# Transfer Learning in Demand Response: a Review of Algorithms for Data-efficient Modelling and Control

Thijs Peirelinck[a,b,c,*], Hussain Kazmi[a], Brida V Mbuwir[b,c], Chris Hermans[b,c], Fred Spiessens[b,c], Johan Suykens[a], Geert Deconinck[a,c]

[a] *ESAT, KU Leuven, Kasteelpark Arenberg 10 bus 2445, 3001 Leuven, Belgium*
[b] *AMO, Flemish Institute for Technological Research (VITO), Boeretang 200, 2400 Mol, Belgium*
[c] *AMO, EnergyVille, Thor Park 8310, 3600 Genk, Belgium*

**Abstract**

A number of decarbonization scenarios for the energy sector are built on simultaneous electrification of energy demand, and decarbonization of electricity generation through renewable energy sources. However, increased electricity demand due to heat and transport electrification and the variability associated with renewables have the potential to disrupt stable electric grid operation. To address these issues using demand response, researchers and practitioners have increasingly turned towards automated decision support tools which utilize machine learning and optimization algorithms. However, when applied naively, these algorithms suffer from high sample complexity, which means that it is often impractical to fit sufficiently complex models because of a lack of observed data. Recent advances have shown that techniques such as transfer learning can address this problem and improve their performance considerably - both in supervised and reinforcement learning contexts. Such formulations allow models to leverage existing domain knowledge and human expertise in addition to sparse observational data. More formally, transfer learning embodies all techniques where one aims to increase (learning) performance in a target domain or task, by using knowledge gained in a source domain or task. This paper provides a detailed overview of state-of-the-art techniques on applying transfer learning in demand response, showing improvements that can exceed 30% in a variety of tasks. We observe that most research to date has focused on transfer learning in the context of electricity demand prediction, although reinforcement learning based controllers have also seen increasing attention. However, a number of limitations remain in these studies, including a lack of benchmarks, systematic performance improvement tracking, and consensus on techniques that can help avoid negative transfer.

## 1. Introduction

### *1.1. Background*

A number of decarbonization scenarios for the energy sector are built on simultaneous electrification of heating and transport, and decarbonization of electricity generation through

*Corresponding author

*Preprint submitted to Elsevier*

renewable energy sources. The electrification of heating and transport significantly increases electricity demand, and the variability in generation is expected to increase sharply with more renewables in the grid. Taken together, these have the potential to disrupt stable electric grid operation [1]. These problems appear on multiple levels. For customers, this often means increasing energy costs, while also leading to voltage and power flow issues on the distribution grid [2]. On the transmission side, it can lead to frequency issues caused by inertia loss and steep ramp rates [3], as well as an increase in capacity requirements [4]. The latter is particularly dangerous as the contracted capacity is often in the form of polluting peaking plants, which detracts from the decarbonization objectives [5].

To address these issues, researchers and practitioners have increasingly turned towards decision support tools that utilize machine learning algorithms [6]. Machine learning algorithms allow stakeholders to accurately model [7] and predict energy demand [8], generation [9] and prices [10]. These predictions can then be used to optimally schedule electricity demand and generation to minimize grid issues, costs and carbon footprint. They can also be used to improve user engagement and comfort, automatically detect and predict operational faults [11] and inform policy choices [12]. Electricity demand scheduling or optimization for grid decision support is more commonly referred to as Demand Response (DR). Automation, relying on machine learning algorithms, is vital for DR, as the alternative of solely relying on human domain experts to construct grid decision support tools is both impractical and economically infeasible [13]. This is true especially in the area of residential user engagement, more commonly known as residential DR, where machine learning will play a vital role to ensure scalability [14]. In a more broader context, Reinforcement Learning (RL) is expected to be of increasing importance in distribution grid decision support, as is illustrated by a special set of RL environments, that should facilitate further research, introduced by Henry and Ernst [15].

However, machine learning algorithms and decision support tools, such as reinforcement learning, when applied naively, suffer from a number of issues. Foremost among these is their high sample complexity, which means that it is often impractical to fit sufficiently complex models in energy systems due to lack of data [16]. This hampers their ability to generalize well in novel conditions, and manifests along all three dimensions of interest in a learning system, i.e. they tend to show poor initial performance [17], improve only slowly with observation data, and show asymptotic performance, which is demonstrably sub-optimal [18]. One real-world example of this is the widespread use of local forecasting models for (peak [19]) energy demand [20], generation and prices [21]. These models rely primarily on on-site observation data to make forecasts [22]. However, due to limited data availability and the curse of dimensionality, they invariably fail to capture both the low frequency seasonality and the possibility of extreme events. The same problems arise in applying reinforcement learning for active control. Often, these algorithms fail to converge to the optimal policy simply because of how little interaction data is available in practice and the naive manner in which the agents are formulated [18, 23, 24, 25, 26]. These are not just theoretical problems; rather they are a critical roadblock in applying machine learning to energy systems in practice.

Recent advances have shown that techniques such as transfer and semi-supervised learning can improve the performance of machine learning models considerably - both in supervised and reinforcement learning contexts [27]. Such formulations allow models to leverage existing data, domain knowledge and human expertise [28, 29]. The biggest advantage of transfer learning is that it reduces the data complexity of machine learning models [30]. More specifically, by leveraging domain knowledge and/or previously gathered data, machine learning models tend to perform better with fewer data points, learn faster as more data becomes available [17], and achieve higher asymptotic performance than their naive counterparts [31].

Due to these benefits, transfer learning can enable large scale real-world roll-out of automated DR programs. This ranges from improved forecast and dynamics models to more efficient reinforcement learning agents. In this paper, we present a thorough review of how these techniques have been applied in practice to date for DR, in both supervised and reinforcement learning settings. For supervised learning, we focus on the three key variables in energy systems: demand, generation and prices. For reinforcement learning, we focus on how transfer learning can be used to directly improve the operational control of energy flexible resources. We also take a closer look at some future directions based on research in other domains, as well as how to address the unique challenges that arise in energy systems modelling and control when applying transfer learning.

### *1.2. Previous Reviews*

A number of recent reviews address machine learning and DR [32, 33], as well as how reinforcement learning relates to it [34]. However, the focus in these reviews is typically on general techniques, and not specifically on how to use transfer learning to operationalize them in practice. On the other hand, a number of transfer learning surveys have been presented in recent years, both on general transfer learning [35, 36] and on transfer within the reinforcement learning setting. None of these focus on applications within the smart grid or DR setting.

It is important to make the distinction between transfer learning and the broader field of informed Machine Learning (ML). Informed machine learning covers a broader range of possibilities to inform a ML agent. This can be through adding differential equations to the loss-function, simulation results, knowledge graphs, etc [37]. Furthermore, informed ML also incorporates what is referred to as physics informed ML [38]. This branch of ML aims to incorporate physical knowledge into the learning pipeline. This can, for example, be achieved by the introduction of physics-constrained convolutional encoder-decoder networks [39]. Recent literature has even showed data free learning of parametric partial differential equations by physics-informed convolutional neural networks [40]. Transfer learning, on the other hand, focuses specifically on methods that inform a ML agent through transfer, mostly of data, model parameters or feature representations. In transfer learning, one thus typically makes a distinction between the source domain (or task), *i.e.*, the source of the initial knowledge and the target domain (or task), *i.e.*, where the knowledge is used to improve performance. Thus, while informed machine learning consists of all genres to inform a machine learning agent, transfer learning is its subset, only focusing on those methods that inform ML agents by transferring data or model parameters from a source domain/task to

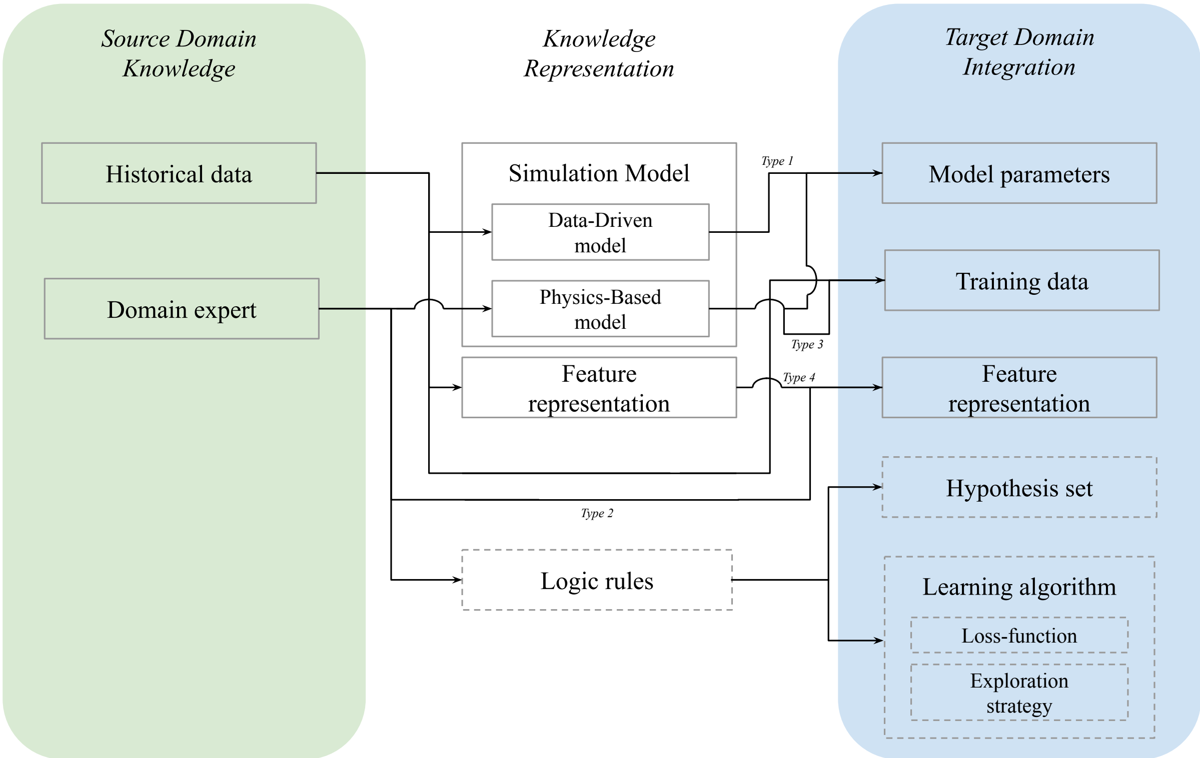


Figure 1: Knowledge representation and integration in informed machine learning. The knowledge is transferred from left to right and 4 archetypes of transfer learning have been identified, which have been explained in Section 2. Full lines: research in demand response exist. Dashed lines: open research questions.

a target domain/task. Very recently, Von Rueden *et al.* [37] have presented an extensive literature review on informed machine learning. However, this literature review is general in its scope, and does not address the smart grid or DR case. In a subsequent section, we make the link between transfer and informed ML more explicit.

Fig. 1 is a (non-exhaustive) visualization of how knowledge can be represented and integrated in a ML pipeline. The full lines indicate the main focus of transfer learning, and hence this paper. The dotted squares give a few examples how informed ML is broader than transfer learning.

As different surveys have adopted different nomenclature, the nomenclature as used in this paper is defined in Table 1.

### *1.3. Organisation*

The following section discusses the other transfer learning reviews more in depth and formalises a taxonomy based on the adopted definitions. The two sections thereafter present how the different DR settings fit into the taxonomy. Section 3 focuses on transfer learning in RL, while Section 4 focuses on transfer learning in supervised learning contexts. Section 5 presents different transfer learning applications within the DR setting. Section 6 identifies future research opportunities. The final section concludes this review.

