Applied Energy 238 (2019) 1022–1035

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads

Hussain Kazmi[a,b,*], Johan Suykens[b], Attila Balint[a], Johan Driesen[b]

[a] Enervalis, Belgium
[b] Dept. of Electrical Engineering, KU Leuven, Belgium

HIGHLIGHTS

- A multi-agent reinforcement learning framework for black-box modelling is presented.
- Agency can be used interchangeably with increased sensing or domain knowledge.
- The framework accelerates modelling performance linearly with increasing agents.
- Multi-agent systems learn faster and better than a comparable single agent system.
- Efficiency gains of 20% were realized in a real world pilot running for a year.

ARTICLE INFO

Keywords:
Multi agent reinforcement learning
Distributed learning
Optimal control
Thermostatically controlled loads
Domestic hot water storage vessel
Heat pumps

ABSTRACT

Increasing energy efficiency of thermostatically controlled loads has the potential to substantially reduce domestic energy demand. However, optimizing the efficiency of thermostatically controlled loads requires either an existing model or detailed data from sensors to learn it online. Often, neither is practical because of real-world constraints. In this paper, we demonstrate that this problem can benefit greatly from multi-agent learning and collaboration. Starting with no thermostatically controlled load specific information, the multi-agent modelling and control framework is evaluated over an entire year of operation in a large scale pilot in The Netherlands, constituting over 50 houses, resulting in energy savings of almost 200 kW h per household (or 20% of the energy required for hot water production). Theoretically, these savings can be even higher, a result also validated using simulations. In these experiments, model accuracy in the multi-agent frameworks scales linearly with the number of agents and provides compelling evidence for increased agency as an alternative to additional sensing, domain knowledge or data gathering time. In fact, multi-agent systems can accelerate learning of a thermostatically controlled load's behaviour by multiple orders of magnitude over single-agent systems, enabling active control faster. These findings hold even when learning is carried out in a distributed manner to address privacy issues arising from multi-agent cooperation.

1. Introduction

Thermostatically controlled loads (TCLs) represent a substantial portion of energy consumption in most industrialized economies [1]. Most frequently, these reflect either space or water heating (or cooling) and are controlled by rule-based decision systems which maintain temperatures within a dead-band using hysteresis [2]. These rule based control strategies are often demonstrably sub-optimal. Better, more sophisticated control strategies can not only improve energy efficiency of TCLs [3] but also use the energy flexibility inherent in TCLs to provide ancillary services to the grid [4]: these services include demand response [5], frequency or voltage regulation [6], and self-consumption of local renewable generation [7].

Historically, optimal control strategies for TCLs have relied on a model of the system to be optimized. A number of possible alternatives, ranging from black-, grey- and white-box models, have been presented in the literature [8]. These represent increasing levels of involvement of a human domain expert who, with his knowledge or experience, creates a model for the TCL to be controlled. This model is, in turn, used to plan and execute optimal policies for the TCL. Black box models, in learning directly from data, are intended to circumvent this reliance on costly human expertise [9]. An example of data-driven control of TCLs can be

[*] Corresponding author.
E-mail address: hussainsyed.kazmi@kuleuven.be (H. Kazmi).

https://doi.org/10.1016/j.apenergy.2019.01.140
Received 22 June 2018; Received in revised form 13 January 2019; Accepted 19 January 2019

found in [10], showing efficiency gains of 7% for the heating, ventilation and air conditioning in an office building. However, in most practical systems, an expensive human domain expert is traded off for equally (or possibly even more) expensive sensing technologies. These sensors are vital in data acquisition required by black box models but come with their own installation and maintenance costs. Furthermore, in most practical situations, the amount of data required by black-box systems to learn a reliable model of the TCL can be substantial. This leads to additional delays before the devices can be actively controlled.

Contemporary developments in model-free control algorithms have shown that it is possible to translate sensor observations directly into control actions. However, these methods rely just as much, if not more, on adequate sensor data for reliable system identification or state estimation [11]. Additionally, classical RL algorithms like Q-learning, employed by [12,13], are extremely data-inefficient i.e. require much more data to attain the same performance level as their model-based counterparts [14]. More advanced model-free RL controllers, such as the one used in [15], improve on the data-efficiency of classical RL algorithms but still do not attain the level of their model-based counterparts [16]. Finally, the use of model-free control rules out the use of a developed model in diagnostic settings, for instance for fault detection or as part of a prescriptive analysis.

Residential TCLs have remained largely untapped as a demand-side technology because of the expensive modelling requirements highlighted above. Another complicating factor is the complexity of distributed control: thousands of devices need to be operated in real-time, simultaneously optimizing for both household (preserving thermal comfort) and market objectives (optimizing for energy or costs, etc.).

2. State of the art and contributions

In this paper, we posit that the trade-off facing TCL modelling is not just between sensing and prior knowledge. Rather, agency provides an additional dimension to facilitate and even accelerate learning. We consider agency as a measure of both the quality and the quantity of collaborative agents. To test this hypothesis, we begin with identifying the set of conditions under which agency can be useful. The setting (and data) for this was provided by a large scale European Horizon 2020 project REnnovates[1] where hundreds of Dutch houses were renovated to net-zero energy status. All the houses were equipped with identical heat pumps and hot water vessels. This replication of TCLs is by no means atypical and is often the case in social housing schemes across Europe where housing corporations replicate the same building and equipment design to entire neighbourhoods, relying on economies of scale to reduce costs.

The unique factor differentiating TCLs installed in different households is their usage: building occupants interact with even identical TCLs in different ways. This means that identical devices will operate in very different circumstances in different households because of occupant influence. This has been shown for hot water systems [17] and the thermal behaviour of entire buildings [18]. Considered together, this corresponds to a far wider exploration of the state space and can lead to a much richer representation of the device than can be obtained by considering just a single TCL installed in a household. A better model learned in this way potentially allows for improved control policies.

The focus of this paper is on hot water systems because learning a dynamics model for both the storage vessel and the heating element is a challenging problem. Previous researchers have frequently resorted to learning offline models for the storage vessel [19] or installing additional sensors in the storage vessel for online learning [20]. In practice, most residential hot water vessels are equipped with a solitary temperature sensor which, as we will demonstrate, is insufficient for online learning of an accurate dynamics model.

This paper also extends earlier reported work on model-based optimization of TCLs [21] in several ways. First, it proposes and demonstrates the use of multiple agents as a practical alternative to increasing domain knowledge or sensing information during the modelling of TCLs. It also quantifies and explores ways to minimize the temporal cost of such modelling, i.e. the amount of time (or data) required to learn a data-driven model online. This cost is frequently brushed under the carpet, but is arguably one of the most important factors in black-box control. Furthermore, the paper benchmarks the learned model using a simulated vessel, and also investigates the reason for the efficiency gains over an entire year in much greater detail in tens of real-world households. Other model-based approaches include [22,23] which have focused more on using ensembles of TCLs to provide services to the grid. In doing so, the actual modelling of the TCL is simply considered as a cyclical load which has to be modulated appropriately to provide grid services. While the proposed methodology can be used to provide grid services as well, the focus is instead to learn an accurate dynamics model for the TCL in as little time as possible, and use it for optimizing efficiency. The control part of the formulation draws some parallels with previous approaches as fundamentally it is the frequency of the reheat cycles which is being modulated here too. However, the emphasis of this research is on detailed bottom-up optimization (which focuses on household behaviour) rather than aggregated response of an entire cluster of TCLs to grid requirements.

The primary contribution of the paper thus lies in demonstrating the value of multi-agent systems which can quickly optimize TCL performance. Collaborative multi-agent decision making has been extensively used to demonstrate improvements in cooperative tasks [24]. This also holds for smart grids concepts [25] such as peak shaving and frequency response for both centralized [26] and distributed decision making systems [27]. It is, however, not obvious how local control problems such as energy efficiency can benefit from multi-agent formulations and has not been addressed in existing literature. This paper uses a novel multi-agent formulation for modelling of TCLs which circumvents typical issues with black- and gray-box modelling techniques as mentioned previously. We show that multiple agents can reduce the data gathering period necessary for training comparable single-agent black- and grey-box models by at least an order of magnitude. This is, in most practical instances, the difference between optimal and sub-optimal control. Furthermore, additional agency can serve as an alternative to increased sensing or domain knowledge in the problem formulation. This further alleviates concerns typically associated with detailed modelling of TCLs.

Another contribution of the paper is its handling of privacy issues. These are a ubiquitous concern for systems recording consumer data directly [28]. These issues have also been explored in the wider context of smart grids where data collection is a necessary prerequisite to unlocking many innovative value streams [29]. Extending to multiple agents can further worsen this situation. We show that data-driven learning using the proposed framework does not necessarily have to raise (and can even alleviate) privacy concerns by making use of a distributed learning framework, loosely inspired by the parameter server approach [30]. Such a formulation also successfully reduces communication overhead costs - an important issue in modern smart grids.

The rest of this paper is organized as follows: we begin by presenting the proposed methodology, followed by results obtained using a simulation framework as well as the real world case study.

3. Methodology

3.1. Problem formulation

The problem of hot water system optimization can be seen as an *N*-player finite, non-zero sum game of hidden information [31]. *N*-player refers to the fact that an individual agent operates in a single household

[1] www.rennovates.eu.

but N such agents operate simultaneously in different households to optimize their respective rewards. Non-zero sum illustrates that, for a local problem such as energy efficiency, an agents strategy does not directly affect other agents or their rewards. In fact, cooperation can further increase individual rewards. Hidden information alludes to the partial observability of the hot water system. In most real world hot water systems, sensing for state estimation is limited to a single temperature sensor, often mounted at the mid-way point [21]. Conventionally, hidden observation refers to the setting where different agents are oblivious of other agents' states, however we primarily consider the case where agents are aware of other agents' states. In the section on distributed control, we consider a relaxation of this condition.

In reinforcement learning terminology, the N-player and hidden information aspects of the problem lead to the formulation of a multi-agent POMDP (Partially observable Markov Decision Process) [32]. By learning a dynamics model of the system, it is possible to sidestep the hidden information problem and formulate an MDP as: $M = \{S, A, P, R\}_n$ [33]. $\{S, A, P, R\}_n$ represent the tuple of state, action, transition function and reward stream respectively and the n identifies that while the structure of the MDP is the same for each agent, the individual components can be different (e.g. while the state-space is shared across all agents, instantaneous states are not).

