

Article

Personalized Federated Actor–Critic Learning for Joint Cost–Comfort Optimization in Energy Communities

Sotirios Spantideas ^{1,*}  and Anastasios Giannopoulos ² ¹ Department of Electrical and Electronics Engineering, School of Engineering, University of West Attica, 12244 Athens, Greece² School of Electrical and Computer Engineering (ECE), National Technical University of Athens, 15780 Athens, Greece; angianno@mail.ntua.gr

* Correspondence: sspantideas@uniwa.gr

Abstract

Home energy management systems (HEMS) aim to provide intelligent control of the thermal comfort inside smart buildings with the minimum energy cost, while satisfying the energy consumption requests and increasing the use of energy from renewable sources. The capabilities of these intelligent HEMS agents are restricted due to the personalized observability of the environment, resulting in limited knowledge gathering and potentially sub-optimal decisions. Furthermore, several buildings have recently been organized into small energy communities, with the ultimate goal of sharing intelligence between agents in federated learning schemes. In this context, we propose a personalized federated deep reinforcement learning method using Moreau envelopes (pFedMe) for joint energy cost and household comfort optimization in energy communities that consist of multiple smart homes. Specifically, a Twin-Delayed Deep Deterministic Policy Gradient (TD3) actor–critic model is introduced, dynamically observing the state of the smart home environment and suggesting control actions on the operation of the Energy Storage System and on the regulation of the indoor temperature. The TD3 actor–critic model leads to improved policy performance in the continuous control of these systems, mitigating the overestimation bias and improving the training stability of the intelligent agents. The efficiency of the proposed method is verified via simulations based on real data, achieving a beneficial trade-off between the energy cost and the thermal comfort compared to FedAvg and Fedprox baselines. The results show that the proposed pFedMe framework consistently outperforms FedAvg and FedProx in both convergence speed and overall reward, achieving an energy cost reduction of approximately 10% compared to the other schemes, while exhibiting marginal thermal comfort behavior.



Academic Editor: Fco Javier Rodríguez

Received: 28 February 2026

Revised: 25 April 2026

Accepted: 4 May 2026

Published: 8 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: energy management system (EMS); energy storage systems (ESSs); heating, ventilation, and air conditioning (HVAC) systems; personalized federated learning with using Moreau envelopes (pFedMe); home automation; smart homes; deep reinforcement learning (DRL); energy community

1. Introduction

Residential and commercial buildings are considered key contributors to worldwide energy consumption, and several technical enablers have emerged during the last few years to limit their energy footprint and provide sustainable energy management frameworks [1,2]. These frameworks aim to deliver multi-optimization solutions, ranging from

thermal comfort inside a household to cost minimization related to the energy consumption and increase in the utilization of energy gathered from renewable sources [3–5]. In this context, energy management systems (EMS) use monitoring data generated by the building and the environment, for instance, from Internet of Things (IoT) devices, sensors, or cameras, to suggest control actions towards a fully automated solution. Moreover, they also leverage technical solutions such as Energy Storage Systems (ESS) to store the excess energy and IoT–edge–cloud continuum concepts to host the data processing and computing operations. Finally, the EMS can be assisted by Artificial Intelligence (AI) and Machine Learning (ML) models to incorporate cognitive agents that can interact with the building environment and provide long-term benefits towards optimization objectives, especially those relating to energy footprint minimization and the sustainability of smart buildings [6–9].

Although these smart residential and commercial buildings include monitoring capabilities, the design of an efficient EMS is extremely challenging, since several environmental, energy market and demand-driven parameters are involved, which exhibit combined dependencies and significant temporal variations [10,11]. To this end, the EMS must take into account periodical and seasonal variations in some parameters (for instance, the ambient temperature and the output of renewable energy generators), as well as parameters that might follow specific temporal patterns (the energy demands and loads of residential hosts that are typically increased during afternoon hours) and variables that cannot be estimated (e.g., the fluctuation in the electricity price in the energy market). Moreover, several environmental parameters are correlated, such as the use of Heating, Ventilation and Air Conditioning (HVAC) systems with ambient temperature (minimal use of HVAC is expected in cases that the outdoor temperature is in the range of 20–25 °C) or even the energy demands of the household with the electricity price, since an abrupt increase in the energy cost may prevent residents from requesting heavy consumption loads [12–14]. Although the inclusion of an ESS system that can act as an energy buffer is certainly a technical enabler for overcoming the latter scenario, it hinders the decision-making process imposed by the management system, since current choices of EMS impact future smart building states, requiring the configuration of an optimal EMS policy [15,16].