Table 1: Terminology adopted in the paper

| | Source domain | Target domain |
|---|---|---|
| State-space | $\mathcal{X}_S$ | $\mathcal{X}_T$ |
| Action-space | $\mathcal{U}_S$ | $\mathcal{U}_T$ |
| Policy | $\pi_S(x_S)$ | $\pi_T(x_T)$ |
| Domain data | $\mathcal{D}_S$ | $\mathcal{D}_T$ |
| Reward function | $r_S(x, u, x)$ | $r_T(x, u, x)$ |
| Task | $\mathcal{T}_S$ | $\mathcal{T}_T$ |
| Input feature space | $X_S$ | $X_T$ |
| Target space | $Y_S$ | $Y_T$ |

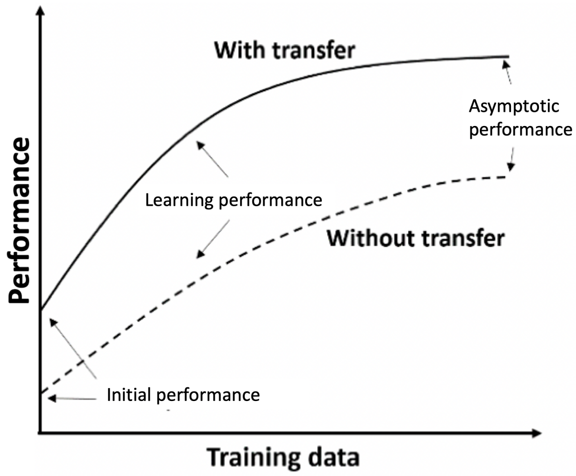


Figure 2: Rationale for transfer learning in supervised and reinforcement contexts.

## 2. Taxonomy of Transfer Learning Algorithms

### *2.1. Conventional Machine Learning*

In conventional machine learning, a functional mapping is learnt between input and output variables for a specific problem using a well-defined dataset. The only expert or domain knowledge in this case is typically in the learning pipeline, i.e. in the choice of learning algorithm or feature selection. Expert knowledge can therefore enable the practitioner to identify the correct learning algorithm or set its hyperparameters accurately. Likewise, expert knowledge is useful in feature engineering or hand-crafting features, which are amenable to the learning process. In all this, there is no notion of domain knowledge being used explicitly in the machine learning process, neither is there any transfer of knowledge across subsequent models that may be built.

### *2.2. Transfer Learning*

Informally, transfer learning can be defined as the process of extracting information from a source domain and task and using this information to improve in a target domain and task [41]. Both Pan *et al.* [41] and Weiss *et al.* [35] formally define transfer learning as the process of improving the target predictive function $f_T(\cdot)$, given target domain $\mathcal{D}_T$ and target task $\mathcal{T}_T$, by using the information of a given source domain $\mathcal{D}_S$ with corresponding source task $\mathcal{T}_S$.

Fig. 1 shows how transfer learning differs from conventional machine learning. In contrast to a single learning pipeline, with a well-defined dataset, there is no knowledge flow from a source domain/task to a target domain/task. This allows for two learning pipelines in the overall problem. For example, in *type 1* first a model is learned using available historic data. Thereafter, this model can be used in the target domain. Furthermore, apart from domain knowledge in the learning pipeline, there thus arises the possibility to add domain knowledge about the learning problem itself, this is identified as *type 2* in Fig. 1. A similar approach is used in *type 3*. However, here, a domain expert constructs a physics-based model that can be used to guide the learner in the target domain. *Type 4* considers these transfer learning applications that extract a feature representation from the source to initialise the target feature representation.

Fig. 2 illustrates how the initial and asymptotic performance of conventional machine learning models can be improved with transfer learning algorithms, which incorporate domain knowledge or human expertise into machine learning models, both in the supervised and reinforcement learning contexts. It should be noted that neither initial nor asymptotic performance increase is guaranteed.

Based on the above definition, and the applications of transfer learning, Pan *et al.* [41] subdivide transfer learning in three categories; inductive, transductive and unsupervised transfer learning. These categories can be further subdivided based on the ML setting they are applied in. DR control applications mainly benefit from ML in the supervised or RL setting. However, Pan *et al.* [41] explicitly mention that they do not consider transfer in reinforcement learning. Therefore, and because RL differs considerably from both supervised learning and unsupervised learning [42], we believe transfer in reinforcement learning and supervised learning should be dealt with separately. To date, the unsupervised transfer learning setting has only seen limited applications within the smart grid setting, and especially the DR setting. Consequently, a thorough discussion of this transfer learning setting has been considered outside the scope of this review.

The next subsection begins with a description of the different methods to evaluate transfer learning and the differences with evaluating other ML approaches. Thereafter, the different categories of transfer learning: transductive and inductive transfer learning are described. This is important to understand how and when what type of transfer learning can be applied in practice. The fifth and sixth subsections describe transfer learning in the context of reinforcement and supervised learning, respectively. Throughout this discussion the DR control use cases have always been kept in mind.

### *2.3. Evaluating Transfer Learning*

Reinforcement learning agents are evaluated on how much reward they accrue over time. Likewise, supervised learning algorithms are evaluated on a predefined loss function. When these supervised learning algorithms are applied in an online fashion, i.e. where the model parameters are updated over time with newly observed data, the evolution of this loss function over time is also an important metric to consider. Therefore, in practice, it is not sufficient to consider how a supervised learning model or reinforcement learning agent is performing, rather it is more relevant to track its performance over time as it gains access to increasing amounts of data. These metrics can be summarized by the initial performance, learning performance and asymptotic performance, as shown in Fig. 2 for supervised learning. It is straightforward to extend these metrics to the case of reinforcement learning. It is also important to note that these three metrics are agnostic to the defined performance metric i.e. reward or loss function.

### *2.4. Categories of Transfer Learning*

There are slight variations on how different authors have categorized transfer learning methods in the past. These different types of categories arise from using either the feature space or the task and domains as separators. For instance, Pan *et al.* [41] use differences in task ($\mathcal{T}$) and domain ($\mathcal{D}$) to subdivide transfer learning methods. In a RL setting, the task is defined by the reward function. In contrast to Pan *et al.*, Weiss *et al.* [35] subdivide transfer learning methods based on the feature space. In a RL setting, the feature space is defined by the state-space $\mathcal{X}$. Consequently, Weiss *et al.* [35] use two categories: homogeneous and heterogeneous transfer learning. In homogeneous transfer learning, the source and target feature spaces are the same, *i.e.* $\mathcal{X}_S = \mathcal{X}_T$. On the other hand, in heterogeneous transfer learning, source and target domains are represented in different feature spaces.

Contrary to a focus on solution methods, Pan *et al.* [41] adopt a focus on the field of transfer learning or, rather, the transfer learning problems. Naturally, this results in a subdivision based on the task, *i.e.* reward function, and the domain $\mathcal{D}$. Taylor *et al.* [36], whose survey of transfer learning focuses on RL, in some sense also use this division. We have adopted this as well. In transductive transfer learning the source and target task are the same. However, the domain differs. Both types of transfer learning taxonomy, and their interactions, have been visualised in Fig. 3. An example of transductive transfer learning within the DR setting could be optimising local photovoltaic (PV) self-consumption with a battery in the source domain and another device, such as an Electric Water Heater (EWH), in the target domain. Intuitively, one can see that transfer learning can be of use in such a scenario, as the control policy will be fairly similar in source and target domain. Potentially, transfer learning could thus provide a *jump start* in the target domain.

Transductive transfer learning loans its name from transduction, or transductive learning, as introduced by Vapnik [43, 44]. They are related in the sense that in the transductive transfer learning setting, like in transductive learning, there is no interest in building a general model that can be transferred (as $\mathcal{D}_S \neq \mathcal{D}_T$) [45, 43], *i.e.*, there is no interest in a general model for transferring all future new tasks. Rather, interest lies in knowledge transfer for this specific task. Thus, while Vapnik introduced transductive learning on the level of a single data-point, here transductive learning is concerned with tasks and domains.

On the other hand, inductive transfer learning contains transfer learning problems with a different task, within the same domain. Going back to the above mentioned battery control use case, inductive transfer learning would be to switch using the battery's energy buffer for

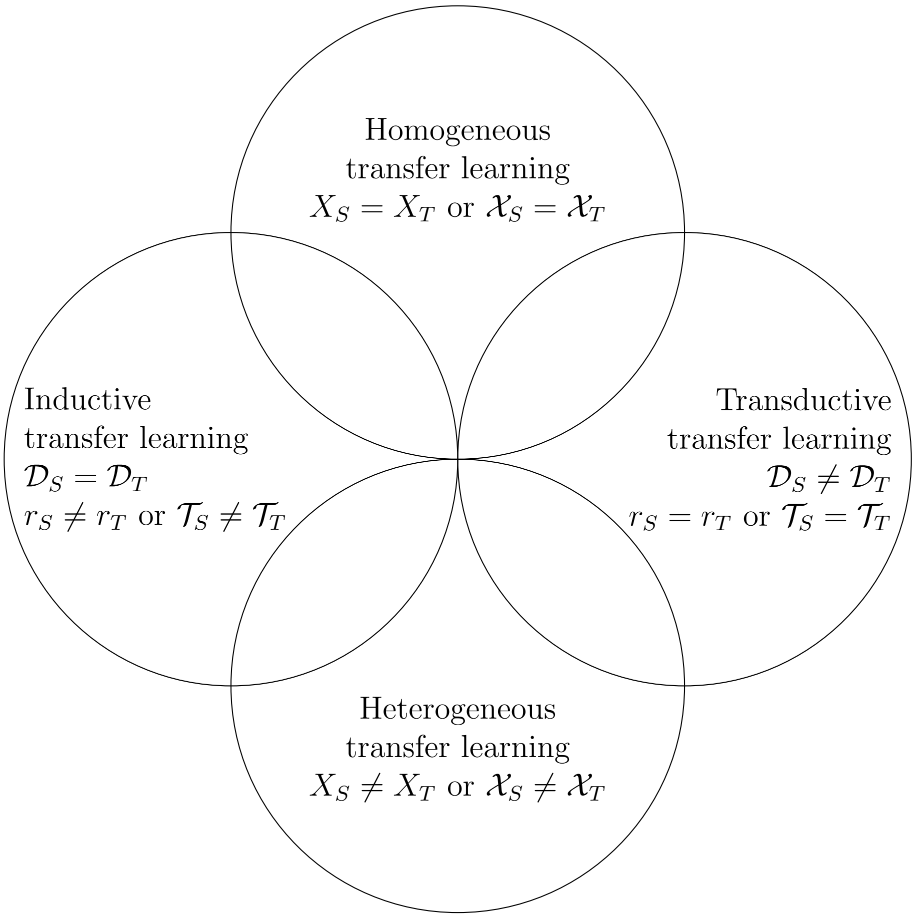


Figure 3: Venn diagram of different transfer learning taxonomies.

a self-consumption goal to an energy arbitrage goal. The dynamics of the battery stay the same in both cases, and, therefore, relevant knowledge about the control problem can be transferred between these two use cases. A more detailed treatment of how different transfer learning techniques are applied in practice is deferred to Section 6.