The structure of the MDP derives from the interactions between a hot water storage vessel, a heating element and the human user. The reinforcement learning (RL) agent controls the heating element to reheat the vessel following a policy, π_n, designed to minimize energy consumption while maintaining occupant comfort.

3.1.1. State

The state, $s_t \in S = \{0, 1, \ldots, 100\}$ is defined as the state of charge of the hot water vessel at time t. This state of charge is not directly observable because only a single mid-point temperature sensor (T_m) is available by default in most residential hot water vessels. This value, by itself, is not enough to observe the state of charge of the vessel which is a non-linear function as shown in Fig. 1. Therefore, this temperature distribution in the storage vessel is first derived from observed sensor data as a function of user hot water consumption over time, $w_{t,n}$, agent control actions (reheat cycles), $a_{t,n}$ and the previous observed temperatures in the vessel, $T_{m_{t,n}}$, using supervised learning:

$$s_t = f(T_{m_{t-\tau:t}}, a_{t-\tau:t}, w_{t-\tau:t}) \tag{1}$$

s_t, the instantaneous state of charge, is then defined as the ratio between the estimated hot water left in the vessel above a certain temperature threshold and the vessel capacity. The threshold is, in practice, usually defined around 45 °C.

3.1.2. Action

The agent's action, $a_t \in A = \{0, 1\}$, is the control of the heating element. An action of 1 means the initiation of a reheat cycle while a control value of 0 implies the heating element is inactive. When the heating element is turned on, it consumes some energy to reheat the storage vessel from some initial state of charge to a final state of charge.

3.1.3. Transition function

The agent's actions are influenced by the learned transition function, which takes the form:

$$s_{t+1,n} = P_n(s_{t,n}, a_{t,n}, w_{t,n}) \tag{2}$$

where, the transition function, $P_n(\cdot)$, defines the probability distribution over the next state of the vessel as a function of current state $s_{t,n}$, the agent's action $a_{t,n}$ and occupant behavior, i.e. instantaneous water consumption $w_{t,n}$. This function is a further composition of two functions. In the first, the vessel temperature distribution is calculated for the next time step. In the second, the threshold is applied to this distribution to calculate the current state of charge.

3.1.4. Reward

The reward $r_{t+1,n}$ is a function of the final state, $s_{t+1,n}$ and action $a_{t,n}$, where the former derives from the energy consumed and the latter from impact on occupant comfort:

$$r_t = r_t^e + r_t^c \tag{3}$$

where

$$r_t^e = \begin{array}{cl} -B & \text{if } a_t = 0 \\ -f(s_t) & \text{otherwise} \end{array} \tag{4}$$

and

$$r_t^c = \begin{array}{cl} 0 & \text{if } s_{t+1} > s_{th} \\ -A.\ \mathbf{E}[f(s_t)] & \text{otherwise} \end{array} \tag{5}$$

where $f(s_t)$ maps the energy consumed to reheat the vessel given an initial state of charge and is learned from observation data using supervised learning. $-B$ is a small negative reward given to the agent for every time step. $A \gg 1$ is the value placed on the lost load, to incentivize the agent to prioritize human comfort over energy efficiency gains while s_{th} is a safety margin to ensure that the user always has sufficient hot water, even in case of unexpected draws. $\mathbf{E}[f(s_t)]$ is the expected value of the energy consumed to reheat the storage vessel from any state of charge and is a proxy for the lost load in the absence of any feedback mechanism from the user. The motivation to use this value is to strongly penalize the agent when its actions lead to lost user comfort. Where occupant feedback is present, other metrics such as the expected waiting time until hot water availability can be used in the reward function. Finally, the control actions are assumed to be non-

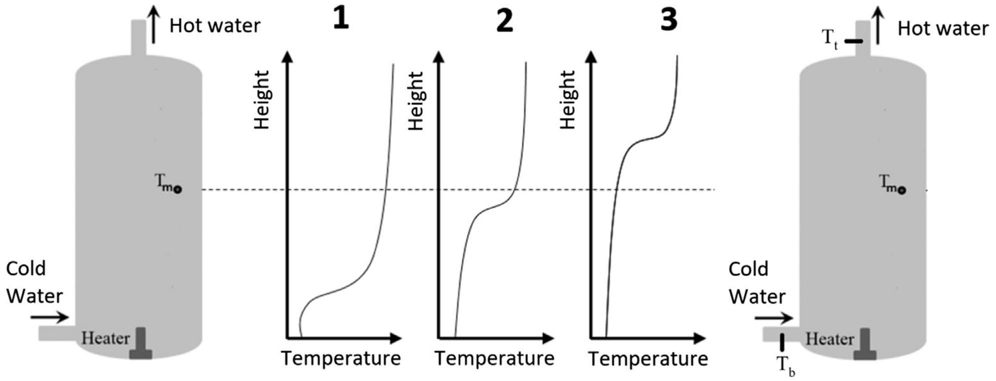

Fig. 1. Default (left) and additional (right) sensor placement in the vessel; also shown are some representative temperature distributions in the vessel illustrating why a single sensor offers only incomplete sensing; adopted from [19].

interruptible, i.e. once reheating commences, the vessel's state is reset. In this way, the problem takes on an episodic nature.

3.1.5. Learning the transition and reward functions

Learning the temperature profile as a function of the storage vessel, occupant behavior and ambient conditions is central to this problem. Likewise, an accurate model of the heating element allows the agent to estimate its future rewards and improve planning.

Feature extraction. Before a supervised learning problem can be posed, features can be extracted from the observed time series. There are two reasons to do this. First, reducing the dimensionality of the input feature vector simplifies the learning problem. Second, interpretable features bring structure to the learning problem. More concretely, in this research, a number of features are extracted from observed sensor data. These basically make use of the episodic nature of the task in which the state of each agent can be assumed to be reset periodically, i.e. with a reheat cycle [33]. Another alternative is to use an automated feature extraction algorithm like principle component analysis or autoencoders [15], however the features extracted by these algorithms are seldom interpretable. The extracted features used in this research include: (1) initial temperature in the vessel (as measured by the sensor) after a reheat cycle, (2) cumulative hot water consumption since the last reheat cycle, (3) time elapsed since the last reheat cycle, and (4) ambient temperature conditions during the reheat cycle. Of these, the first feature provides initial conditions, while the next two identify the effect of occupant behavior and thermodynamic losses on the storage vessel's state of charge respectively. These features also influence the heating element's energy consumption. Additionally, ambient conditions also affect the energy consumption in case a heating element like heat pump is employed. A large number of other features were also investigated however they did not lead to a substantial improvement in learning performance, and were consequently discarded.

It is important to point out what is being learned here. The supervised learning model for the storage takes as input the three features described above and predicts the temperature as it would have been observed at the sensor. By varying the three variables, an entire temperature profile inside the storage vessel can be constructed, as depicted in Fig. 1, which can yield the instantaneous state of charge. More concretely, keeping the initial temperature and elapsed time fixed at the current observation state, the water consumption variable is increased until the (estimated) temperature at the outflow of the vessel falls below a specified threshold. This estimate is derived from the learned vessel model and the amount of hot water divided by the vessel capacity is the current state of charge of the vessel. Similarly, the model for the heating element is only used when a reheat event takes place. The heating model, in this case, simply predicts the amount of energy in kWh's that would be required to reheat the storage vessel to a final temperature at the mid-point (or wherever the sensor is installed).

While feature extraction from raw time series data helps with the learning process, other avenues also exist which may speed it up. These include the following:

Sensory information (I). This is the standard form of black-box modelling, whereby observed sensor data is used to learn a mapping from input features to output states. The default configuration that we consider (and that is prevalent in most residential systems) is a single temperature sensor providing this sensory input. This, however, is not enough to generalize to the entire storage vessel as the temperature distribution is non-uniform because of stratification effects and other nonlinear dynamics [19]. Additional sensors, placed at the top, bottom or elsewhere in the vessel, can provide further information which facilitates learning a model for the temperature distribution. This is done by translating these additional sensor measurements into additional features which are then used to train the model. The additional sensor configuration is visualized in Fig. 1. However, this additional sensing comes at increased installation and maintenance cost. This configuration is represented by (I) in subsequent sections.

Domain knowledge (K) This is the conventional alternative to more sensing, whereby expert knowledge is somehow integrated with the black box system to facilitate or accelerate learning. Frequently this takes the form of online calibration of a pre-built thermodynamics model, constituting a grey box system. However, in this paper we do not consider such systems because of their lack of generalizability to new systems. Instead, we consider domain knowledge in black-box settings in the form of data augmentation using insights from general thermodynamics laws which take the following form:

- Temperature is assumed to monotonically increase with vessel height; thus $T_h \leqslant T_{h+x}$ where $x \geqslant 0$.
- Water temperature in the vessel is always bounded between $0 < T_x < 100$ [°C] $\forall\, x$.
- Even stricter bounds can be introduced by observing that a residential water vessel invariably operates between $10 \leqslant T_x \leqslant 65$ [°C] $\forall\, x$.
- $T_x \to T_b \; \forall\, x$ as w (the cumulative water consumption) $\to \infty$ (without reheating the vessel), where T_b is the lower bound as defined above.

These constraints are included as additional training features for the supervised learning process. In the absence of these constraints, the temperature distribution values can oscillate wildly outside of the observed feature vectors. This configuration is represented by (K) in subsequent sections and constitutes a weak form of general domain knowledge. It is to be noted that none of these constraints is vessel-specific and therefore remains generalizable to new vessels.