In this context, an intelligent EMS in a smart building must consider all the aforementioned parameters in the decision-making process of optimizing the overall energy consumption, which includes loads of inelastic demand (e.g., lights, television), controllable loads (such as the HVAC system) and adjustable loads (the operation of a washing machine can be scheduled according to the household's needs). The key contributor to the energy consumption of a smart building is the HVAC system (up to 40% of the energy cost is linked to HVAC operation), regulating the indoor temperature between desirable limits to maximize the comfort of the residents through a feedback loop [6,17,18]. Moreover, the number of electric vehicles (EV) that are charged either autonomously or through smart buildings has been significantly increasing during the last few years, impacting grid operations and presenting a huge concern for Electric Distribution System Operators (DSOs) [19]. The use of Vehicle to Grid (V2G) bidirectional chargers also complicates the management process, since EVs can work both as loads and ESSs. The key strategy of DSOs is to balance the local energy demand requested by the loads with the available energy produced by renewable sources in close proximity to the demand, considering smart buildings and EVs as prosumers at the edge that can dynamically act as consumers and producers, exchanging energy through the local electric network [20–22].

In principle, DSOs tackle this problem on three different levels: first of all, at the edge (EMS, EV chargers) with intelligent algorithms that optimize energy consumption and production directly at the origin, also adjusting the data exchanged between the entities with the grid and the upper layers [23–26]. Furthermore, an additional hierarchical layer is the fog, where an energy community or an aggregator entity named Community Orchestrator has been introduced to manage the energy exchange between several prosumers (smart buildings and EVs), aiming to achieve an inter-community energy balance and also coordinate in an intra-community manner in case of energy surplus or deficit [27–29]. Finally, the cloud level is a centralized controlling system that regulates the energy management of multiple communities through the Community Orchestrators. Focusing on the edge and fog layers of the distributed architecture, optimization algorithms based on AI/ML have been developed, such as reinforcement learning (RL) models, where an intelligent agent can control energy-related operations of some loads of EMS and EV systems. These intelligent agents also have a cognitive awareness of the energy market, since interaction with the grid might be imperative in cases of energy deficit. It is worth noting that in the case that communication with hierarchically higher entities fails, the intelligent RL models deployed at the edge are expected to manage the local needs autonomously [30,31].

In this context, these household-level or EV-level intelligent agents relying on RL or Deep RL (DRL) models are individually trained offline, since they require interaction with a simulated environment on a trial-and-error basis [6,7]. Then, the pre-trained models can be used for online inference to provide intelligent control of the household parameters, i.e., the operation or scheduling of the loads and interaction with the energy community and the power grid. However, these models in principle exhibit generalizability issues, since they are trained based on a local dataset that reflects the local environmental conditions encountered by the agent (e.g., ambient temperature) and the resident-specific patterns for energy consumption, as well as the temporal-specific electricity price fluctuations [32–34]. To overcome these challenges, the DRL models can be trained in a federated configuration, where, periodically, the model weights are shared between the participants (households and EVs) during the training process, disseminating the gathered knowledge among multiple participants and converting the local observability of each intelligent agent to a more generalized perception [35,36].

2. Related Work and Contributions

Regarding the energy management optimization problem of an individual building, several methods have been developed in the literature. Among others, the DRL-based methods are considered dominant for solving non-convex problems with conflicting optimization objectives in complex environments [37]. As mentioned, the DRL agents interact with the environment during their training, receiving feedback based on their actions and storing their experience in a replay memory [6,38]. For instance, the authors in [17] describe the modeling of the environment and the definition of a DRL model that controls the heating in smart buildings in order to jointly reduce the energy cost and maintain a comfortable temperature range. Specifically, they use Deep Q-network (DQN) agents to demonstrate the efficiency of the DRL with real data for the ambient temperature. Furthermore, Ref. [39] presents OCTOPUS, an HEMS system based on DRL that performs the intelligent scheduling of multiple electrical loads inside the smart building, including the HVAC system, the lighting, the blinds and the windows. The authors develop the DRL agents utilizing a branching dueling Q-network and showcase its efficacy in controlling the subsystems of the smart building environment. Similarly, the authors of [40] take into consideration the fluctuating electricity price of the market in the modeling process to configure the price of the individual energy surplus through a Deep Deterministic Policy Gradient (DDPG) agent

that is used as a DRL method to provide continuous suggestions across the control actions. Finally, the authors in [41] propose a generalized actor–critic learning (GACL) optimal control method, aiming to jointly optimize the energy management of a smart home and minimize the relevant cost by introducing three iteration processes in the GACL algorithm, namely, the global iteration, the local iteration, and the interior iteration.

Apart from training individually cognitive agents, multiple research studies that deal with collaborative intelligent schemes can be found in the literature. Specifically, the authors of [32] propose a federated reinforcement learning (FRL) approach for HEMS in homes with appliances/loads, a solar photovoltaic system as a renewable energy sources, and an ESS. The distributed HEMS agents are trained with local energy consumption data, uploading their local model to a global server (GS) that aggregates the local models into a global model, redistributing it back to the local HEMS. Moreover, the authors in [42] present a hierarchical framework for enabling collaborative intelligence between multiple smart homes using federated deep reinforcement learning (FDRL). In the context of the proposed framework, the intelligent HEMS agents that manage the smart building, consisting of ESS, loads and renewables, share their gained experience as model hyperparameters through a federation layer with the optimization objective of reducing costs and CO₂ emissions. In addition, Ref. [43] introduces a privacy-preserving FRL method that uses a shared ESS for multiple smart buildings, describing a hierarchical architecture based on a GS and local HEMS clients. To this end, the authors extensively describe how the FL framework is deployed, i.e., how the GS aggregates the local trained neural networks for energy consumption models and broadcast the global model, as well as training the shared ESS system to exchange energy (charging and discharging operations) between the utility grid and smart buildings. Finally, the authors of [44] propose a personalized energy management approach for each smart home that participates in the FL process, based on the method of personalized federated deep reinforcement learning (PFDRL). In particular, the proposed approach includes a global energy management strategy using FDRL, as well as a personalized fine-tuning scheme based on the divergence between the personalized strategy of each individual agent and the global strategy.