## 3. Transfer in Reinforcement Learning

There does not exist one single method of utilising transfer learning for RL, and different parts of a typical RL pipeline can benefit from it. This subsection first gives a short introduction to RL and the terminology used. Thereafter, the different possibilities of transfer learning within RL are explored, with a focus on DR applications. Fig. 4 summarizes the different categories of utilizing transfer learning within RL.

| *Transfer in Reinforcement Learning* | |
|---|---|
| Transductive Transfer | Inductive Transfer |
| $\mathcal{D}_S \neq \mathcal{D}_T$ | $\mathcal{D}_S = \mathcal{D}_T$ |
| $r_S = r_T$ | $r_S \neq r_T$ |
| Transfer in Model-Based Reinforcement Learning | |

Figure 4: Summary of Section 3

### *3.1. Background*

RL uses the Markov Decision Process (MDP) mathematical framework to formalize the decision making process. A MDP models the interaction between a decision making agent and its environment. The agent interacts with the environment at discrete time-steps $t$. At every such time-step, the agent perceives a state $x \in \mathcal{X}$ of the environment. Based on this state, the agent has to decide upon an action $u$ out of the set of all possible actions $\mathcal{U}$. In the subsequent time-step, the agent receives a reward. The value of this reward is determined by a reward function $r(x_t, u_t, x_{t+1})$, and thus partly influenced by the agent's behaviour. The dynamics of a finite MDP, with optimization horizon $T \in \mathbb{N}\backslash\{0\}$, are defined by the transition function of the environment, given by (1) with $\nu_k$ process noise

$$x_{k+1} = f(x_k, u_k, \nu_k) \quad \forall k \in \{0, \dots, T-1\}. \tag{1}$$

The conditional probability of perceiving state $x_{t+1}$ and reward $r$, given particular values of the preceding state $x_t$ and the chosen action $u_t$, is given by $P(x_{t+1}, r_{t+1}|x_t, u_t)$. The goal of the agent is then to find a policy $\pi : \mathcal{X} \rightarrow \mathcal{U}$, which maximizes the expected discounted reward, given a certain discount factor $0 \leq \gamma < 1$. The value of state $x$ under policy $\pi$ is defined by the value function (2)

$$V_\pi(x) = \mathbb{E}_\pi \left[ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | x_t = x \right], \quad \forall x \in \mathcal{X}. \tag{2}$$

Similarly, equation (3) defines the state-action value function $Q_\pi(x,u)$ under the policy $\pi$, *i.e.* the value of taking action $u$ while in state $x$ and following policy $\pi$ thereafter

$$Q_\pi(x,u) = \mathbb{E}\left[\sum_{k=1}^{\infty}\gamma^k r_{t+k+1}|x_t = x, u_t = u\right], \quad \forall x \in \mathcal{X}, u \in \mathcal{U}. \tag{3}$$

The goal of any RL agent is then to find a policy that maximizes $V(x)$ for every state $x$.

State-of-the-art RL algorithms introduce function approximation in one or several parts of the MDP. As a first example consider a value iteration algorithm, such as Q-learning. In Q-learning one aims to iteratively update $Q(x,u)$ in order to find the optimal Q-function, and as a result the optimal policy. With a finite state-space, Q can be represented in tabular form. However, with a large state-space, and in particular with an infinite one, this becomes infeasible. Therefore, Mnih *et al.* [46] proposed to use a Neural Network (NN) to approximate the Q-function. After Mnih *et al.* showed the benefits of using function approximation within RL, it has been widely used by researchers in other parts and/or other RL algorithms. A second part where function approximation might be useful, is in policy iteration algorithms. These algorithms aim to estimate the policy directly, rather than through the (state-action) value function, as in Q-learning. As a consequence, function approximation can be used to represent $\pi(x)$. An example of such an algorithm has been proposed by Schulman *et al.* [47]. Finally, in model-based RL the transition function itself is approximated [48]. In almost every instance of function approximation, it is possible to use transfer learning. The following subsections discuss transfer learning for each of the three parts of an RL algorithm discussed previously.

Vazquez *et al.* [34] show in their survey of RL in DR, that almost all applications use some form of Q-learning. Hence, our focus on Q-learning. However, many of the algorithms subsequent to deep Q-learning [46], have been introduced in an attempt to improve upon the sample efficiency of RLs algorithms [47]. More recently, researchers have been looking at transfer learning to increase sample efficiency in the target domain [31]. Furthermore, in some critical applications, exploration, which is inherent to RL should be avoided. In such situations, transfer learning can be used to jump-start the agent's performance [31]. While exploration might not necessarily lead to critical errors in the DR case, users can benefit from reduced amounts of exploration in the start-phase of deployment [29].

### *3.2. Transductive Transfer: source and target domain differ*

Transductive transfer learning, as defined earlier and by Pan *et al.* [41], encompasses these scenarios of transfer learning where source and target domains differ, but source and target task coincide. In RL, transductive transfer learning thus refers to these learning problems where source and target environment, and thus transition function $f$, are different. But, the reward function remains the same after transfer.

The idea of transferring RL agents between environments became widespread in the ML research community when openAI launched its retro contest in 2018 [49]. This contest aimed to accelerate (transductive) transfer learning by providing researchers with benchmark computer games. Algorithms submitted were tested on a new set of (unseen) levels in their respective games. Good performing RL agents should thus be able to generalise to unseen levels of the same game, *i.e.* the domain is different while the reward is the same.

In a smart grid, and more specifically in a DR context, there is vast potential for transductive transfer learning. Different parts of the environment can alter the underlying transition function, while the general principles of the problem can remain the same. In DR programs there are three knowledge sources that can be used for transfer: (other) real world data, domain knowledge - including simulations, and shared domain features. All three call for different transfer strategies, which we discuss next.

1. **Real world data.** Transfer from real world data includes transfer from earlier work, but also transfer from other related data sources with the same reward-function. As a first example, consider the work of Paridari *et al.* [50] and Mbuwir *et al.* [51]. In their work, they aim to design a plug-and-play Home Energy Management System (HEMS) for a PV-battery system. However, they realise user behaviour can result in differences in the transition function and, therefore, optimal policy. To mitigate this challenge, they cluster different households and use transfer learning between households that end up in the same cluster. A new household might lack enough historical data to design a tailor made HEMS, but by using limited data it is possible to pick the right cluster and use its policy as an initial starting point.
2. **Domain knowledge.** DR potential arises when there is a form of energy flexibility available, such as a battery in the previous example. Thermostatically Controlled Loads (TCLs) provide another source of energy storage and their potential for DR has been proven numerous times in recent literature [18, 16, 52, 53, 54, 55, 17, 56, 57]. Although RL does not strictly need a model of the control environment, it certainly can benefit from one. Domain knowledge, for example in the form of a dynamic model of the environment, has been used to mitigate low data efficiency of certain RL algorithms. Lampe *et al.* [28] introduced Model-Assisted Fitted Q-Iteration (MAFQI). MAFQI is a variation of the Q-learning algorithm, in which virtual trajectories, originating from a learned environment model, are added to the RL agent's training set used to update the Q-function. Their results show an improved data efficiency, compared to regular Q-learning. Costanzo *et al.* [29] show these results can also be obtained in a DR application. Consequently, Patyn *et al.* [58] have expanded this idea to obtain informed Fitted Q-Iteration (FQI). In their approach, model-free FQI is provided with domain knowledge through the use of models constructed by domain experts.
   In the previous examples, models of the environment have been used to provide the RL agent with an increased amount of state-transitions in the start-up phase. Further research, mainly in the domain of robotics, has shown simulations can also be used to explicitly initialise the policy $\pi(x)$ [31]. Peng *et al.* [31] observe that there will always be discrepancies between source, here simulated, and target domain. With domain randomisation, these discrepancies are modelled as variability in the source domain. Peirelinck *et al.* [17] have shown that domain randomisation successfully provides an RL agent with a jump-start.
3. **Shared domain features.** A third opportunity for transfer learning lies in the nature

of energy flexibility options. This is because energy flexibility can be provided based on different technologies, yet the main principles largely stay the same. In a lot of applications, energy flexibility is provided by some form of energy storage, *e.g.* heat or chemical storage. While it is clear that the transition function of these types of storage is different, at a high enough abstraction layer their functionality remains the same. It remains to be seen if, for example, RL agents trained on battery storage applications have policies that are general enough to be used as initial policies in an application with TCLs. But, when differences between domains are minor, for example different rooms in the same building, sharing features can be a successful approach. Kim *et al.* [59] demonstrate this in their multi-task learning setting, by sharing features (and transitions) between control policies for different rooms in a building.

### *3.3. Inductive Transfer: source and target task differ*

Recall that, by definition [41], inductive transfer learning is the case where the domains are the same, but tasks differ. This is visually presented in Fig. 3. In a RL scenario, this means both tasks share transition function $f$, but have a different reward function $r$. Note however that, although the transition function is the same, the conditional probability distribution $P(x_{t+1}, r_{t+1}|x_t, u_t)$ in the two domains can differ, as these probabilities are policy dependent (and the policy depends on the reward function). Since domains are the same, methods used for inductive transfer learning often correlate with those used for multi-task learning [41].

In a supervised learning setting this implies labels have to be available in both the source and target domains. Or, it needs to be possible to induce them [41]. In a RL scenario, the reward function should be available in both domains. The different approaches for inductive transfer in RL can be divided in similar categories as the above mentioned transductive learning scenarios.

A first intuitive approach is to transfer knowledge of instances, *i.e.*, use source domain instances to accelerate and jump-start target domain performance [41]. As domains are the same, it is not strictly necessary to use schemes such as domain randomization to account for domain difference. One approach is to weigh source and target domain samples differently in order to prioritise target domain experience [60]. The energy arbitrage application, as presented by Ruelens *et al.* [18], can be considered an example of inductive transfer with RL. Day-ahead electricity prices are changing from day to day and, thus, rewards of previously seen state transitions are not representative for future rewards. Even if the exact same transition would occur in the future, the received reward (or cost) would (likely) be different, since the electricity price will be different. Therefore, Ruelens *et al.* recalculate all rewards for the new prices occurring the next day. They thus use samples of the past (in the same domain) to train for a new task (new day-ahead electricity prices).

Furthermore, knowledge can be transferred using feature representations [41]. If the possible tasks, *i.e.*, reward functions, are known in advance, the RL agent can be trained on the different tasks together [61]. When using a NN for function approximation, this allows to share learned feature representations.

In a similar fashion, knowledge can be transferred using the knowledge incorporated in the parameters of the RL agent [41], be it the hyperparameters or the parameters of the regressor used to represent the policy or value-function. In this type of setting, model-based RL comes to mind. As the domain is the same, but the tasks differ, it is possible to learn an approximate model of the environment, which can then be used in an optimization setting for multiple objective functions.

### *3.4. Transfer Learning for Model-Based RL Algorithms*

It is clear RL plays a vital role in recent DR control applications. The main benefit of RL is the lack for the need of an environment model, and, therefore, domain knowledge. In all examples presented until now, this was achieved by directly learning a (state-action) value-function or a control policy. It is, however, also possible to eliminate the need for a domain expert by *learning the model* of the environment, using the transitions the agent experienced. With model-based RL, there is still no need for a domain expert, as the model is learned using a data-driven approach. The dynamic's model of the environment is learned (mostly) on-line, while control is active. This predictive model can then be used to estimate the cost or reward of a certain action, when in a certain state [48].

While model-based RL has proven to be relatively sample-efficient, compared to model-free RL, there is still room for improvement [48, 62]. Similar to model-free RL, model-based RL can use transitions of a source domain to jump-start control performance in the target domain [62]. Taylor *et al.* [62] have developed a model-based transfer learning method where source domain transitions are transformed to fit the target domain and task. This transformed source-data can then be used to build an initial model in the target domain.

Recent literature has shown that transfer learning can be used to mitigate the lack of sensing in EWHs [16]. With transfer learning Kazmi *et al.* accomplish few-shot learning, both with homogeneous and heterogeneous appliances. It thus enables all benefits of black-box modeling, while limiting the need for extensive sensing. It is exactly in this regard that the aim of transfer learning for model-based RL differs from general supervised learning. In model-based RL, the model is needed for control of a certain environment, and one mostly operates in a few-shot learning setting. Sample efficiency is therefore very important.

## 4. Transfer in Supervised Learning Settings

A similar review and analysis can be performed for transfer learning in the supervised learning context. This section starts with a short introduction to supervised learning. Thereafter, the transductive and inductive transfer learning contexts are explored. Fig. 5 summarizes the different transfer learning concepts in a supervised learning setting.