Agency (MARL/SARL). The *N*-agents, across whom the MDP is replicated, all act independently in learning their own transition functions to plan accordingly. Here MARL and SARL identify multi-agent and single-agent reinforcement learners respectively. Given different human behavior in all these buildings, the agents are driven to different regions of the state-space. This means in general:

$$P(\mathbf{s}_i, \mathbf{a}_i, \mathbf{w}_i) \neq P(\mathbf{s}_n, \mathbf{a}_n, \mathbf{w}_n), \ \ \forall\, n \neq i \tag{6}$$

where $\mathbf{s}_i$, $\mathbf{a}_i$, $\mathbf{w}_i$ represents the tuple of historic observations for agent i. However, since the storage vessel has fundamentally the same characteristics (i.e. is identical across all households), the differences in learned transition function only arise because of observation bias. By aggregating features collected by individual agents, it is possible to learn a single transition function of the form:

$$P = f(\mathbf{s}_1, \ldots, \mathbf{s}_N, \mathbf{a}_1, \ldots, \mathbf{a}_N, \mathbf{w}_1, \ldots, \mathbf{w}_N) \tag{7}$$

In general, this unified model should outperform each individual transition function because of improved state-space exploration. This also paves the way for cooperation between individual agents. This can be achieved by targeted exploration to drive different agents to regions of the state-space where the learned representations uncertainty is still high. Uncertainty estimates can be derived using either ensemble [34] or Bayesian methods [35], depending on the choice of function approximation method. For (deep) neural networks, a recent approach has been the use of dropout as a practical approximation for model uncertainty [36]. In our work, we have experimented with both random forests and neural networks obtaining similar results. Furthermore, by extracting features which decouple thermodynamic losses from consumption based losses, stochastic human behavior can be disentangled from deterministic storage vessel behavior.

3.2. Observation period

The performance of data-driven methods depends both on the quality and quantity of the data used to train the model. In addition to the sources of data mentioned previously, the observation time period also plays a large role in determining model performance. This is to say that given sufficient learning capacity, a model trained with one year of observational data will most likely outperform a model trained with only one week of data. Increasing training data is the simplest way to improve model performance, this remains true regardless of the source of the additional data. However, the increased monitoring period takes away from the time available to the controller to optimize system performance.

3.3. Evaluating model performance

Numerous metrics are available to evaluate model performance. In this research, we focus on two simple evaluation mechanisms: mean absolute error (MAE) and the coefficient of determination (R^2). Both of these metrics are aimed at evaluating how well the learned model can predict the system's present and future states. Furthermore, we also incorporate the temporal aspect of learning into the evaluation criterion, because a modelling approach requiring prohibitive amounts of data to perform well is impractical in the real world. This leads us to define three characteristics: (1) initial model performance, (2) learning rate, and (3) asymptotic model performance.

3.4. Privacy aware learning

It is possible for each agent to learn the transition function using its own experiences locally. If no data is transferred off-site, this does not pose any privacy challenges and represents the SARL condition as defined above. However, once cooperation between multiple agents is considered as described in Eq. (7), additional concerns arise. Primarily, a malicious agent with access to every agents data can estimate both occupancy patterns as well as draw inferences about demographics. This can be addressed using two possible formulations:

1. *Feature sharing*, whereby each individual agent only shares extracted features, rather than raw consumption data, with a central oracle where learning takes place. The central oracle then communicates the learned model to each individual agent. By scrambling the order of the time series through feature extraction, occupancy patterns can not be directly observed by a malicious party. This form of learning is directly compatible with the already presented formulation.
2. *Parameter sharing*, a form of distributed learning, goes one step further and only shares the parameters of individual models learned by each agent with the oracle. These parameters can, for instance, be the weights of a neural network. Here, the transition function takes the form:

   $$P = f(P_1, \ldots, P_N) \tag{8}$$

   By combining N individual models, the overall performance can be improved by ensembling through a mixture of experts approach, while preserving privacy [37,38]. It has been shown that this approach works even when only a tiny fraction of all parameters are shared [39], which assuages both privacy concerns and reduces network communication costs.

From a privacy perspective, parameter sharing offers even stronger protection against malicious parties than direct feature sharing which is, in itself, an improvement over sharing raw time series data. The same holds for network communication costs.

3.5. From modeling to control

The default policy in the considered systems is a naive rule based controller with a dead-band hysteresis of the form:

$$a_{t,i} = \begin{cases} 1, & \text{if } T_s < T_{th} - \Delta T \\ 0, & \quad \text{if } T_s \geqslant T_{th} \end{cases} \tag{9}$$

This rule-based controller forms the default baseline. However, system efficiency can be improved by only reheating the vessel when required [40]. This makes use of the state of charge of the vessel which is derived from the learned vessel model as described above. The agent then maximizes its future reward stream in time according to the learned dynamics model, given its current state:

$$\max \mathrm{J}(\pi|\mathrm{s}_0) = 1/\tau \sum_{0}^{\tau-1} \mathrm{r}_\mathrm{t} \tag{10}$$

The reward function, as defined earlier, includes both occupant comfort and energy efficiency. The optimal policy can be learned through policy search [41,42] or derivative free optimization methods [17]. In this paper, we make use of a Monte Carlo based reinforcement learning algorithm (Monte Carlo with Exploring Starts - MCES) [33]. MCES exploits the episodic nature of the task and makes use of the learned model and observed occupant behaviour to simulate many trajectories (episodes) into the future. Each episode terminates with a reward, as defined earlier, which is then assigned to every state and action pair visited during the episode. By averaging the rewards obtained over a large number of episodes, the true expected rewards from that state and action pair are estimated. Thereafter, starting from any given state, an ϵ-greedy policy can be implemented which chooses the action corresponding to the highest expected reward with probability $1 - \epsilon$ and a random action otherwise. An advantage of using this controller, instead of a classical model predictive controller, is that it is much less compute intensive and can handle nonlinear objective functions or constraints in a more seamless manner. Foresee, a recent framework for optimizing building energy consumption and provision of demand response, is one example of a model predictive controller [43]. This is primarily due to the policy-side learning which helps avoid the need to repeatedly optimize for the same, or similar, conditions. However, there should be no discernible benefit in the control performance achieved by this controller as opposed to a more traditional controller also using the same learned model.

Using a controller such as MCES is only necessary for the general case of time-variant energy tariffs or efficiency (such as with heat pumps). In this case, while the model for each household sharing an identical TCL will be the same, the optimal policy will be different for each household. With tens or hundreds of households executing individual policies, this can be difficult to keep track of, or debug when some things goes wrong. For the more limited case of fixed electricity prices and time-invariant efficiency, a rule based controller with the learned system dynamics is guaranteed to be optimal if state of charge and future consumption can be accurately estimated/ predicted. This controller takes the following form:

$$a_t = \begin{cases} 1, & \text{if } s_t < \hat{s}_{t+1} \\ 0, & \text{if } s_t \geqslant \hat{s}_{t+1} \end{cases} \tag{11}$$

In this case, the agent only reheats the vessel when the predicted required state of charge in the subsequent time period ($\hat{s}_{t+1}$) is expected to exceed the (estimated) current state of charge, s_t. These are derived from the learned representation of temperature distribution in the storage vessel. While sub-optimal for the case of time-variant prices or efficiency, an additional advantage of using this controller is that it simplifies control by removing the dependence on the heating element model.

4. Experimental setup

In the modelling and control of a TCL, multiple factors have to be considered. These include the storage vessel and the heating element, human occupant behaviour and ambient conditions. We consider two case studies to investigate the performance of our algorithm on different variations of these factors. One is based on a simulated hot water system and the other presents results of a real world large scale pilot involving tens of net-zero energy buildings in the Netherlands.

4.1. Simulation based case study

The first case study makes use of a simulation framework employing a simulated hot water storage vessel and an ideal heating element. The duration for this case study was one year, and four households were considered. Each of the households had an identical TCL setup, i.e. the storage vessel and heating element models were identical in all households. The consumption patterns were however different for each household and were drawn from real world data.

Each agent corresponded to an individual household, which meant the system state for every agent was unique at any given time step. The storage vessel model was simulated using a logistic regression based model fit to empirical data [17]. The heating element was considered to be an ideal electric heater with an efficiency of 100%. The agents were required to learn the vessel model from the gathered data but the heating element model was assumed to be known to all the agents. This simplified both the learning and control problems.

To explore the effect of increased sensing on learning performance, we considered the case where additional temperature sensors were placed in the storage vessel. Finally, as the data was simulated using a known model, the learned model accuracy could be estimated for the entirety of the temperature distribution. To do this, the temperature as predicted by the model was compared with ground truth for randomly drawn input feature vectors.

4.2. Real world case study

The simulated case study was designed to mirror the real world situation and therefore test the same hypothesis. Nevertheless, there were some important differences. Most notably, unlike the simulation based study, the real world case study involved many more houses. Most notably, data from 53 households was analysed for a year. Of these, up to 32 randomly chosen houses were used to train the storage vessel and heating models while the remaining were used for validation purposes. The actual storage vessel and heating models were not known for the real world study which meant the accuracy of the learned models could only be calculated through this held-out test dataset of 21 households.

All the households considered in this study employed identical TCLs (i.e. storage vessel and heating element) so the framework could be applied in a straightforward manner. More specifically, the storage vessel was a modern 200 L vessel with a single temperature sensor mounted at the midway point and a water flow meter. The heating element was an air source heat pump (ASHP) which provided both the hot water and space heating to the buildings. The households were all recently refurbished social houses. This situation is quite common in The Netherlands where social housing makes for a large proportion of all residential buildings, ensuring a large degree of homogeneity. For standardization and replication purposes, the EEBus protocol[2], an open standard for the internet of things, was employed to communicate between the central server and the installed heat pumps. Finally, in The Netherlands, net-zero energy buildings are legally obliged to prove that their annual energy demand equals production (via rooftop solar panels). This necessitates a system to monitor energy flows irrespective of system optimization, meaning that the smart control framework can make use of the already present sensing and communication infrastructure.

One final complication that we encountered with learning models in the real world case study was with additional sensing. While in the simulation, we could simply sample the entire distribution by placing virtual sensors, in the real world additional sensors had to be physically installed. These sensors were placed in a noninvasive manner, which ensured that the extra effort required to install the sensors was replicable and not excessive. However, this meant that the sensors recorded temperature values only when hot water was being consumed, thereby severely limiting the amount of usable data they generated.

Active control was interleaved with learning. More specifically, after an initial period of one month, control was executed using the learned models. This furnished results of active control for 11 months. More practically, this meant sending reheat commands to each heat pump being controlled when the central controller decided it was time to reheat the storage vessel. While the control was executed in a centralized manner to keep implementation straightforward, it can be also be performed in a distributed manner. The 53 houses under consideration were split into three groups for control purposes: a default group, an energy efficient group and a third group. This third group was used by another project partner to test different algorithms including solar self-consumption and grid congestion reduction. The split was not fixed however, and the houses were cycled into different groups weekly to minimize the effect of occupant behaviour. In this way, any households demonstrating anomalous behaviour were present in every group over the course of the year. In the remainder of this paper, we will focus exclusively on the first two groups, i.e. the houses running the default controller and those running the proposed efficient controller. This information is summarized in Fig. 2.

