNN)*. 4056–4063. https://doi.org/10.1109/IJCNN.2016.7727727
[41] Judith Neugebauer, Oliver Kramer, and Michael Sonnenschein. 2015. Classification Cascades of Overlapping Feature Ensembles for Energy Time Series Data. In *Data Analytics for Renewable Energy Integration (Lecture Notes in Computer Science)*, Wei Lee Woon, Zeyar Aung, and Stuart Madnick (Eds.). Springer International Publishing, Cham, 76–93. https://doi.org/10.1007/978-3-319-27430-0_6
[42] Judith Neugebauer, Oliver Kramer, and Michael Sonnenschein. 2017. Instance Selection and Outlier Generation to Improve the Cascade Classifier Precision. In *Agents and Artificial Intelligence (Lecture Notes in Computer Science)*, Jaap van den Herik and Joaquim Filipe (Eds.). Springer International Publishing, Cham, 151–170. https://doi.org/10.1007/978-3-319-53354-4_9
[43] Astrid Nieße, Michael Sonnenschein, Christian Hinrichs, and Jörg Bremer. 2016. Local Soft Constraints in Distributed Energy Scheduling. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*. 1517–1525.
[44] Anthony Papavasiliou. 2011. *Coupling Renewable Energy Supply with Deferrable Demand*. Ph. D. Dissertation. UC Berkeley.
[45] M. K. Petersen, K. Edlund, L. H. Hansen, J. Bendtsen, and J. Stoustrup. 2013. A Taxonomy for Modeling Flexibility and a Computationally Efficient Algorithm for Dispatch in Smart Grids. In *2013 American Control Conference*. 1150–1156. https://doi.org/10.1109/ACC.2013.6579991
[46] Alec Radford, Luke Metz, and Soumith Chintala. 2016. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]* (Jan. 2016). arXiv:1511.06434 [cs]
[47] Fredy Hernan Rios Silva. 2016. *Stigmergy-based Load Scheduling in a Demand Side Management Context*. Ph. D. Dissertation. Karlsruher Institut für Technologie (KIT). https://doi.org/10.5445/IR/1000055801
[48] Helen Sawall, Andreas Scheuriker, and Daniel Stetter. 2021. Flexibility Definition for Smart Grid Cells in a Decentralized Energy System. In *7th International Conference on Smart Cities and Green ICT Systems*. 130–139.
[49] Johan Simonsson, Khalid Tourkey Atta, Gerald Schweiger, and Wolfgang Birk. 2021. Experiences from City-Scale Simulation of Thermal Grids. *Resources* 10, 2 (Feb. 2021), 10. https://doi.org/10.3390/resources10020010
[50] Thomas Storek, Laurin Oberkirsch, Jonathan Kriwet, Marc Baranski, Thomas Schreiber, and Dirk Müller. 2020. Auto-Generation of Hybrid Automata for Real-Time Operation Optimization of Building Energy Systems. In *The 33rd International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems (ECOS 2020)*. Curran Associates, Inc, Osaka , Japan, 12.
[51] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second edition ed.). The MIT Press, Cambridge, Massachusetts.
[52] Anjan Kumar Swain and Alan S. Morris. 2003. An Evolutionary Approach to the Automatic Generation of Mathematical Models. *Applied Soft Computing* 3, 1 (July 2003), 1–21. https://doi.org/10.1016/S1568-4946(02)00074-1
[53] David M.J. Tax and Robert P.W. Duin. 2004. Support Vector Data Description. *Machine Learning* 54, 1 (Jan. 2004), 45–66. https://doi.org/10.1023/B:MACH.0000008084.60811.49
[54] Martin Ward. 1995. A Definition of Abstraction. *Journal of Software Maintenance: Research and Practice* 7, 6 (1995), 443–450. https://doi.org/10.1002/smr.4360070606
[55] Likun Xia, Muhammad Umer Farooq, Ian M. Bell, Fawnizu Azmadi Hussin, and Aamir Saeed Malik. 2013. Survey and Evaluation of Automated Model Generation Techniques for High Level Modeling and High Level Fault Modeling. *Journal of Electronic Testing* 29, 6 (Dec. 2013), 861–877. https://doi.org/10.1007/s10836-013-5401-0
[56] Lin Zhao, Wei Zhang, He Hao, and Karanjit Kalsi. 2017. A Geometric Approach to Aggregate Flexibility Modeling of Thermostatically Controlled Loads. *IEEE Transactions on Power Systems* 32, 6 (Nov. 2017), 4721–4731. https://doi.org/10.1109/TPWRS.2017.2674699