### *4.1. Background*

Transfer in a supervised context in DR is primarily used to improve models that can explain and predict different factors, ranging from user, device and grid behaviour. These models are helpful in creating future trajectories of device, user or grid behaviour, which can be used as input to control algorithms such as model predictive control and model-based RL

| *Transfer in Supervised Learning* | |
|---|---|
| Transductive Transfer | Inductive Transfer |
| $\mathcal{D}_S \neq \mathcal{D}_T$ | $\mathcal{D}_S = \mathcal{D}_T$ |
| $\mathcal{T}_S = \mathcal{T}_T$ | $\mathcal{T}_S \neq \mathcal{T}_T$ |
| Related Concepts in Time Series Analysis | |

Figure 5: Summary of Section 4

to achieve predefined objectives such as reducing electricity costs, emissions or grid impact etc. [63]. User models can also be utilized to engage and to inform energy consumers of modifications to their energy behaviour that may lead to savings in costs or emissions etc. Likewise, device models can also be used to estimate existing energy flexibility potential as well as in early fault detection. Grid models can allow system operators to better understand existing hosting capacity for distributed energy resources, and plan their network capacity expansions accordingly.

It is important to note the many different ways to characterize these DR related models. The most notable characterization is in terms of the data used to build them and how interpretable they are to a domain expert; a classification referred to as white-box, grey-box and black-box models [64]. While white-box models typically rely exclusively on domain expertise, black-box models are often purely data-driven i.e. based on observation data. Grey-box models straddle the line between the two by combining domain knowledge with machine learning, and can offer improved performance and interpretability. In many ways, these can often be thought of as a sub-class of informed machine learning as well.

More concretely, in a data-driven black-box context, given a set of $N$ training examples of the form $\{(\mathtt{x}_1, \mathtt{y}_1), (\mathtt{x}_2, \mathtt{y}_2), ..., (\mathtt{x}_n, \mathtt{y}_n)\}$, where $\mathtt{x}_i \in X$ is the input feature vector of the $i^{th}$ example and $\mathtt{y}_i \in Y$ is the target variable, the supervised learning algorithm attempts to learn a function $f : X \rightarrow Y$, which maps $X$, the input feature space, to $Y$, the target space. This function can subsequently be used to generate predictions given any arbitrary set of input features. In DR, where regression problems are quite common, this function typically minimizes an error metric such as mean absolute error or mean squared error on the training and validation dataset. Likewise, other cost functions such as the pinball loss are used as well, e.g. in the creation of quantile regression models.

As opposed to the naive formulation of a machine learning model, transfer learning reuses existing data or knowledge. However, as in the case of reinforcement learning, a number of distinctions can be applied on how to approach transfer in supervised contexts. Given the definitions given above, these include the case where (1) the feature space is different between the source and target domain, i.e. $X_S \neq X_T$; (2) the marginal probability distribution differs between the source and target domain, i.e. $P(X_s) \neq P(X_t)$; (3) the label space differs across the source and target domain, i.e. $Y_S \neq Y_T$; and (4) the conditional probability distribution varies between the source and target task, i.e. $P(Y_s|X_s) \neq P(Y_t|X_t)$.

Here, the second case is analogous to transductive transfer and the fourth case to inductive transfer as described earlier in this paper. It is important to note that it is possible to combine these two conditions to still achieve transfer, i.e. where both the marginal and conditional probabilities differ between source and target. We take a look at the different ways this can be, and has been, realized in practice in the supervised learning context next.

### *4.2. Transductive Transfer: source and target domain differ*

This refers to the case where the source and target domain differ, but the source and target task coincide. An example of this is when a model needs to be learned for identical, or very similar, devices which are being operated in different operating conditions. This model is expected to generalize from the given training dataset, and learn a mapping from input feature space to target feature space. In practice, this can be done in a number of ways which mirror transductive transfer in reinforcement learning settings.

1. **Real world data.** Transfer using (models trained on) similar datasets can often lead to improved modelling performance. This is especially true now that a large number of energy related datasets have been open-sourced [65].
2. **Domain knowledge.** Transfer can also be achieved by making use of domain knowledge. This is valid only in cases where the process can be described succinctly using physical equations, logic rules or through simulation models.
3. **Shared feature representations.** Transfer can also be effectively realized when, rather than transferring examples or model parameters, knowledge of how to transform observational data into a more usable form (i.e. one that is more amenable to learning) is transferred. This is arguably the automated analog of the historically human-intensive operation of feature engineering.

The different information sources, arising from real world data and domain knowledge, can be incorporated in the supervised learning algorithm through the use of (1) a modified loss function while learning the input-output mapping [37], and (2) simulation tools, which include techniques such as domain randomization and data augmentation [66], besides others. In either case, when data points gathered from previous similar projects or those generated from domain knowledge based simulations are used in the learning process directly, a weighting factor depending on the 'dissimilarity' between the source and target is applied. The training examples then take the form $\{(\mathtt{x}_1, \mathtt{y}_1, w_1), (\mathtt{x}_2, \mathtt{y}_2, w_2), ...,$ $(\mathtt{x}_n, \mathtt{y}_n, w_n)\}$, where $w_i$ is the weight assigned to each individual training example. Alternatively, it is increasingly common to use a two step approach, whereby the 'transferred' examples are used to first pre-train an initial model. The observed examples are then used to fine-tune the parameters of this model. This latter also has a close link to sharing feature representations across source and target.

The effect of these strategies can be seen as fundamentally that of regularization, in which domain knowledge or a greater amount of data helps with constraining the output of the mapping function.

### *4.3. Inductive Transfer: source and target task differ*

Unlike in the reinforcement learning case, the task for most supervised learning algorithms is to create a model that explains and predicts the behaviour of a device, a building or its occupants, or energy grids and markets in a DR context. Therefore, the inductive transfer case mostly refers to when data collected from heterogeneous sources is used in an attempt to accelerate learning. This is the case, for instance, when data collected from a heat pump or solar panel is used as an example for a new instance of the device, which has a different behaviour (i.e. the conditional probability distributions of the source and target device are different). This can make use of similar techniques to achieve transfer as the transductive case. However, here, observed training examples naturally receive a higher weight than 'transferred' examples, with the exact weights decided by a distance metric between the source and target.

*4.4. Related Concepts in Time Series Analysis*

An alternative formulation in time series forecasting literature is the use of "Global Forecasting Models" [67]. These are used to predict many time series simultaneously, using a single model with shared parameters. When the time series are related in some way, this can lead to a considerable boost in predictive accuracy while also substantially lowering barriers to scalability by cutting down on models that need to be trained and maintained. Another benefit of such types of models is their ability to more accurately predict extreme or rare events [68]. This formulation makes it quite similar to multi-task learning in supervised learning contexts [69]. One important distinction to keep in mind between transfer and the formulation of global forecasting models is between the source and target domain or task. In transfer learning, as defined above, only the model's performance on the target is relevant. On the other hand, in a global forecasting model, there is no such distinction between a source and a target: every task contributes to the loss function evaluation.

An alternative research direction has been the use of feature-based learning in large time series datasets to improve forecasting results. The FFORMS and FFORMA [70] algorithms perhaps best typify this line of work. The underlying idea behind these algorithms is either automatic model selection or model combination. The first step in such methods is to extract underlying features from a large number of time series. The next step is to build different models for these time series. The final step in the training phase is to build a meta-model which learns the relationship between the extracted features and the predictive accuracy of different models. During the test phase, when a new dataset is encountered, the same features are extracted and fed to the meta-model. The meta-model provides information on which model should be used (FFORMS) or how multiple models should be combined (FFORMA). Note that there is a difference between how this method and the shared feature representations mentioned earlier have been applied so far. In the feature-based learning method discussed here, the extracted features are static, i.e. defined using mathematical formulas, and are only used for the purpose of summarizing different time series and deciding, which model is to be used with a particular time series. The shared feature representations, on the other hand, can be used directly in the modelling task. Nevertheless, it is still possible to combine the two approaches.

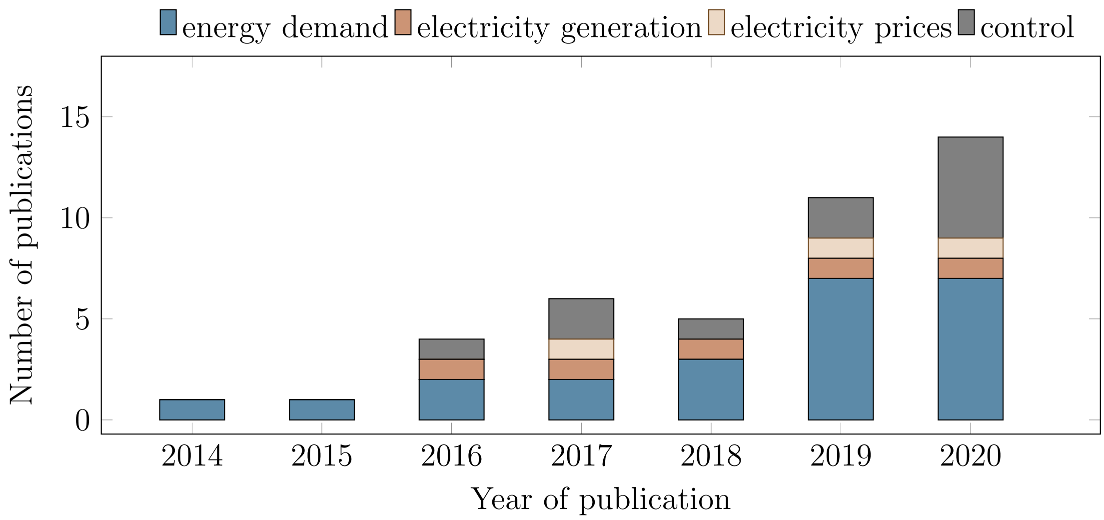


Figure 6: Articles classified per application and year of publication based on Table 2.

## 5. Application Scenarios

The benefits of transfer learning in the smart grid setting, specifically in DR applications have been illustrated in the literature through several uses cases - both in simulation and real-world settings. As shown in Fig. 6, the use of transfer learning in demand response applications has steadily gained attention since 2017, with a bulk of the applications focusing on energy demand forecasting and control. In this section, we present a detailed discussion of these applications, following the same high-level framework defined in the previous sections.

### *5.1. Transfer in Supervised Learning*

In the supervised learning setting, transfer learning has mainly been used in the context of forecasting electricity generation from renewable energy sources - wind and solar, energy demand and electricity price, as mentioned previously.

#### *5.1.1. Electrical Energy Generation*

The principles behind electricity generation through renewable sources such as solar PV systems and wind turbines are well understood. However, purely relying on these physical models can prove to be quite inaccurate because they ignore local effects such as shading, soiling and real-time effects. Likewise, forecasting electricity production using only historic on-site data is error-prone as it ignores physical knowledge of the systems, and relies solely on (possibly sparse) observations. By combining the two streams of information in one of the ways highlighted above, better performing models can be obtained.

One of the earliest work on transfer learning for forecasting renewable energy generation was done in 2016 by Hu *et al*. [71]. The authors transferred high-level representations of wind-speed patterns learned from data-rich wind farms as additional features for training forecasting models of newly constructed farms. In the work of Qureshi *et al*. [72, 73], weights from the source forecasting model were used as initial weights for the target model in order to reduce the time required to train a model from scratch. The authors considered knowledge transfer across different task domains - wind power to wind speed prediction - and between wind farms in different regions. In a similar vein, the authors of [74, 75] transferred trained weights from the source to the target model for forecasting PV power generation. The source models were trained using historical solar irradiance data from existing solar farms in a nearby location.