5. Results

A number of different configurations are possible for different combinations of the three aspects discussed before (agency, knowledge and sensing). This holds for both the simulated and real world case study, with some caveats. In this section, we summarize the results for both, using the most interesting configurations to evaluate their impact on learning over time:

1. The default rule-based controller (RBC)
2. Aggregation of multiple rule-based controllers
3. Single agent reinforcement learner (SARL(.))
4. Single agent reinforcement learner with additional knowledge (SARL(K))
5. Single agent reinforcement learner with additional knowledge and additional sensors (SARL(K, I))
6. Multi agent reinforcement learner (MARL(.))
7. Multi agent reinforcement learner with additional knowledge (MARL(K))
8. Multi agent reinforcement learner with additional knowledge and additional sensors (MARL(K, I))

The rule-based controllers follow the fixed hysteresis dead-band implementation of Eq. (9). The additional knowledge here refers only to constraining the output values as discussed in a previous section. This is only applicable for the case of the storage vessel. Feature engineering is used in all cases as it replaces raw consumption time series with features which can be generalized across households. Without feature extraction, raw time series learning fails to converge to a reliable model, and multi-agent learning can not take place in a straightforward manner.

5.1. Theoretical study

Fig. 3a summarizes the exploration achieved with different control and aggregation strategies: the contrast in state-space exploration for rule-based controllers and RL based controllers is quite obvious. Common to both mechanisms in varying degrees is the tapering off of experiencing new states as periodic human behavior causes the agents to follow largely similar schedules. It is also obvious that by keeping everything else the same, multi-agent systems explore much better than their single agent counterparts. This is hardly surprising. What is surprising is that single agent systems possessing both domain knowledge

[2] www.eebus.org

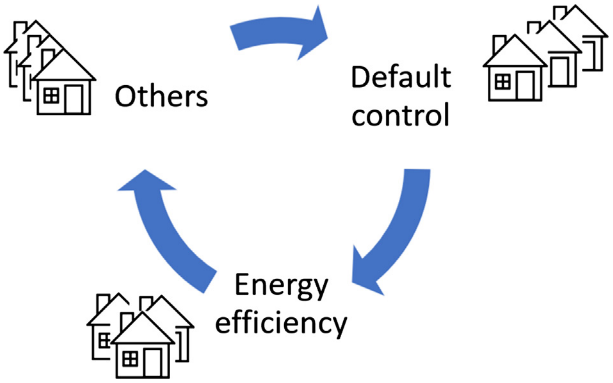

Fig. 2. Experimental setup in the real world pilot: all houses were cycled into different control groups on a weekly basis to remove occupant behaviour influence.

than MARL(K) after a month but it makes some costly mistakes after observing data for a year. While the overall effect is small, the two frameworks also make errors with opposite polarities, i.e. MARL(K) often underestimates the amount of hot water left in the vessel and MARL(K,I) slightly overestimates it. This is primarily because of a mismatch between the data arriving from assumed knowledge (K) and the sensing component (I). The interference between the two data streams leads to some costly errors which can be fixed by fine-tuning the knowledge component, however this would defeat the purpose of having a data-driven controller. This effect also arises as an artifact of learning itself (whereby the MARL(K,I) framework observes so much data that it is not as aggressive in its exploration as the MARL(K) configuration, especially at the interface of user comfort around 45 °C).

The performance gains when moving from a rule based controller to a reinforcement learning based system exceed 40% as shown in Fig. 5. This is true regardless of the RL configuration chosen. It appears that the highest energy efficiency is achieved by the single agent reinforcement learning configuration which makes use of just the additional domain knowledge.

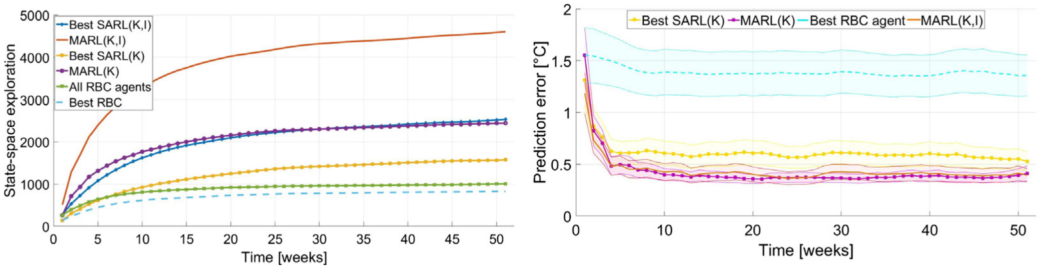

Fig. 3. (a) State-space exploration as a measure of unique states visited by the agents [unit-less quantity]; (b) MAE for learned transition model [°C], with different configurations.

and extra sensors (SARL(K,I)) perform comparably with multi-agent learners with only domain knowledge (MARL(K)). This provides preliminary evidence that agency can be used interchangeably with additional sensing.

Fig. 3b shows that better exploration does indeed correlate well with better accuracy. The prediction error is defined as the one-step ahead prediction compared with the actual realized temperature. As the ground truth (temperature distribution) is known in the simulation environment, we can test the agent's representation on states not previously observed.

Most of the gains towards learning an accurate model are realized in the first few weeks of operation. Nevertheless, there are a few interesting points to note here. Rule based controllers are unable to learn a reliable representation of the vessel even after observing an entire year of data. Likewise, a single agent equipped with only some domain knowledge is unable to reach the performance of a similar multi-agent system, and requires additional sensors to get there. This is not true when we transition from the MARL(K) configuration to the MARL(K,I) configuration; indicating that domain knowledge and extra sensing encode similar information and combining both yields no further benefit to the learning process.

Fig. 4 visualizes the prediction error of the storage vessel in a more detailed manner for the different schemas. It reinforces the findings from Fig. 3 about the rule-based controller and brings to light the potentially costly errors single agent learners continue to make even after observing data from a whole year. This is the case for lost occupant comfort, where the actual temperature has fallen below 45 °C but the predicted temperature is still above 45 °C, indicating the false belief that there is still hot water remaining in the storage vessel. The multi-agent systems on the other hand do not suffer from this problem, having learned a far more accurate representation after only a month.

Fig. 4 also brings to light a curious phenomenon whereby the MARL (K,I) configuration makes different mistakes than the MARL(K) configuration over time. It is obvious that the MARL(K,I) performs better

The impact of this optimization on occupant comfort is visualized in Fig. 6a which shows that the single agent reinforcement learner is the only configuration where consumption temperature falls below 45°C because of a poor system dynamics model caused by inadequate exploration.

Fig. 6b shows that the time period between successive reheat cycles for multi-agent configurations is also substantially more spread out than for the single agent case. This is explained by two reasons: (1) better contingency planning because of a better dynamics model, and (2) greater exploratory actions made possible by greater agency. As expected for the rule based controller, the spread on time period between successive reheat cycles is very limited. The median of this distribution is also much lower than the time for the reinforcement learning based controllers which explains the overall higher energy consumption.

5.2. Privacy aware learning

The proposed multi-agent performance is compatible with the feature-sharing learning paradigm as presented above. Since feature extraction reduces the data dimensionality and scrambles temporal structure, this has the advantage that it alleviates privacy concerns and reduces data communication costs.

Nevertheless, it might be possible to infer some household characteristics from these features. To address this, it is possible to transmit only locally trained models to the central oracle. More concretely, this follows a three-step process: first, the individual agents train and transmit their models to a central oracle. Second, the oracle generates a (random) data set in input feature space and generates outputs from all individual models. Finally, it uses the model outputs to train a single model which it then shares with the contributing agents.

This three-step approach caused problems as the individual models

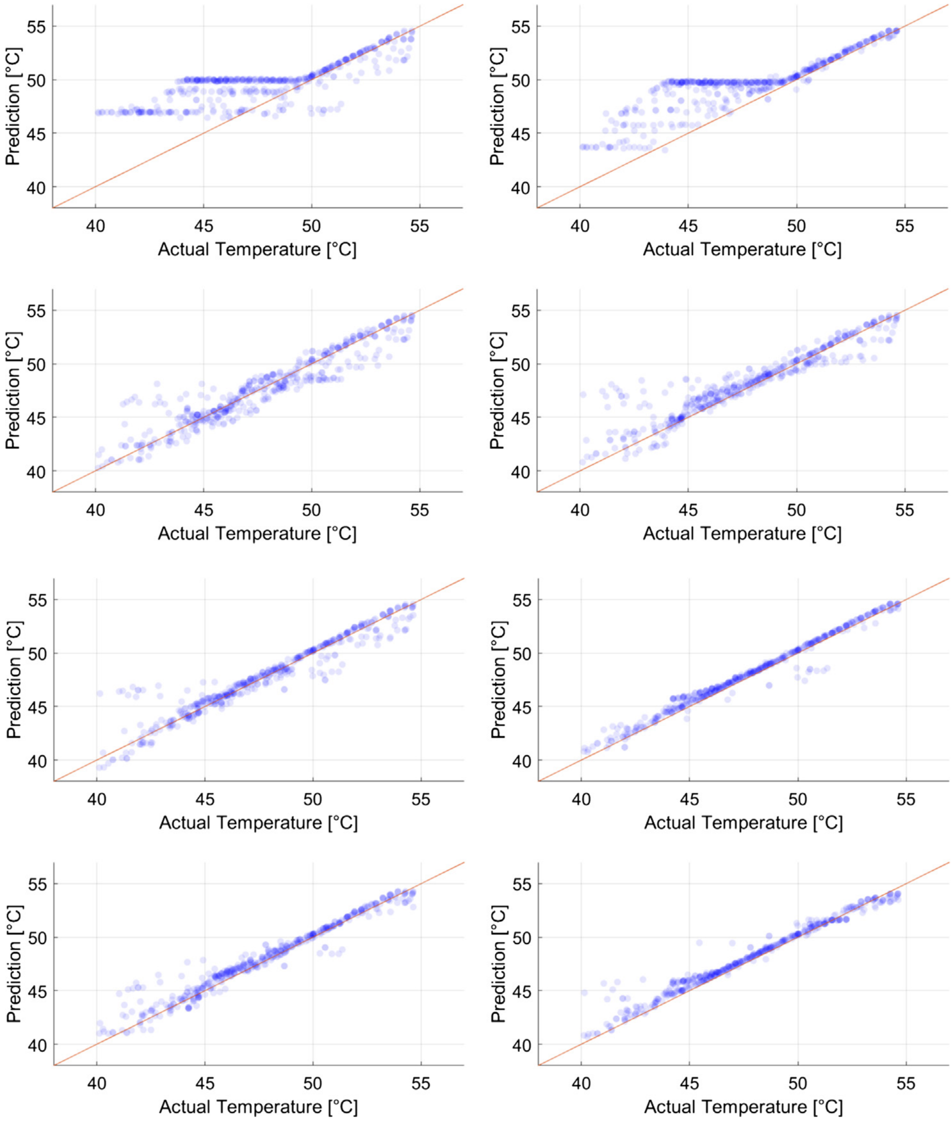

Fig. 4. Predicted and observed temperature, snapshot at different time periods, for [time (left to right): 1 week, and 1 year; configuration (top to bottom): Aggregation of RBC agents, SARL(K), MARL(K), and MARL(K,I)].