*5.1.2. Energy Demand*

Transfer learning for forecasting of electricity demand is arguably the area that has received the most attention in recent years. To improve the load forecasting error in target buildings with limited amounts of data, data set sharing between the source and target has been widely used. For example, by using data sets from different cities [76, 77, 78, 79, 80] and different types of buildings - school and office buildings [81, 82, 83], residential buildings [84] - in the same or different geographical locations, with different distributions and seasonal profiles to train a base forecasting model for the target building. This base model is then fine-tuned with the data from the target building. Also, in Jain *et al.* [85], data and parameters from a physics-based simulation model were used to improve accuracy of a model for forecasting electricity consumption in buildings with limited and sparse data. The above mentioned works showed a significant reduction (up to 78%) in prediction errors when transfer learning is used compared to models trained from scratch even if the buildings are not similar.

An alternative way of using transfer learning is by merging data sets from a large number of residential and non-residential households to train a base model, which extracts relevant features for forecasting the energy demand of the target building [86, 87]. By merging data sets, the trained base model extracts features that are not specific to a particular type of building, making it necessary to fine-tune the model using the (sparse) data of the target. Additionally, sharing essential features in the source data set with the target has also shown to increase forecasting accuracy in the target building: the authors of [88, 89, 90] shared historical load data, whereas [91] shared historical load and weather features such as wind. It is worth mentioning that Wu *et al.* [89] also showed the effectiveness of transfer learning in both homogeneous (up to 43.10% improvement in accuracy) and heterogeneous (up to 46.87% improvement in accuracy) settings.

Motivated by the lack of sufficient data on incentive-based demand response events, Cai *et al.* [92] used load data of related (source) customers to train a model that predicts the customers' response to DR incentives during peak periods. Bandara *et al.* [93] proposed the use of augmented time series - *i.e.* a synthetic collection of time series generated based on the original time series data set - to pre-train a global energy demand forecasting model. The knowledge representations from this model are used to train a more accurate forecasting model for the target. In the above mentioned works, considerable accuracy gains ranging between $9.5 - 30\%$ have been observed, compared to the case where transfer learning was not used.

Transferring knowledge in the form of trained neural network weights has also been exploited in load forecasting. The objective of this weight transfer has been either to reduce computational needs for model training [94] or to compensate for the insufficient data in new buildings/users [95, 96, 97]. This weight transfer resulted in a $7.3 - 30\%$ improvement in prediction accuracy. Cluster-based transfer learning has also been used to transfer weights of a model - trained with data from the most representative building or centroid of each cluster - between buildings in the same cluster for load forecasting in a neighbourhood with multiple households [98, 99]. These weights are used to initialize the forecasting models for all (target) buildings belonging to the same cluster leading to improvements of $3.17 - 15.07\%$ in the prediction accuracy. In [100] and [101] load profiles in the same cluster were used to construct a pool of profiles, which were then used to train a deep neural network [100] or a Bayesian neural network [101] that learns to extract and select feature representations for training a probabilistic load forecasting model. These additional feature representations led to a prediction accuracy improvement of $5.6 - 7.1\%$.

As opposed to appliance demand, it is possible to build models for thermal systems such as heat pumps, air conditioners, etc, which combine domain knowledge with observation data. These models, built in a purely data-driven supervised learning context, take some observed variables such as ambient conditions, historical temperatures and power drawn by the heating or cooling element as input to predict the internal temperature (or state of charge) in the medium of interest (*e.g.*, a building or hot water vessel, etc). However, models built in this way remain domain agnostic and tend to generalize poorly on unseen data. Thus, incorporating domain knowledge into the predictive models via transfer of knowledge instances or parameters has shown to greatly accelerate learning performance in supervised contexts [66], similar to the case of generation models.

As such, transfer learning has also been used to transfer knowledge on learned system dynamics - the system dynamics are learned in a data-driven supervised learning fashion - across similar systems/devices. Kazmi *et al.* [16] combined features from several households into a single feature vector used by a neural network to learn a dynamics model. The authors showed that by transferring knowledge across both homogeneous and heterogeneous hot water systems, convergence to a reliable model was achieved within a few weeks of data collection as opposed to months or years without knowledge transfer. The authors also reported a performance improvement of $13.7 - 24.3\%$ in the learned dynamics model. Similar work was done by Grubinger *et al.* [102] for predicting the heat-mass transfer dynamics in residential buildings, and by Chen *et al.* [103] for predicting the thermal dynamics of a building: indoor temperatures and relative humidity. Moreover, Jain *et al.* [104] transferred first-order thermal model parameters for simulating temperatures in cold-rooms with the goal of improving refrigerant leakage detection across buildings. These model parameters were continuously updated as more data was collected to accommodate temporal changes in the building's thermal behavior.

A recent application of transfer learning is for identification of residential loads or non-intrusive load monitoring [105]. Nonintrusive load monitoring allows to identify how the different loads in a building contribute to its energy consumption, which is essential for implementation of DR programs. Cavalca *et al.* [105] showed that by using a pre-trained (CNN VGG16 [106]) model to extract relevant features from the electricity consumption data, a performance gain of $6.4\% - 22.4\%$ was obtained compared to other feature extraction methods that do not use transfer learning.

*5.1.3. Market Prices*

In the context of electricity price forecasting, transfer learning has not been widely adopted to date. Gunduza *et al*. [107] combined features from different electricity markets - Belgian, French, German, Nord Pool and Turkish markets - to train a forecasting model, which was then fine-tuned with features from the target market. In the work of Lago *et al*. [108] the authors used some features from the French electricity market for training an electricity price forecasting model for the Belgian electricity market. In their work, the authors observed a 12.5% improvement in the forecasting accuracy compared to when only data from the Belgian electricity market was used.

*5.2. Transfer Learning in Reinforcement Learning*

Transfer learning for reinforcement learning in the context of demand response is still in a nascent stage of development. Nevertheless, a number of case studies have appeared in the recent past that use it to improve models or control policies.

One of the very first applications of transfer in RL was actually for forecasting energy demand of buildings with limited historical data [109]. The authors transferred forecasting models trained using RL algorithms - SARSA and Q-learning with deep belief networks for function approximation - and data from source buildings to predict the energy demand in (commercial and residential) buildings with unlabelled historical data. Similarly, Kong *et al*. [110] transferred knowledge on the user's elasticity of electricity price from regions where DR has been implemented to areas with unknown elasticity. This elasticity is used to estimate the electricity demand of the users in the new region, which is then used to train a RL algorithm - SARSA - that selects suitable retail electricity prices to enforce DR in this new region.

A parallel thread of research has focused on using transfer learning to improve reinforcement learning based control strategies of flexible assets. An early example of using prior knowledge in an RL system using a hybrid simulation learning control can be found in [111], where the authors adopt a two-step approach. In the first step, they pre-train the controller using a calibrated model of the HVAC system under consideration. Then, this controller is updated online during the operational phase in an experimental environment. Likewise, in the works of Costanzo *et al*. [29] and Ruelens *et al*. [18] the authors used transfer learning in the form of expert knowledge to shape and enforce monotonicity in a control policy previously learned from a limited number of observations to improve the accuracy of the policy for controlling TCLs. The authors of [58] proposed using expert knowledge in the form of grey-box model predictive control transitions - transitions based on a grey-box model of an electric water heater - for kick-starting of an informed fitted Q-iteration-based controller. The authors used a linear grey-box model predictive control approach as the expert and showed an increase in cost savings. Peirelinck *et al*. [17] used transfer learning through domain randomization to facilitate knowledge transfer and reduce the exploratory time of the learning agent. The authors reported an 8.8% increase in cost savings compared to the setting without any knowledge transfer. This approach has the added advantage of not requiring a well-calibrated simulation model of the system under consideration. Likewise, domain randomization in RL was also used by Kazmi *et al.* [112] to control batteries in order to solve voltage problems in the low voltage grid. Control agents were trained offline using randomly sampled load and PV generation profiles in many different simulated topologies of the distribution grid. The authors showed that by employing domain randomization more grid violations were resolved compared to the case without.

In the work of Mbuwir *et al.* [51], the authors used cluster-based transfer learning to transfer knowledge in the form of control policies amongst buildings with similar energy usage patterns. A control policy learned using data from a data-rich building in a cluster is used to initialise learning in a target building belonging to the same cluster. The authors showed a faster convergence to a near optimal policy compared to when no knowledge was transferred. Similarly, Paridari *et al.* [50] proposed a plug-and-play planning and control framework for control energy storage devices in buildings with PV installations. In their work, knowledge was transferred in the form of a policy function approximation to new end users with no historical data - from which a control policy could be learned- leading to a 29% increase in cost savings. Likewise, in [59] the authors exploit the structural similarities in the control policies across rooms in a building by sharing features (and transitions) to obtain a policy that can set the suitable energy levels for lighting and air-conditioning units. To avoid excessive use of cloud resources and speed up the training process when training RL-based control agents from scratch in DR applications, Tao *et al.* [113] transferred weights of the control policies between batteries and Heating, Ventilation and Air Conditioning (HVAC) units. The authors showed: a significant cost savings in a homogeneous setting (knowledge transferred between two battery control agents), and a slight cost savings in a heterogeneous setting (knowledge transferred from a battery to an HVAC control agent) compared to training an entirely new policy. Moreover, the authors showed that knowledge can be transferred between different DR programs (price-based to direct load control).

Even though the above applications have shown the positive impact of transfer learning on the target domain learner, the effectiveness of the knowledge transfer is not always guaranteed. Knowledge transfer can also lead to reduced performance in the target domain/task. This is termed negative transfer and can occur due to source and target tasks being unrelated or the domain data distributions being too different. For example, a 1.36% reduction in prediction accuracy was reported in [78] due to negative transfer. Several approaches exists in the literature for mitigating negative transfer as summarised in [114]. In the context of demand response, negative transfer has been tackled by selecting appropriate source tasks using a Gaussian process-based selection algorithm [76] or using TrAdaBoost algorithm [78], which decreases the weights of source instances with distributions different from that of the target. Establishing the similarity between source and target tasks, and performing optimization to determine the appropriate number of source tasks - large number of source task could increase the computational time while too few tasks might not provide sufficient supplementary information - also mitigates negative knowledge transfer [78].

Table 2 provides a summary of the articles that have applied transfer learning to demand response applications. The table is classified based on the demand response application and the percentage improvement when transfer learning is used. As can be seen in the

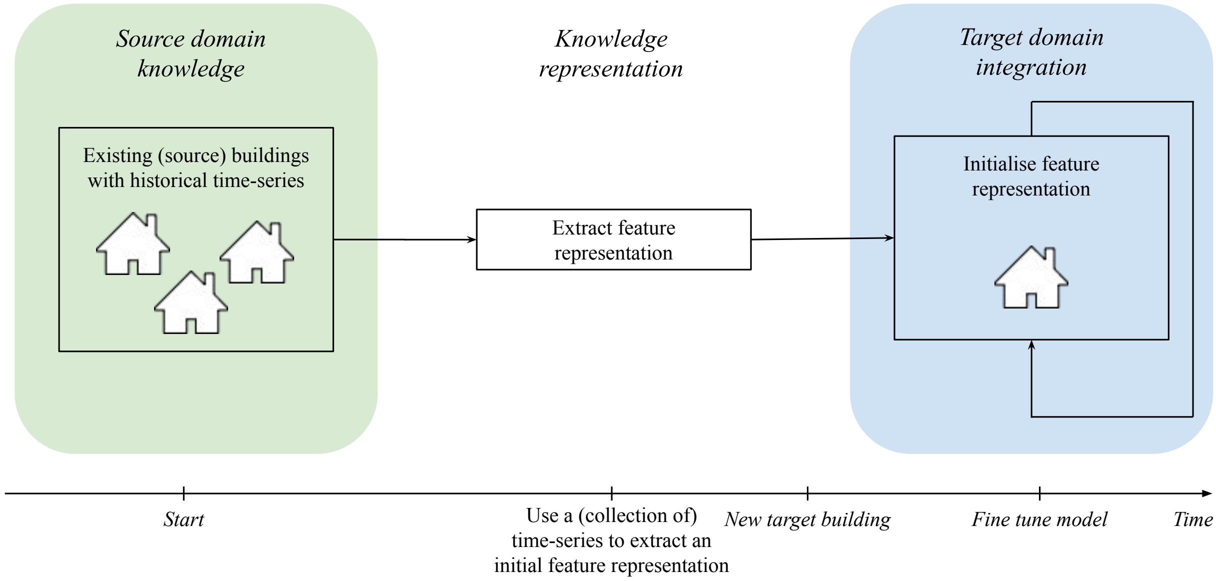


Figure 7: Visual illustration of transfer learning in demand response applications.

table, articles in which the percentage improvement brought by transfer learning have been reported, the improvements are typically greater than 3%, and exceed 10% in many cases. However, some articles do not explicitly report a specific percentage improvement value but provide figures/tables to illustrate the performance improvement. Likewise, many studies only report improvements in asymptotic performance, but do not distinguish between the other dimensions of performance we highlighted earlier (initial performance and learning performance). Fig. 7 provides a general visualization of how the different articles have applied transfer learning in the demand response use cases discussed above.

## 6. Future Directions

The presented review shows that transfer learning has gained considerable traction in recent years, and is currently the subject of intensifying research efforts in DR. This attention can mainly be attributed to the success of transfer learning methods in other domains, and early indications that the methods hold enormous potential for DR applications as well. The review also shows that major research opportunities remain untapped, with most research to date being concentrated on improving demand forecasts using transfer learning. This section explores and elaborates the identified limits and resulting research opportunities of transfer learning within a DR setting. Table 3 summarises all applications that have been reviewed in the preceding section, and provides an overview of the research gaps that remain within transfer learning applications in DR.

A first observation made based on Table 3 is that many of the supervised learning applications focus on using transfer learning to jump-start forecasting ability. They thus transfer features and/or data from a source domain to a target domain. Mostly, source and target

Table 2: An overview of reviewed papers.

| Article ref | Year | Application | Category | Improvement |
|---|---|---|---|---|
| [91] | 2014 | energy demand | real-world data | $4.54 - 23.93\%$ |
| [76] | 2015 | energy demand | real-world data | - |
| [71] | 2016 | wind power generation | real-world data | - |
| [109] | 2016 | energy demand | real-world data | - |
| [90] | 2016 | energy demand | real-world data | - |
| [29] | 2016 | climate control | domain knowledge | - |
| [18] | 2017 | heat pump control | inductive transfer | - |
| [73] | 2017 | wind power generation | real-world data | - |
| [108] | 2017 | electricity prices | inductive transfer | 12.5% |
| [102] | 2017 | system dynamics | real-world data | - |
| [77] | 2017 | energy demand | real-world data | 9.5% |
| [88] | 2017 | energy demand | real-world-data | - |
| [75] | 2018 | PV generation | real-world data | - |
| [81] | 2018 | energy demand | real-world data | 11.2% |
| [84] | 2018 | energy demand | shared features | - |
| [50] | 2018 | battery control | real-world data | 29% |
| [87] | 2018 | energy demand | real-world data | - |
| [78] | 2019 | energy demand | real-world data | 28% |
| [104] | 2019 | energy demand | real-world data | - |
| [89] | 2019 | energy demand | real-world data | $14.37 - 43.10\%$ |
| [72] | 2019 | wind speed | real-world data | - |
| [94] | 2019 | energy demand | real-world data | 7.3% |
| [96] | 2019 | energy demand | real-world data | $20 - 30\%$ |
| [101] | 2019 | energy demand | real-world data | $5.6 - 7.1\%$ |
| [100] | 2019 | energy demand | real-world data | - |
| [110] | 2019 | electricity prices | shared features | - |
| [74] | 2020 | PV generation | real-world data | - |
| [82] | 2020 | energy demand | real-world data | $15 - 78\%$ |
| [80] | 2020 | energy demand | real-world data | $0.01 - 47.4\%$ |
| [92] | 2020 | energy demand | real-world data | - |
| [93] | 2020 | energy demand | inductive transfer | - |
| [79] | 2020 | energy demand | real-world data | 30% |
| [95] | 2020 | energy demand | real-world data | 19.69% |
| [99] | 2020 | energy demand | real-world data | $3.79 - 5.10\%$ |
| [16] | 2020 | system dynamics | real-world data | $24.3 - 13.7\%$ |
| [103] | 2020 | system dynamics | real-world data | $73.3 - 83.3\%$ |
| [107] | 2020 | electricity prices | shared features | - |
| [17] | 2020 | electric water heater control | domain knowledge | 8.8% |
| [51] | 2020 | battery control | real-world data | - |
| [59] | 2020 | air-conditioning system control | shared features | - |
| [97] | 2020 | energy demand | real-world data | $8.2 - 28.9\%$ |
| [83] | 2021 | energy demand | real-world data | - |
| [86] | 2021 | energy demand | real-world data | $7.84 - 15.07\%$ |
| [98] | 2021 | energy demand | real-world data | 3.17% |
| [85] | 2021 | energy demand | simulated data | - |
| [105] | 2021 | energy demand | real-world data | $6.4 - 24.4\%$ |
| [112] | 2021 | battery control | real-world data | - |
| [113] | 2021 | battery and HVAC | real-world data | - |

Table 3: Classification of transfer learning applications in demand response.

| | Supervised Learning | Reinforcement Learning |
|---|---|---|
| Transductive transfer | | |
| Domain knowledge | | [28, 29, 58, 17] |
| Real-world data | [71, 73, 74, 75, 81, 89] [88, 95, 96, 97, 92, 82, 83] [98, 99, 94, 115, 78, 79, 76] [80, 87, 16, 103, 90, 91, 100] | [50, 51, 109] |
| Shared features | [84], [107] | [59], [110] |
| Inductive transfer | [93, 108] | [18, 26] |

domains are different instances of the same physical construct. For example, from one building to another building. In this sense, these applications belong to the transductive transfer learning setting, as they have the same goal, *i.e.*, reducing forecasting error. However, they can be classified under *real-world data* transfer as well as *shared features*, as they use the data and/or learned features of the source domain to jump-start performance in the target domain. Here, only those that explicitly transform features with the aim of increasing transfer performance have been classified under *shared features.* A second observation made based on Table 3 is that most of the reinforcement learning based transfer learning applications within the DR setting focus on utilising domain knowledge to increase control performance. An example of this is controllers trained using simulators.

Furthermore, from Table 3, it is clear that in every area of transfer learning within DR there remain research opportunities. But, especially inductive transfer learning and transductive learning with feature sharing between different domains remain open challenges. Although inductive transfer within reinforcement learning has seen few research applications, it could be a promising field in the near future. For example, one can think of a DR setting where a flexibility provider offers different services to the market, *e.g.*, frequency response and peak shaving. At different times, this provider would then have to switch between reward-functions, transferring as much relevant information about the underlying system dynamics as possible. Likewise, transductive learning with feature sharing has relevant applications in practice but has seen relatively little research interest. For example, energy storage devices can be based on multiple technologies. However, at a high enough abstraction level they have similar dynamics. Therefore, features could be shared among different storage appliances. Transfer could then be used to jump-start performance of a new storage technology entering a flexibility provider's set of appliances. It could also be used to increase overall performance of the flexibility pool among all active appliances.

Finally, it is interesting to revisit Fig. 1. While it is now clear that almost all applications considered represent the knowledge of the source domain either with *Simulation Results* or with *Historic Data*, other representations of knowledge are possible. Simultaneously, most knowledge of the source domain and task is integrated in the target domain and task by incorporating it in the agent's training set. Taking the broader field of informed ML into account, it is also possible to integrate knowledge through other means. For example, one could restrict the hypothesis set by a pre-defined model structure of the NN, or by modifying the loss function of the model. As far as the reinforcement learning algorithm goes, our review clearly shows Q-learning is widespread and well-used in DR applications. Many researchers seem to agree that Q-learning is promising, as it has shown good results in past research and in other domains. However, in other domains, such as robotics, state-of-the-art policy iteration algorithms have proven to handle challenging tasks better [47]. Therefore, it is important to follow-up on the advances that have been made elsewhere and investigate their applicability for DR, especially in the context of transfer learning. This might be necessary for such algorithms to become sophisticated enough to handle real-world challenges, which arise as much from sparse data as they do from other limitations such as poor quality data, and ill-defined and often conflicting objectives.

## 7. Conclusion

The recent adoption of machine learning-based techniques in demand response applications has been influenced by the availability of data through smart metering and the smart grid in general. However, using these techniques in newly constructed systems remains challenging due to their lack of (sufficient) historical data. Transfer learning has the potential to solve this challenge and improve generalizability of machine-learning based models and control policies as seen in this review. This is evident in the performance gains we have observed in the papers reviewed herein. In many cases, this can be the difference between machine learning models that can learn from the available limited data vs. those that fail to converge to an accurate solution. However, to date, a majority of articles has focused on transfer learning for forecasting energy demand, with only limited attention paid to renewable energy generation and electricity price forecasting. A few, but increasing number of, articles on modelling system dynamics and control of electric water heaters and batteries for energy storage have also appeared in the very recent past.

In transfer learning literature, a wide variety of methods - ranging from data and parameter sharing to learned representations have been explored. Although a plurality of the articles reviewed in this paper have considered knowledge transfer in the form of pre-trained weights from the source forecasting model to initialise learning in the target, a few articles have looked into sharing feature representations. Likewise, global forecasting models which employ multi-task learning have emerged as a practical way to handle the operational complexity of multiple individual models, while achieving transfer to improve generalization. Models built in this way also have additional benefits in terms of scalability, whereby a vastly reduced number of models can be used for a large number of tasks. In the context of reinforcement learning, knowledge transfer in the form of domain/expert knowledge has been explored with the literature showing how domain knowledge can accelerate learning of adequate control policies.

A critical shortcoming we have identified is that almost half the articles reviewed do not provide any quantification of gains attributed to transfer learning. Furthermore, even when such numbers are included, they mostly do not provide a complete representation of the gains that could be associated with transfer learning. These include improvements to initial and asymptotic performance as well as the rate of improvement (Fig. 2). These concerns are further exacerbated by the fact that most studies do not open-source their data or codebase, often due to privacy concerns, and are consequently not reproducible. A recommendation for future research is therefore to quantify all of these three metrics, especially against strong baselines, and open-source the trained models in a responsible manner when sharing code and data is not possible.

While research on transfer learning for demand response applications is still in its infancy, we have found that most of the use cases have only been tested in simulation environments. Consequently, in order to prove the effectiveness of transfer learning for demand response applications, more real-world experiments need to be conducted. This is especially true due to the added challenges that can arise while deploying models that make use of transfer learning. Moreover, in the context of reinforcement learning in DR, we have observed only limited applications of transfer learning despite its tremendous potential. In light of all this, we have suggested different directions in which research could advance the field of both supervised and reinforcement learning in demand response by incorporating transfer of knowledge.

## 8. Acknowledgments

Hussain Kazmi acknowledges support from Research Foundation – Flanders (FWO), Belgium (research fellowship 1262921N) in the preparation of this manuscript. Thijs Peirelinck, Brida V Mbuwir, Chris Hermans and Fred Spiessens acknowledge support from the Flemish Institute for Technological Research (VITO) in the preparation of this manuscript. Johan Suykens acknowledges support from ERC AdG E-DUALITY (787960), KU Leuven C14/18/068, FWO GOA4917N, EU H2020 ICT-48 Network TAILOR, Ford KU Leuven Research Alliance KUL0076, Impulsfonds AI: VR 2019 2203 DOC.0318/1QUATER, KU Leuven AI institute.