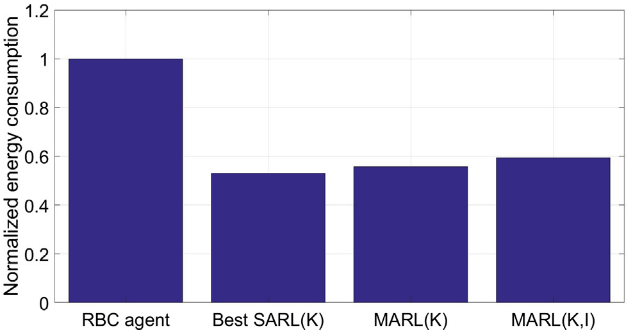

Fig. 5. Normalized energy consumption as a function of different learning configurations.

were prone to over-predicting the vessel state of charge. As explained before, this is caused by poor exploration at the edge of occupant comfort as discussed before. By training a single model which makes risk-constrained predictions (i.e. taking a lower confidence bound), it is possible to minimize this risk. As Fig. 7 shows, this distributed learning mechanism can approximate the performance of centralized learning. The system dynamics model learned this way however produces more conservative estimates of the state of charge and can be less efficient than a centralized controller.

5.3. Real world case study

In this section, we present results from a real world case study conducted using the proposed framework for a set of 53 houses analyzed over a year. A subset of these - 32 to be precise - formed the basis of the learned models, while another subset was used to test the efficacy of the controller. A subset of six households was equipped with additional sensors. Some of the differences between this and the simulated case study have already been highlighted in a previous section. As expected, both the quantity and quality of data in real life was much lower than the simulated study especially for the case of additional sensing, and reflected real world limitations to experiment design. Furthermore, it wasn't possible to explore all the different configurations presented above because of the limited amount of households. Therefore, while different learning strategies were evaluated, only two controllers were evaluated. In the remainder of this section, we present results of interleaving learning and control.

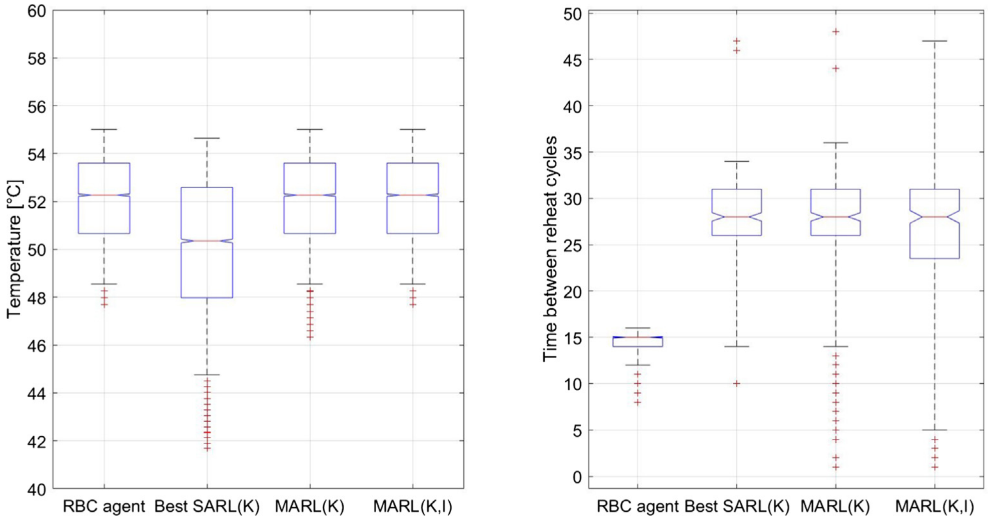

Fig. 6. Results for the simulated case study: (a) water consumption temperature [°C]; (b) duration between reheat cycles [h].

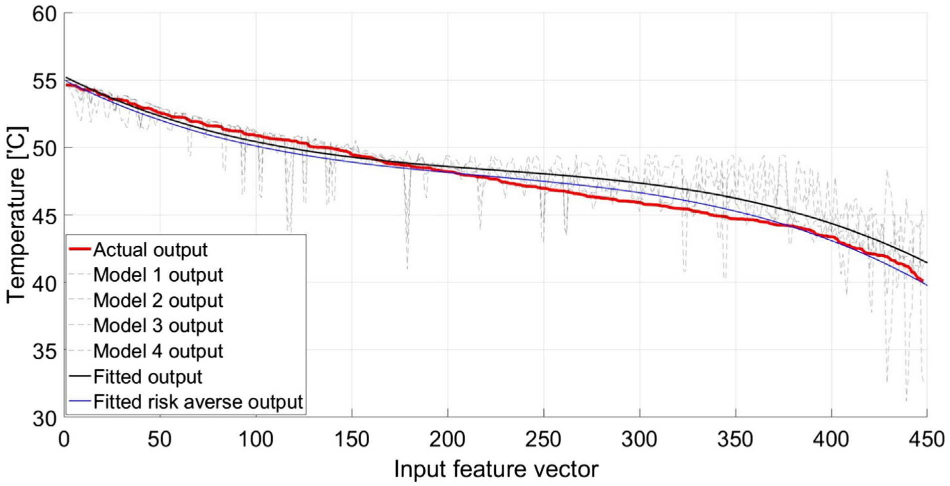

Fig. 7. Distributed learning with parameter sharing; modelling performance remains remarkably similar to centralized learning.

5.3.1. Storage model

Fig. 8 provides compelling proof of the efficacy of multi-agent learning of the TCL model. It summarizes the performance of the learned model as we increase the amount of data used to train the model. In doing so, it provides a detailed comparison of the MARL(K) and SARL(K) configurations. It is obvious that model performance improves with increasing amounts of data i.e. irrespective of whether the data is observed during a prolonged period with a single vessel or if it is aggregated over multiple vessels in a brief time period. This is evidenced by the decreasing mean absolute error as we traverse along the matrix towards increasing agents (y-axis) or time (x-axis). It is important to note that there is a large difference between the initial performance of the MARL(K) configuration and the SARL(K) configuration as we increase the amount of agents. On the other hand, the MARL(K) model learned after a single week of data collection is already close to the asymptotic performance, so there is not much improvement as more data is gathered. In this case, the primary utility of a multi-agent system over a single-agent system is the speed with which an accurate model is learned.

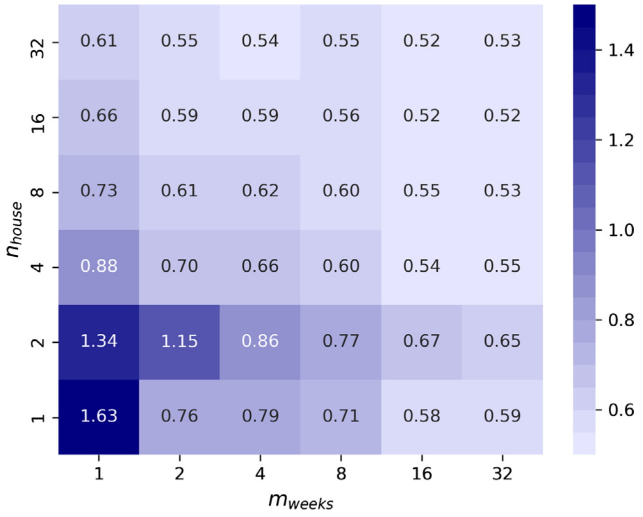

Fig. 8. Mean absolute error (MAE) [°C] for the hot water vessel model as a function of time (x-axis) and agents (y-axis).

It is obvious that the MARL(K) configuration outperforms the SARL (K) configuration. What is not clear is how the multi-agent configuration without any extra domain knowledge MARL(.) performs compared to the SARL(K) configuration. To evaluate this, we visualized the estimated state of charge (SoC) as a function of thermodynamic and mixing losses caused by hot water consumption. The vessel SoC is expected to be close to 1 right after a reheat cycle, after which it is expected to drop monotonically to 0 with the passage of time and/or user consumption

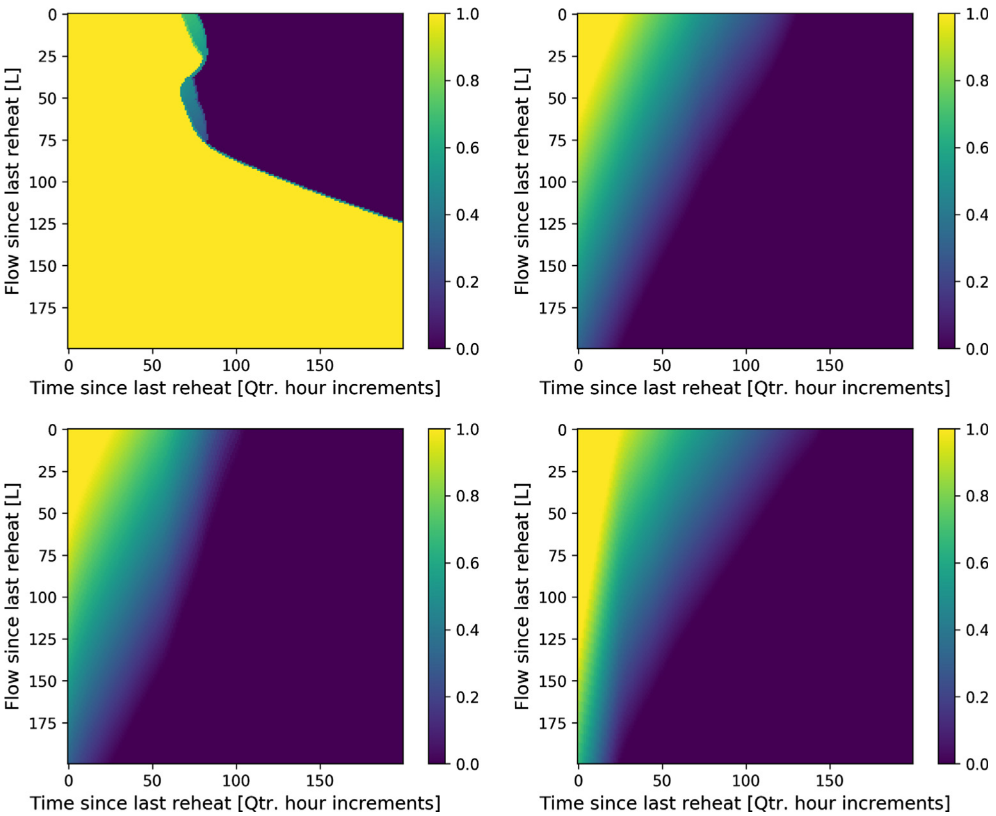

Fig. 9. Estimated state of charge in the storage vessel as a function of hot water consumption and thermodynamic losses, where the model has been trained for varying amounts of data: (top-left) 1 agent observed for 1 week; (top-right) 1 agent observed for 32 weeks; (bottom-left) 32 agents observed for 1 week; (bottom-right) 32 agents observed for 32 weeks.