## References

[1] F. Steinke, P. Wolfrum, and C. Hoffmann, "Grid vs. storage in a 100% renewable Europe," *Renewable Energy*, vol. 50, pp. 826–832, Feb 2013.

[2] N. Leemput, F. Geth, J. Van Roy, P. Olivella-Rosell, J. Driesen, and A. Sumper, "MV and LV residential grid impact of combined slow and fast charging of electric vehicles," *Energies*, vol. 8, pp. 1760–1783, Mar 2015.

[3] S. Riaz, H. Marzooghi, G. Verbic, A. C. Chapman, and D. J. Hill, "Generic Demand Model Considering the Impact of Prosumers for Future Grid Scenario Analysis," *IEEE Transactions on Smart Grid*, vol. 10, pp. 819–829, Jan 2019.

[4] R. Liu, G. Verbič, J. Ma, and D. J. Hill, "Fast stability scanning for future grid scenario analysis," *IEEE Transactions on Power Systems*, vol. 33, pp. 514–524, Jan 2018.

[5] N. Hary, V. Rious, and M. Saguan, "The electricity generation adequacy problem: Assessing dynamic effects of capacity remuneration mechanisms," *Energy Policy*, vol. 91, pp. 113–127, Apr 2016.

[6] D. Jin, R. Ocone, K. Jiao, and J. Xuan, "Energy and AI," *Energy and AI*, vol. 1, p. 100002, aug 2020.

[7] I. Antonopoulos, V. Robu, B. Couraud, and D. Flynn, "Data-driven modelling of energy demand response behaviour based on a large-scale residential trial," *Energy and AI*, vol. 4, p. 100071, jun 2021.

[8] J. Bedi and D. Toshniwal, "Deep learning framework to forecast electricity demand," *Applied Energy*, vol. 238, pp. 1312–1326, Mar 2019.

[9] M. Naimur Rahman, A. Esmailpour, and J. Zhao, "Machine Learning with Big Data An Efficient Electricity Generation Forecasting System," *Big Data Research*, vol. 5, pp. 9–15, Sep 2016.

[10] P. Mandal, T. Senjyu, and T. Funabashi, "Neural networks approach to forecast several hour ahead electricity prices and loads in deregulated market," *Energy Conversion and Management*, vol. 47, pp. 2128–2142, Sep 2006.

[11] M. Shafiullah, M. Ijaz, M. A. Abido, and Z. Al-Hamouz, "Optimized support vector machine & wavelet transform for distribution grid fault location," in *2017 11th IEEE International Conference on Compatibility, Power Electronics and Power Engineering, CPE-POWERENG 2017*, pp. 77–82, IEEE, Apr 2017.

[12] R. Habibifar, A. Aris Lekvan, and M. Ehsan, "A risk-constrained decision support tool for EV aggregators participating in energy and frequency regulation markets," *Electric Power Systems Research*, vol. 185, p. 106367, Aug 2020.

[13] J. Cigler, D. Gyalistras, J. Široky, V. Tiet, and L. Ferkl, "Beyond theory: the challenge of implementing model predictive control in buildings," in *Proceedings of 11th Rehva World Congress, Clima*, vol. 250, 2013.

[14] M. Liu, S. Peeters, D. S. Callaway, and B. J. Claessens, "Trajectory tracking with an aggregation of domestic hot water heaters: Combining model-based and model-free control in a commercial deployment," *IEEE Transactions on Smart Grid*, vol. EA, p. 1, 2019.

[15] R. Henry and D. Ernst, "Gym-ANM: Reinforcement learning environments for active network management tasks in electricity distribution systems," *Energy and AI*, vol. 5, p. 100092, sep 2021.

[16] H. Kazmi, J. Suykens, and J. Driesen, "Large-Scale Transfer Learning For Data-Driven Modelling Of Hot Water Systems," in *Proceedings of Building Simulation 2019: 16th Conference of IBPSA*, vol. 16, pp. 2611–2618, IBPSA, Mar 2020.

[17] T. Peirelinck, C. Hermans, F. Spiessens, and G. Deconinck, "Domain Randomization for Demand Response of an Electric Water Heater," *IEEE Transactions on Smart Grid*, vol. 12, pp. 1370–1379, Sep 2020.

[18] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuska, and R. Belmans, "Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2149–2159, 2017.

[19] M. R. Kazemzadeh, A. Amjadian, and T. Amraee, "A hybrid data mining driven algorithm for long term electric peak load and energy demand forecasting," *Energy*, vol. 204, p. 117948, Aug 2020.

[20] E. M. de Oliveira and F. L. Cyrino Oliveira, "Forecasting mid-long term electric energy consumption through bagging ARIMA and exponential smoothing methods," *Energy*, vol. 144, pp. 776–788, Feb 2018.

[21] F. Ziel and R. Weron, "Day-ahead electricity price forecasting with high-dimensional structures: Univariate vs. multivariate modeling frameworks," *Energy Economics*, vol. 70, pp. 396–420, Feb 2018.

[22] F. Pallonetto, C. Jin, and E. Mangina, "Forecast electricity demand in commercial building with machine learning models to enable demand response programs," *Energy and AI*, p. 100121, oct 2021.

[23] B. J. Claessens, P. Vrancx, and F. Ruelens, "Convolutional Neural Networks For Automatic State-Time Feature Extraction in Reinforcement Learning Applied to Residential Load Control," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3259 – 3269, 2018.

[24] T. Leurs, B. J. Claessens, F. Ruelens, S. Weckx, and G. Deconinck, "Beyond theory: Experimental results of a self-learning air conditioning unit," in *2016 IEEE International Energy Conference (ENERGYCON)*, pp. 1–6, Apr 2016.

[25] B. J. Claessens, S. Vandael, F. Ruelens, and M. Hommelberg, "Self-learning demand side management for a heterogeneous cluster of devices with binary control actions," in *2012 3rd IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, pp. 1–8, IEEE, Oct 2012.

[26] F. Ruelens, B. J. Claessens, S. Quaiyum, B. De Schutter, R. Babuska, and R. Belmans, "Reinforcement Learning Applied to an Electric Water Heater: From Theory to Practice," *IEEE Transactions on*

*Smart Grid*, vol. 9, pp. 3792–3800, Jul 2018.

[27] M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning Dexterous In-Hand Manipulation," *The International Journal of Robotics Research*, vol. 39, Nov 2020.

[28] T. Lampe and M. Riedmiller, "Approximate model-assisted Neural Fitted Q-Iteration," in *Proceedings of the International Joint Conference on Neural Networks*, pp. 2698–2704, IEEE, Sep 2014.

[29] G. T. Costanzo, S. Iacovella, F. Ruelens, T. Leurs, and B. J. Claessens, "Experimental analysis of data-driven control for a building heating system," *Sustainable Energy, Grids and Networks*, vol. 6, pp. 81–90, Jun 2016.

[30] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: a Survey," Sep 2020.

[31] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-Real Transfer of Robotic Control with Dynamics Randomization," Oct 2017.

[32] F. Pallonetto, M. De Rosa, F. Milano, and D. P. Finn, "Demand response algorithms for smart-grid ready residential buildings using machine learning models," *Applied energy*, vol. 239, pp. 1265–1282, 2019.

[33] I. Antonopoulos, V. Robu, B. Couraud, D. Kirli, S. Norbu, A. Kiprakis, D. Flynn, S. Elizondo-Gonzalez, and S. Wattam, "Artificial intelligence and machine learning approaches to energy demand-side response: A systematic review," *Renewable and Sustainable Energy Reviews*, vol. 130, p. 109899, 2020.

[34] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," vol. 235, pp. 1072–1089, Feb 2019.

[35] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, "A survey of transfer learning," *Journal of Big Data*, vol. 3, pp. 1–40, Dec 2016.

[36] M. E. Taylor and P. Stone, "Transfer Learning for Reinforcement Learning Domains: A Survey," vol. 10, pp. 1633–1685, 2009.

[37] L. von Rueden, S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese, B. Kirsch, M. Walczak, J. Pfrommer, A. Pick, R. Ramamurthy, J. Garcke, C. Bauckhage, and J. Schuecker, "Informed Machine Learning - A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–1, May 2021.

[38] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nature Reviews Physics 2021 3:6*, vol. 3, pp. 422–440, may 2021.

[39] H. Wu and R. Qiao, "Physics-constrained deep learning for data assimilation of subsurface transport," *Energy and AI*, vol. 3, p. 100044, mar 2021.

[40] H. Gao, L. Sun, and J. X. Wang, "PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain," *Journal of Computational Physics*, vol. 428, p. 110079, mar 2021.

[41] S. J. Pan and Q. Yang, "A survey on transfer learning," vol. 22, no. 10, pp. 1345–1359, 2010.

[42] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction.* MIT Press, 2 ed., 2017.

[43] V. N. Vapnik, *The Nature of Statistical Learning Theory.* Springer New York, 2000.

[44] V. N. Vapnik, *Statistical learning theory.* Wiley, 1998.

[45] A. Gammerman, V. Vovk, and V. N. Vapnik, "Learning by Transduction," in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 148–155, 1998.

[46] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," in *NIPS Deep Learning Workshop 2013*, 2013.

[47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Jul 2017.

[48] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, R. Sepassi, G. Tucker, and H. Michalewski, "Model-Based Reinforcement Learning for Atari," in *The International Conference on Learning Representations (ICLR) 2020*, Mar

2020.
[49] C. Hesse, J. Schulman, V. Pfau, A. Nichol, O. Klimov, and L. Schiavo, "Retro Contest," 2018.
[50] K. Paridari, D. Azuatalam, A. C. Chapman, G. Verbič, L. Nordström, D. Azuatalam, A. C. Chapman, and G. Verbič, "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)*, IEEE, 2018.
[51] B. V. Mbuwir, K. Paridari, F. Spiessens, L. Nordstrom, and G. Deconinck, "Transfer learning for operational planning of batteries in commercial buildings," in *2020 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, SmartGridComm 2020*, IEEE, Nov 2020.
[52] B. V. Mbuwir, F. Spiessens, and G. Deconinck, "Distributed optimization for scheduling energy flows in community microgrids," *Electric Power Systems Research*, vol. 187, p. 106479, Oct 2020.
[53] C. Patyn, F. Ruelens, and G. Deconinck, "Comparing neural architectures for demand response through model-free reinforcement learning for heat pump control," in *2018 IEEE International Energy Conference, ENERGYCON 2018*, pp. 1–6, IEEE, Jun 2018.
[54] C. Patyn, T. Peirelinck, and G. Deconinck, "Intelligent Electric Water Heater Control with Varying State Information," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, p. 7, IEEE, 2018.
[55] T. Peirelinck, F. Ruelens, and G. Deconinck, "Using reinforcement learning for optimizing heat pump control in a building model in Modelica," in *2018 IEEE International Energy Conference (ENERGYCON)*, pp. 1–6, IEEE, 2018.
[56] T. Peirelinck, F. Spiessens, C. Hermans, and G. Deconinck, "Double Q-learning for Demand Response of an Electric Water Heater," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*, IEEE, 2019.
[57] A. Soares, D. Geysen, F. Spiessens, D. Ectors, O. De Somer, and K. Vanthournout, "Using reinforcement learning for maximizing residential self-consumption – Results from a field test," *Energy and Buildings*, vol. 207, 2020.
[58] C. Patyn and G. Deconinck, "Electric Water Heater Control Through Informed Fitted Q-Iteration," in *Proceedings of 2019 IEEE PES Innovative Smart Grid Technologies Europe, ISGT-Europe 2019*, IEEE, Sep 2019.
[59] S. Kim, R. Mowakeaa, S. J. Kim, and H. Lim, "Building energy management for demand response using kernel lifelong learning," *IEEE Access*, vol. 8, pp. 82131–82141, 2020.
[60] A. Tirinzoni, A. Sessa, M. Pirotta, and M. Restelli, "Importance Weighted Transfer of Samples in Reinforcement Learning," in *Proceedings of the 35th International Conference on Machine Learning*, pp. 4936–4945, 2018.
[61] C. D'eramo, D. Tateo, A. Bonarini, M. Restelli, and J. Peters, "Sharing Knowledge in Multi-task Deep Reinforcement Learning," in *2020 International Conference on Learning Representations*, 2020.
[62] M. E. Taylor, N. K. Jong, and P. Stone, "Transferring instances for model-based reinforcement learning," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5212 LNAI, pp. 488–505, Springer, Berlin, Heidelberg, 2008.
[63] C. Finck, P. Beagon, J. Clauß, T. Péan, P. J. C. Vogler-Finck, K. Zhang, and H. Kazmi, "Review of applied and tested con-trol possibilities for energy flexi-bility in buildings," tech. rep., IEA EBC Annex 67 Energy Flexible Buildings, 2018.
[64] A. Afram and F. Janabi-Sharifi, "Review of modeling methods for HVAC systems," *Applied Thermal Engineering*, vol. 67, pp. 507–519, Jun 2014.
[65] H. Kazmi, Í. Munné-Collado, F. Mehmood, T. A. Syed, and J. Driesen, "Towards data-driven energy communities: A review of open-source datasets, models and tools," *Renewable and Sustainable Energy Reviews*, vol. 148, p. 111290, 2021.
[66] H. K. F. Mehmood, S. Lodeweyckx, and J. Driesen, "Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads," *Applied Energy*, vol. 238, pp. 1022–1035, Mar 2019.
[67] P. Montero-Manso and R. J. Hyndman, "Principles and Algorithms for Forecasting Groups of Time