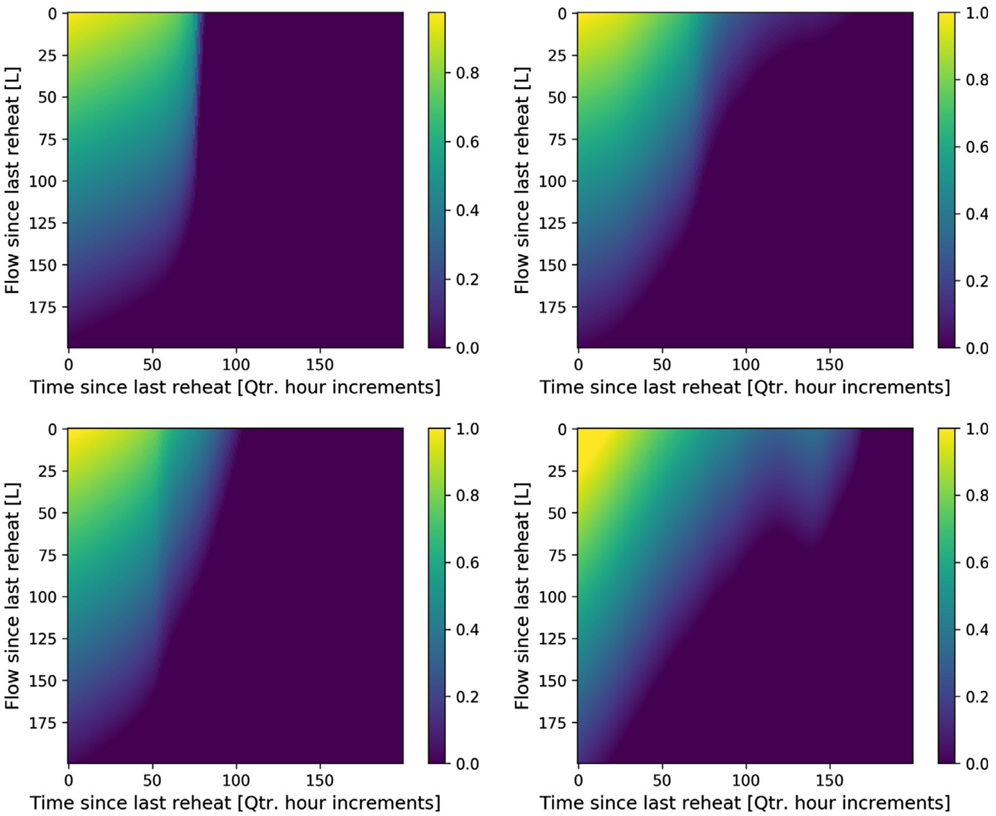

Fig. 10. Estimated state of charge in the storage vessel as a function of hot water consumption and thermodynamic losses while incorporating constraints on the model, which has been trained for varying amounts of data: (top-left) 1 agent observed for 1 week; (top-right) 1 agent observed for 32 weeks; (bottom-left) 32 agents observed for 1 week; (bottom-right) 32 agents observed for 32 weeks.

of hot water. The different SoC plots can be seen in Fig. 9 for the case of SARL(.) and MARL(.) and Fig. 10 for the case of SARL(K) and MARL(K). The biggest difference is between the SoC plot learned in the SARL(.) configuration after 1 week and the SARL(K) configuration after the same amount of time. It is obvious that the regularizing effect of domain knowledge incorporated via constrained learning has enabled the SARL(K) configuration to make sensible predictions about the SoC in the vessel while the SARL(.) configuration fails to do so. It is important

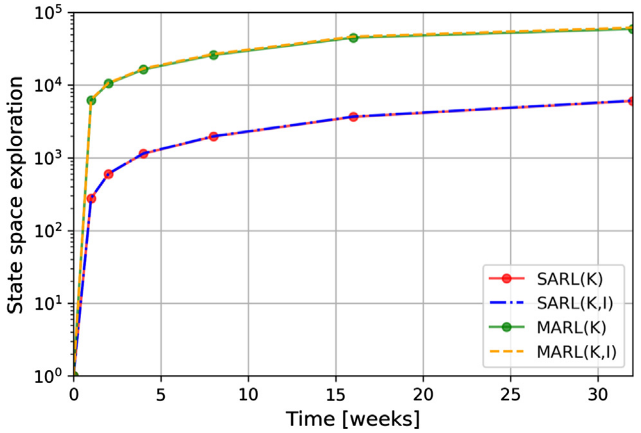

Fig. 11. Exploration as a function of time and different configurations with real world TCLs; here exploration is defined as the count of unique states visited by the agent and is a unitless quantity.

to note that the SARL(.) configuration has learned a fairly similar SoC representation after 32 weeks as its MARL(.) counterpart after only 1 week. These are also not too different from their SARL(K) and MARL (K) counterparts with additional data. In offline tests (performed by physically draining the vessel and measuring the outflow water temperature), the error of the multi-agent SoC estimates was consistently less than 10% on average.

Based on the evidence from Figs. 8–10, it is clear that multi-agent modelling brings about many benefits. The same effect, albeit to a less generalizable degree, can be achieved by incorporating domain knowledge in the form of constraints which enables even single-agent systems to learn with extremely limited amounts of data (in this case, only one week). At its heart lies a better exploration of the state-space which is visualized in Fig. 11. This is not a cosmetic difference as it is the difference between optimally controlling the vessel almost immediately as opposed to waiting for a prolonged period of time during which data is collected for a reliable model to be constructed.

Finally, additional sensing in the real world brought only limited benefits. This was seen most clearly by the fact that additional sensing did not noticeably increase state-space exploration (Fig. 11). The amount of data gathered by these sensors was usually fewer than the temperature data recorded by the midpoint sensor by at least one to two orders of magnitude. Furthermore, the data gathered was of questionable quality because of limited sampling time and conduction delays. In our experiments, the MARL(K,I) and SARL(K,I) configurations therefore did not offer any benefit over the MARL(K) configuration in the real world.

5.3.2. Heating model

The situation is vastly different for the heating element model where data gathered by each household is much more sparse, by almost two orders of magnitude. Fig. 12 summarizes the result of learning for both the MARL(.) and SARL(.) configurations. Like the vessel model, the mean absolute error for the heat pump model is also reduced drastically when increasing either the amount of time the data is gathered or the amount of agents under observation. This translates to a much higher initial performance for the multi-agent case. However, unlike the vessel model, performance continues to increase throughout the data collection period and the asymptotic performance achieved by 32 agents over 32 weeks of data collection is far superior to that achieved by a comparable single agent system. As the average reheat energy consumed by the heat pump is between 1.5 and 2 kW h's, the asymptotic performance for the multi-agent system shows a relative error of less than 10% while it is still over 20% for the single-agent case.

Mean absolute error can, on occasion, be a rather misleading measure of quality. From Fig. 12, it seems that the model learned with data collected from 32 agents for 1 week is better than the model learned with 32 weeks of data for a single agent. This is however not completely true as visualized in Fig. 13 which highlights an interesting effect. While the multi-agent performance is indeed better than a comparable single-agent system on the MAE metric, it is actually slightly worse at generalizing. This is highlighted by the R^2 metric in Fig. 13. One reason for this slightly poorer performance with multiple agents is the heat pump's dependence on ambient conditions. A model trained with data observed only for one week will fail to generalize to very different ambient conditions. This is one reason why the model continues to improve throughout the data collection period as it gets to experience different ambient conditions.

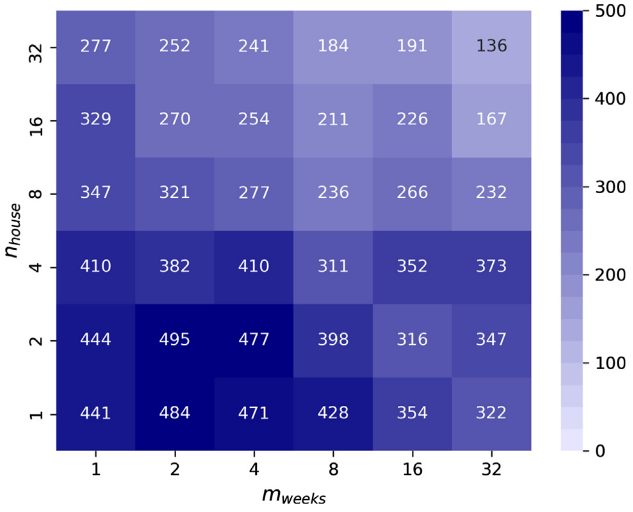

Fig. 12. Mean absolute error (MAE) [Wh] for the heat pump model as a function of time (x-axis) and agents (y-axis).

5.3.3. Control

As the vessel model was learned in a very short time period, this was used to optimize the operation of the hot water systems in operation according to Eq. 11. The heat pump model took substantially longer to train to a reliable accuracy and was therefore not employed by the controller. Another reason to only apply a simplified controller, rather than the full reinforcement loop, was because of complexity. While Eq. 11 means that every household runs a simple controller (i.e. the only difference is the instantaneous predicted household water demand), a full reinforcement controller means every household follows a unique policy. This leads to complications with debugging when things go wrong - as they often do in the real world - and makes it difficult to disentangle problems with the learning, control and other parts of the work flow (e.g. communication, etc.).

Fig. 14 visualizes the energy consumed over 11 months for households running the default controller and the efficient controller. It is obvious that the efficient controllers consistently consume less energy on average. As these results are usually averaged over anywhere between 15 and 20 houses for both groups every week, they are fairly representative. While the energy consumption spikes during the winter months, it is interesting to note that the efficiency improvements hold, in relative terms, throughout the year. Over the course of 11 months, these savings came out to be around 200 kW h per household and confirmed our earlier, limited scale tests. This translates to around 20% of total energy demand for domestic hot water. Fig. 15 provides a clearer view of the energy savings observed over the year in a cumulative fashion. This plot was created by randomly sampling energy consumption of different households from the two different subgroups (default and efficient) for 10,000 times. It clearly shows that the default group consumed over 200 kW h's more than the energy efficient group. However, there is some uncertainty in the results which means that

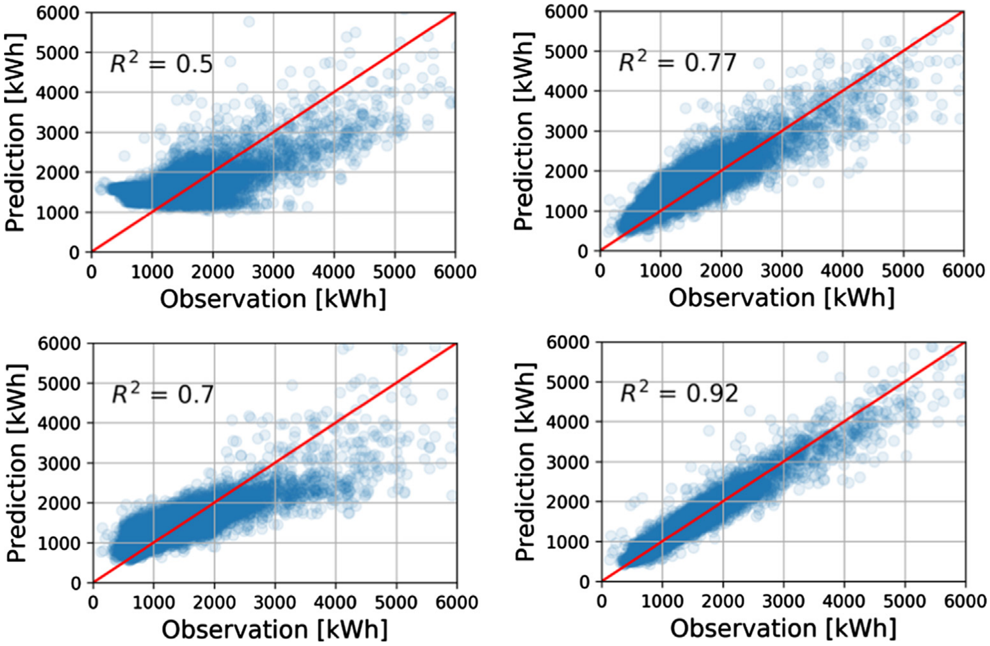

Fig. 13. Scatter plot between observed and predicted electricity consumption for the heat pump as a function of increased data and agency: (top-left): 1 week of data for 1 agent; (top-right) 32 weeks of data for 1 agent; (bottom-left): 1 week of data for 32 agents; (bottom-right): 32 weeks of data for 32 agents.

savings could range between 100 and 350 kW h over the course of a year (Fig. 15).

Despite the large difference in energy consumption, the hot water consumption in both groups was unsurprisingly very similar at the end of the year (there is less than 1% difference in hot water consumption in the two groups). No complaints pertaining to comfort loss were received from the householders that could be attributed to active control. Householders had the option to inform when control went wrong and, in many instances, they communicated actively about loss of comfort caused by equipment or coordination problems. An example of this was seen when a building's occupants noticed and complained about a lack of hot water; it turned out that set-points had been incorrectly set during the weekly hand-off between different control groups. However, over the course of a year, there were no complaints caused by modelling or control errors.

It is important to note here that the controller used for efficiency in this case was demonstrably sub-optimal (i.e. it failed to take into account the time-varying efficiency of the heat pump). In fact, simulations show that had such information been taken into account, the savings would have been even higher at almost 30% of the energy demand (or almost 300 kW h per household). While these additional savings are important, they come at the cost of increased complexity.

An additional point to consider is that, unlike the simulated case, hot water consumption profile of a household can be rather unpredictable in real life [17]. We use a backup controller to ensure that user comfort is guaranteed at all times which reduced the efficiency gains when compared with the simulated case study. Another reason for the lower than theoretical efficiency gains was the complexity associated with learning a more complex, stochastic model for both the storage and heating elements.

6. Conclusion and future work

In this paper, we have demonstrated that using multiple agents can

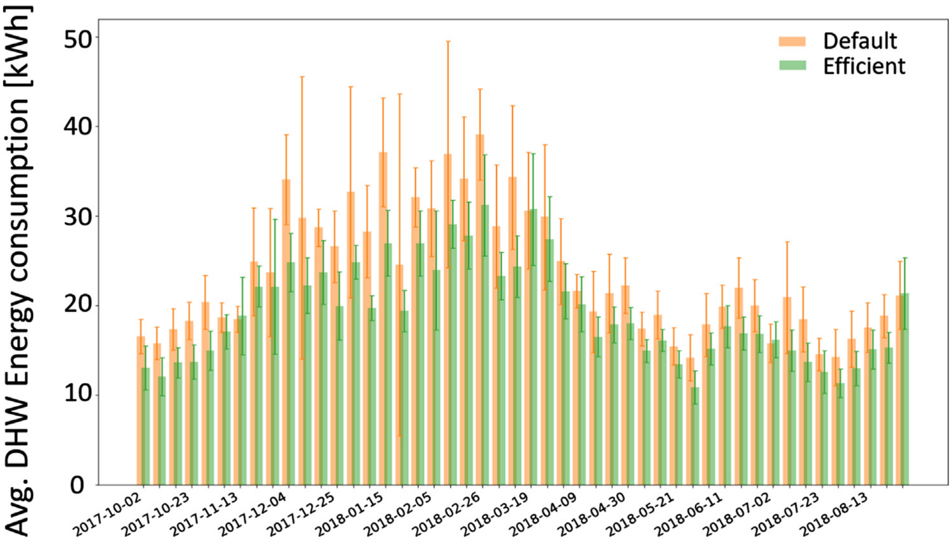

Fig. 14. Efficiency gains over 11 months of active control compared with the default controller for hot water production in sub-groups of 53 households.

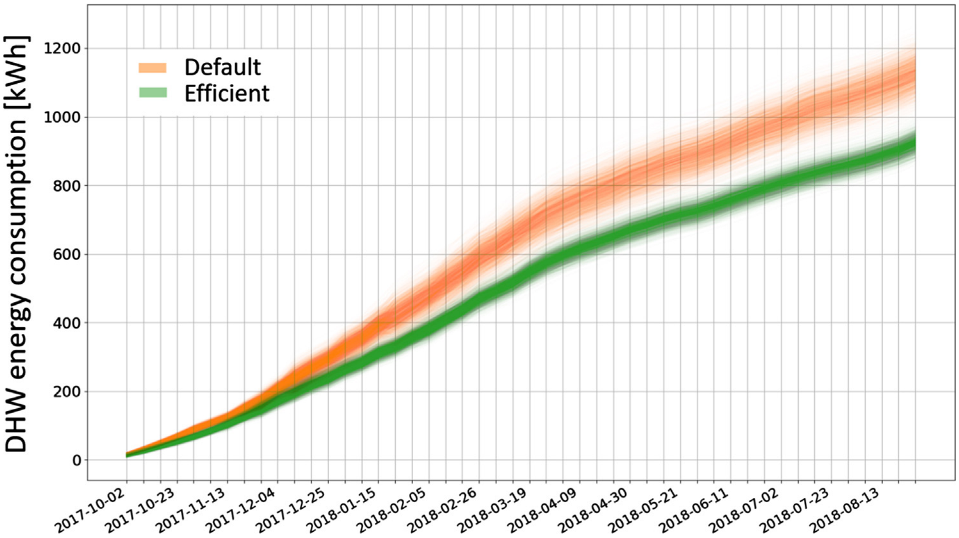

Fig. 15. Cumulative energy consumption over 11 months of active and default control for hot water production in sub-groups of 53 households; different draws are sampled randomly from measured data.

reduce dependence on sensing for black-box modeling and control of thermostatically controlled loads. They also largely circumvent the need for specialized human domain knowledge. Results obtained using both theoretical and actual thermostatically controlled loads confirm this hypothesis. The only requirement for the framework is the existence of multiple thermostatically controlled loads interacting with different households and the infrastructure enabling multi-agent communication and control.

These research findings have positive consequences for the practical applicability of such systems as sensors represent a significant cost of smart systems, both initially and operationally. By reducing sensor dependence, smarter buildings can become more cost-effective and therefore more attractive to investors and home owners.

The research also serves as an important guide for the future. By disambiguating different sources of information and their impact on model performance, it highlights the way to learning reliable dynamics models for hot water systems in the least amount of time at the lowest possible cost. The multi-agent framework is also a marked improvement on both white- and gray-box models, as it removes the dependence on human domain expertise and is completely generalizable to new vessel types.

Additionally, in countries like The Netherlands where social houses make up for a large fraction of the national housing stock, the replication potential opens up endless possibilities for the presented system. In buildings where energy monitoring is already taking place, the framework allows additional optimal control which can save between 200 and 300 kW h's annually at no extra cost. Part of these efficiency gains can also be offered to the electric grid as ancillary services. At this point, it is also important to reflect on how these efficiency gains are possible. The single biggest contributing factor to realizing the efficiency gains is the increased cycling time between reheat cycles which has been borne by both simulations and the real world case study. The improved heat pump efficiency, because of reheating the vessel from a lower temperature, also contributes to reducing the energy consumption.