Series: Locality and Globality," *International Journal of Forecasting*, vol. Pre-Print, Aug 2020.
[68] L. Zhu and N. Laptev, "Deep and Confident Prediction for Time Series at Uber," in *IEEE International Conference on Data Mining Workshops, ICDMW*, pp. 103–110, IEEE Computer Society, Dec 2017.
[69] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.
[70] P. Montero-Manso, G. Athanasopoulos, R. J. Hyndman, and T. S. Talagala, "FFORMA: Feature-based forecast model averaging," *International Journal of Forecasting*, vol. 36, pp. 86–92, Jan 2020.
[71] Q. Hu, R. Zhang, and Y. Zhou, "Transfer learning for short-term wind speed prediction with deep neural networks," *Renewable Energy*, vol. 85, pp. 83–95, 2016.
[72] A. S. Qureshi and A. Khan, "Adaptive transfer learning in deep neural networks: Wind power prediction using knowledge transfer from region to region and between different task domains," *Computational Intelligence*, vol. 35, no. 4, pp. 1088–1112, 2019.
[73] A. S. Qureshi, A. Khan, A. Zameer, and A. Usman, "Wind power prediction using deep neural network based meta regression and transfer learning," *Applied Soft Computing*, vol. 58, pp. 742–755, 2017.
[74] S. Zhou, L. Zhou, M. Mao, and X. Xi, "Transfer learning for photovoltaic power forecasting with long short-term memory neural network," in *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 125–132, IEEE, 2020.
[75] H. Zang, L. Cheng, T. Ding, K. W. Cheung, Z. Liang, Z. Wei, and G. Sun, "Hybrid method for short-term photovoltaic power forecasting based on deep convolutional neural network," *IET Generation, Transmission & Distribution*, vol. 12, no. 20, pp. 4557–4567, 2018.
[76] Y. Zhang and G. Luo, "Short term power load prediction with knowledge transfer," *Information Systems*, vol. 53, pp. 161–169, 2015.
[77] P. Zeng, D. Wu, and M. Jin, "Compress-filtering and transfer-expanding of data set for short-term load forecasting," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1095–1101, IEEE, 2017.
[78] P. Zeng, C. Sheng, and M. Jin, "A learning framework based on weighted knowledge transfer for holiday load forecasting," *Journal of Modern Power Systems and Clean Energy*, vol. 7, no. 2, pp. 329–339, 2019.
[79] X. Xu and Z. Meng, "A hybrid transfer learning model for short-term electric load forecasting," *Electrical Engineering*, vol. 102, no. 3, pp. 1371–1381, 2020.
[80] S.-M. Jung, S. Park, S.-W. Jung, and E. Hwang, "Monthly electric load forecasting using transfer learning for smart cities," *Sustainability*, vol. 12, no. 16, p. 6364, 2020.
[81] M. Ribeiro, K. Grolinger, H. F. ElYamany, W. A. Higashino, and M. A. Capretz, "Transfer learning with seasonal and trend adjustment for cross-building energy forecasting," *Energy and Buildings*, vol. 165, pp. 352–363, 2018.
[82] C. Fan, Y. Sun, F. Xiao, J. Ma, D. Lee, J. Wang, and Y. C. Tseng, "Statistical investigations of transfer learning-based methodology for short-term building energy predictions," *Applied Energy*, vol. 262, p. 114499, 2020.
[83] A. Li, F. Xiao, C. Fan, and M. Hu, "Development of an ann-based building energy model for information-poor buildings using transfer learning," in *Building Simulation*, vol. 14, pp. 89–101, Springer, 2021.
[84] N. Laptev, J. Yu, and R. Rajagopal, "Applied timeseries transfer learning," in *Sixth International Conference on Learning Representations (ICLR)*, 2018.
[85] M. Jain, K. Gupta, A. Sathanur, V. Chandan, and M. M. Halappanavar, "Transfer-learnt models for predicting electricity consumption in buildings with limited and sparse field data," in *2021 American Control Conference (ACC)*, pp. 2887–2894, 2021.
[86] E. Lee and W. Rhee, "Individualized short-term electric load forecasting with deep neural network based transfer learning and meta learning," *IEEE Access*, vol. 9, pp. 15413–15425, 2021.
[87] N. Laptev, J. Yu, and R. Rajagopal, "Reconstruction and regression loss for time-series transfer learning," in *Proc. SIGKDD MiLeTS*, 2018.
[88] D. Wu, B. Wang, D. Precup, and B. Boulet, "Boosting based multiple kernel learning and transfer regression for electricity load forecasting," in *Joint European Conference on Machine Learning and*

*Knowledge Discovery in Databases*, pp. 39–51, Springer, 2017.
[89] D. Wu, B. Wang, D. Precup, and B. Boulet, "Multiple kernel learning-based transfer regression for electric load forecasting," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1183–1192, 2019.
[90] J.-B. Fiot and F. Dinuzzo, "Electricity demand forecasting by multi-task learning," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 544–551, 2016.
[91] Y. Zhang, G. Luo, and F. Pu, "Power load forecasting based on multi-task gaussian process," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 3651–3656, 2014.
[92] L. Cai, H. Wen, J. Gu, J. Ma, and Z. Jin, "Forecasting customers' response to incentives during peak periods: A transfer learning approach," *International Transactions on Electrical Energy Systems*, vol. 30, no. 7, p. e12251, 2020.
[93] K. Bandara, H. Hewamalage, Y.-H. Liu, Y. Kang, and C. Bergmeir, "Improving the accuracy of global forecasting models using time series data augmentation," *arXiv preprint arXiv:2008.02663*, 2020.
[94] Y. Tian, L. Sehovac, and K. Grolinger, "Similarity-based chained transfer learning for energy forecasting with big data," *IEEE Access*, vol. 7, pp. 139895–139908, 2019.
[95] Y. Gao, Y. Ruan, C. Fang, and S. Yin, "Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data," *Energy and Buildings*, vol. 223, p. 110156, 2020.
[96] A. Hooshmand and R. Sharma, "Energy predictive models with limited data using transfer learning," in *Proceedings of the Tenth ACM International Conference on Future Energy Systems*, pp. 12–16, 2019.
[97] D. Zhou, S. Ma, J. Hao, D. Han, D. Huang, S. Yan, and T. Li, "An electricity load forecasting model for integrated energy system based on bigan and transfer learning," *Energy Reports*, 2020.
[98] S. Gomez-Rosero, M. A. Capretz, and S. Mir, "Transfer learning by similarity centred architecture evolution for multiple residential load forecasting," *Smart Cities*, vol. 4, no. 1, pp. 217–240, 2021.
[99] T. Le, M. T. Vo, T. Kieu, E. Hwang, S. Rho, and S. W. Baik, "Multiple electric energy consumption forecasting using a cluster-based strategy for transfer learning in smart building," *Sensors*, vol. 20, no. 9, p. 2668, 2020.
[100] Y. Yang, W. Hong, and S. Li, "Deep ensemble learning based probabilistic load forecasting in smart grids," *Energy*, vol. 189, p. 116324, 2019.
[101] Y. Yang, W. Li, T. A. Gulliver, and S. Li, "Bayesian deep learning-based probabilistic load forecasting in smart grids," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 7, pp. 4703–4713, 2019.
[102] T. Grubinger, G. C. Chasparis, and T. Natschläger, "Generalized online transfer learning for climate control in residential buildings," *Energy and Buildings*, vol. 139, pp. 63–71, 2017.
[103] Y. Chen, Y. Zheng, and H. Samuelson, "Fast adaptation of thermal dynamics model for predictive control of hvac and natural ventilation using transfer learning with deep neural networks," in *2020 American Control Conference (ACC)*, pp. 2345–2350, IEEE, 2020.
[104] M. Jain, M. Gupta, A. Singh, and V. Chandan, "Beyond control: Enabling smart thermostats for leakage detection," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–21, 2019.
[105] D. L. Cavalca and R. A. Fernandes, "Deep transfer learning-based feature extraction: An approach to improve nonintrusive load monitoring," *IEEE Access*, 2021.
[106] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[107] S. Gunduz, U. Ugurlu, and I. Oksuz, "Transfer learning for electricity price forecasting," *arXiv preprint arXiv:2007.03762*, 2020.
[108] J. Lago, F. De Ridder, P. Vrancx, and B. De Schutter, "Forecasting day-ahead electricity prices in Europe: the importance of considering market integration," *Applied Energy*, vol. 211, pp. 890–903, Aug 2017.
[109] E. Mocanu, P. H. Nguyen, W. L. Kling, and M. Gibescu, "Unsupervised energy prediction in a smart grid context using reinforcement cross-building transfer learning," *Energy and Buildings*, vol. 116, pp. 646–655, 2016.

[110] D. Kong, X. Kong, J. Xiao, J. Zhang, S. Li, and L. Yue, "Dynamic pricing of demand response based on elasticity transfer and reinforcement learning," in *2019 22nd International Conference on Electrical Machines and Systems (ICEMS)*, pp. 1–5, IEEE, 2019.

[111] S. Liu and G. P. Henze, "Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis," *Energy and buildings*, vol. 38, no. 2, pp. 148–161, 2006.

[112] H. Kazmi, D. Didden, N. Wiese, and J. Driesen, "Sample efficient reinforcement learning with domain randomization for automated demand response in low-voltage grids," *IEEE Journal of Emerging and Selected Topics in Industrial Electronics*, 2021.

[113] Y. Tao, J. Qiu, and S. Lai, "A hybrid cloud and edge control strategy for demand responses using deep reinforcement learning and transfer learning," *IEEE Transactions on Cloud Computing*, 2021.

[114] W. Zhang, L. Deng, L. Zhang, and D. Wu, "Overcoming negative transfer: A survey," *arXiv preprint arXiv:2009.00909*, 2020.

[115] P. Banda, M. A. Bhuiyan, K. Zhang, and A. Song, "Transfer learning for leisure centre energy consumption prediction," in *International Conference on Computational Science*, pp. 112–123, Springer, 2019.