It is important to point out that the reduction in energy demand was not at the cost of increased hot water demand (or altered consumption patterns). We were able to test the hypothesis whether the control mechanism was altering hot water demand, as each house was controlled with the default and energy efficient controller in an alternating manner. There was no statistically significant difference in hot water consumption for the same household, regardless of whether it was controlled by the default rule-based controller or the energy efficient controller. This provides credible evidence that the temperature of hot water exiting the vessel is sufficiently hot to not increase hot water demand and therefore bias efficiency calculations. This was also confirmed by temperature sensors placed at the outflow of a subset of the storage vessels.

In our experiments, learning using raw time series data did not generalize well. While we solved this problem by feature engineering, it reinforced the notion that black-box models should not be treated as silver bullets to modelling tasks. The proposed system additionally incurs additional costs because of the infrastructure required to communicate the gathered data. With increasing proliferation of smart meters and the internet of things, this will become less of a concern over time. Despite the sizable energy efficiency gains offered by the framework, the situation is less clear in buildings which do not have access to a communication link. There, a decision on the cost-effectiveness of the system has to be made on a case-by-case basis. This decision will depend not just on the communication costs but also the prevalent energy tariffs and eco-consciousness of householders. Novel formulations, like the distributed learning algorithm presented in this paper, can help with minimizing communication overhead while still offering privacy-preserving optimization.

Future directions to extend this work include other system identification problems which can benefit from multi-agent configurations. These include the broader class of thermostatically controlled loads (i.e. space heating and cooling), as well as ventilation systems for indoor air quality control. Additionally, more general global optimization problems for solving grid interaction issues can also benefit from the improved system identification potential of the proposed multi-agent framework. Finally, while the focus of this research has been on homogeneous devices (i.e. identical hot water systems), learning from and acting on heterogeneous devices in multi-agent contexts is an even more challenging problem. This remains a promising avenue for future research.

Acknowledgment

Hussain Kazmi acknowledges support and valuable feedback from VLAIO, InnoEnergy and the IEA Annex 67. Johan Suykens acknowledges support of CoE PFV/10/002 (OPTEC), FWO G0A4917N, G088114N, IUAP P7/19 (DYSCO), ERC Advanced Grant E-DUALITY (787960), KU Leuven C1.

References

[1] Prez-Lombard Luis, Ortiz Jos, Pout Christine. A review on buildings energy consumption information. Energy Build 2008;40(3):394–8.

[2] Kundu, Soumya, Sinitsyn, Nikolai, Backhaus, Scott, Hiskens, Ian. Modeling and control of thermostatically controlled loads; 2011. arXiv preprint arXiv:1101.2157.

[3] Ben-Nakhi Abdullatif E, Mahmoud Mohamed A. Energy conservation in buildings through efficient A/C control using neural networks. Appl Energy 2002;73(1):5–23.

[4] Koch, Stephan, Mathieu, Johanna L, Callaway Duncan S. Modeling and control of aggregated heterogeneous thermostatically controlled loads for ancillary services. In: Proc PSCC; 2011.

[5] Yin R, Kara EC, Li Y, DeForest N, Wang K, Yong T, et al. Quantifying flexibility of commercial and residential loads for demand response using setpoint changes. Appl Energy 2016;177:149–64.

[6] Hao He, Sanandaji Borhan M, Poolla Kameshwar, Vincent Tyrone L. Aggregate flexibility of thermostatically controlled loads. IEEE Trans Power Syst 2015;30(1):189–98.

[7] Ali Adhra, Kazmi Hussain. Minimizing grid interaction of solar generation and DHW loads in nZEBs using model-free reinforcement learning. International workshop on data analytics for renewable energy integration. Springer, Cham; 2017.

[8] Hensen Jan LM, Roberto Lamberts, editors. Building performance simulation for design and operation. Routledge; 2012.

[9] Kazmi, Hussain, Mehmood, Fahad, Amayri, Manar. Smart home futures: algorithmic challenges and opportunities. In: 2017 14th international symposium on pervasive systems, algorithms and networks & 2017 11th international conference on Frontier of computer science and technology & 2017 third international symposium of creative computing (ISPAN-FCST-ISCC). IEEE; 2017.

[10] Kusiak A, Li M, Tang F. Modeling and optimization of HVAC energy consumption. Appl Energy 2010;87(10):3092–102.

[11] Ruelens Frederik, Claessens Bert, Quaiyum Salman, De Schutter Bart, Babuska Robert, Belmans Ronnie. Reinforcement learning applied to an electric water heater: from theory to practice. IEEE Trans Smart Grid 2016.

[12] Henze Gregor P, Schoenmann Jobst. Evaluation of reinforcement learning control for thermal energy storage systems. HVAC&R Res 2003;9(3):259–75.

[13] Wen Zheng, ONeill Daniel, Maei Hamid. Optimal demand response using device-based reinforcement learning. IEEE Trans Smart Grid 2015;6.5:2312–24.

[14] Deisenroth Marc, Rasmussen Carl E. PILCO: a model-based and data-efficient approach to policy search. Proceedings of the 28th international conference on machine learning (ICML-11). 2011.

[15] Ruelens Frederik, Claessens Bert J, Vandael Stijn, De Schutter Bart, Babuka Robert, Belmans Ronnie. Residential demand response of thermostatically controlled loads using batch reinforcement learning. IEEE Trans Smart Grid 2017.

[16] Nagy, Adam, Kazmi, Hussain, Cheaib, Farah, Driesen, Johan. Deep reinforcement learning for optimal control of space heating; 2018. arXiv preprint arXiv:1805.03777.

[17] Kazmi H, D'Oca S, Delmastro Chiara, Lodeweyckx S, Corgnati Stefano Paolo. Generalizable occupant-driven optimization model for domestic hot water production in NZEB. Appl Energy 2016;175:1–15.

[18] Majcen Dasa, Itard Laure, Visscher Henk. Actual and theoretical gas consumption in Dutch dwellings: what causes the differences? Energy Policy 2013;61:460–71.

[19] Vanthournout Koen, D'hulst Reinhilde, Geysen Davy, Jacobs Geert. A smart domestic hot water buffer. IEEE Trans Smart Grid 2012;3(4):2121–7.

[20] Kreuzinger Tobias, Bitzer Matthias, Marquardt Wolfgang. State estimation of a stratified storage tank. Control Eng Pract 2008;16(3):308–20.

[21] Kazmi Hussain, Mehmood Fahad, Lodeweyckx Stefan, Driesen Johan. Gigawatt-hour scale savings on a budget of zero: deep reinforcement learning based optimal control of hot water systems. Energy 2017.

[22] Chertkov M, Chernyak VY, Deka D. Ensemble control of cycling energy loads: Markov decision approach. Energy markets and responsive grids. New York, NY: Springer; 2018. p. 363–82.

[23] Bomela W, Zlotnik A, Li Jr, S. A phase model approach for thermostatically controlled load demand response; 2018. arXiv preprint arXiv:1803.03379.

[24] Tan Ming. Multi-agent reinforcement learning: independent vs. cooperative agents. Proceedings of the tenth international conference on machine learning. 1993.

[25] Mathieu, Johanna L., Callaway, Duncan S. State estimation and control of heterogeneous thermostatically controlled loads for load following. In: 2012 45th Hawaii International Conference on System Science (HICSS). IEEE; 2012.

[26] Lu Ning, Zhang Yu. Design considerations of a centralized load controller using thermostatically controlled appliances for continuous regulation reserves. IEEE Trans Smart Grid 2013;4(2):914–21.

[27] Kok J Koen, Warmer Cor J, Kamphuis IG. PowerMatcher: multiagent control in the electricity infrastructure. Proceedings of the fourth international joint conference on autonomous agents and multiagent systems. ACM; 2005.

[28] McKenna Eoghan, Richardson Ian, Thomson Murray. Smart meter data: balancing consumer privacy concerns with legitimate applications. Energy Policy 2012;41:807–14.

[29] McDaniel Patrick, McLaughlin Stephen. Security and privacy challenges in the smart grid. IEEE Secur Priv 2009;7.3.

[30] Li Mu, Andersen David G, Park Jun Woo, Smola Alexander J, Ahmed Amr, Josifovski Vanja, et al. Scaling distributed machine learning with the parameter. Server OSDI 2014;1(10.4).

[31] Osborne Martin J, Rubinstein Ariel. A course in game theory. MIT Press; 1994.

[32] Monahan George E. State of the arta survey of partially observable Markov decision processes: theory, models, and algorithms. Manage Sci 1982;28(1):1–16.

[33] Sutton Richard S, Barto Andrew G. Reinforcement learning: an introduction vol. 1(1). Cambridge: MIT Press; 1998.

[34] Criminisi Antonio, Shotton Jamie, Konukoglu Ender. Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found Trends Comput Grap Vis 2012;7(23):81–227.

[35] MacKay David JC. Bayesian neural networks and density networks. Nucl Instrum Meth Phys Res Sect A: Accel Spectrom Detect Assoc Equip 1995;354.1:73–80.

[36] Gal, Yarin, Ghahramani, Zoubin. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: International conference on machine learning; 2016.

[37] Jacobs Robert A, Jordan Michael I, Nowlan Steven J, Hinton Geoffrey E. Adaptive mixtures of local experts. Neural Comput 1991;3(1):79–87.

[38] Avnimelech Ran, Intrator Nathan. Boosted mixture of experts: an ensemble learning scheme. Neural Comput 1999;11(2):483–97.

[39] Shokri Reza, Shmatikov Vitaly. Privacy-preserving deep learning. Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. ACM; 2015.

[40] Zhang, Jiangfeng, Xia, Xiaohua. Best switching time of hot water cylinder-switched optimal control approach. AFRICON 2007. IEEE; 2007.

[41] Ng, Andrew Y, Jordan, Michael. PEGASUS: a policy search method for large MDPs and POMDPs. In: Proceedings of the sixteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 2000.

[42] Peshkin, Leonid, Kim, Kee-Eung, Meuleau, Nicolas, Kaelbling, Leslie Pack. Learning to cooperate via policy search. In: Proceedings of the sixteenth conference on uncertainty in artificial intelligence. Morgan Kaufmann Publishers Inc.; 2000.

[43] Jin X, Baker K, Christensen D, Isley S. Foresee: a user-centric home energy management system for energy efficiency and demand response. Appl Energy 2017;205:1583–95.