Recent studies have also explored advanced actor–critic architectures and their integration with communication-constrained or distributed environments. In [45], the authors propose an MSRA-TD3 framework for throughput maximization under Age of Information (AoI) constraints in energy-harvesting D2D-enabled cellular networks, demonstrating the stability and improved convergence of TD3 in stochastic and resource-constrained settings. Although this work highlights the advantages of TD3 over traditional actor–critic methods, it focuses on communication networks rather than energy management in smart buildings. Similarly, TD3-based approaches have been employed in residential microgrid energy management, achieving improved convergence and enhanced trade-offs between energy cost and user comfort [46]. Other works have investigated TD3 for power scheduling in distributed renewable energy systems and energy management in hybrid electric vehicles, demonstrating its robustness in handling continuous control tasks under uncertainty [47,48]. Moreover, recent extensions combine TD3 with optimization techniques, such as particle swarm optimization, to further enhance performance in energy harvesting and wireless network scenarios [49].

The authors in [50] investigate multi-agent reinforcement learning for coordinated energy management across multiple buildings, but rely on centralized training mechanisms, which may raise scalability and privacy concerns. On the federated learning side, recent works such as [51,52] have introduced personalized FL frameworks that decouple global model learning from local personalization using regularization techniques, including Moreau envelopes. While these methods provide strong theoretical guarantees and

improved personalization, their application to DRL-based control problems, particularly in continuous control environments such as HEMS, remains largely unexplored.

In practical energy communities, smart homes may operate under highly heterogeneous conditions, including different occupancy patterns, thermal dynamics, appliance usage profiles, and local renewable generation. This heterogeneity makes it challenging to learn a single global control policy that performs well across all participants, as traditional federated learning approaches such as FedAvg tend to average out local characteristics, leading to sub-optimal decisions for individual homes. At the same time, directly sharing raw energy consumption and behavioral data is undesirable due to privacy concerns and communication constraints. To address these challenges, we adopt a personalized federated deep reinforcement learning framework based on Moreau envelopes (pFedMe), which enables collaborative learning without exposing sensitive data while explicitly accounting for local differences. The key mechanism in pFedMe is to decouple global knowledge sharing from local model adaptation through a regularized optimization process, allowing each smart home to benefit from the collective experience of the community while maintaining a personalized control policy tailored to its own environment. This is particularly important in the considered energy communities, where the trade-off between energy cost and thermal comfort is inherently user-dependent, and thus requires individualized decision-making strategies. The proposed approach combines TD3-based continuous control with personalized federated learning for joint optimization of energy cost and indoor comfort, thereby addressing both system heterogeneity and privacy-preserving learning in smart home energy management. The key innovations of the present work include the following:

- We develop a comprehensive smart home energy management model that captures dynamic energy demand, renewable generation, electricity pricing, and indoor thermal dynamics, while incorporating an ESS for local energy balancing [7,18].
- We formulate a joint optimization problem for minimizing energy cost and maximizing indoor comfort, and solve it using a TD3 actor–critic framework for continuous control of HVAC operation and ESS scheduling.
- We extend the single-agent setting to an energy community scenario by proposing a PF-DRL learning framework based on personalized FL using Moreau envelopes (pFedMe), enabling collaborative learning while preserving user-specific preferences [52].
- We validate the proposed approach using real-world data, demonstrating improved convergence and performance over FedAvg and FedProx, achieving over 10% energy cost reduction while maintaining temperature deviations within 1.5 °C.

3. System Model

3.1. Smart Home Modeling

The modeling of an individual smart home environment is shown in Figure 1. Without loss of generality, it is assumed that the energy generators included in the smart home are solar generators that are used to gather power from the sunlight energy during the day. Moreover, the smart home also exhibits varying energy demands, i.e., energy consumption requests that reflect the normal operations of the households. To this end, three types of loads are present in a smart home: (i) non-shiftable requests that require the immediate provision of energy (e.g., lights, oven operation); (ii) shiftable and non-interruptible loads (e.g., the operation of a washing machine); (iii) controllable loads (e.g., the HVAC system) that can be scheduled or regulated over a period of time. Apart from the energy generators and loads, an ESS is also present in the environment, enabling dynamic charging or discharging depending on its capacity and the current electricity price. The ESS is used as a buffer to cover the local and immediate demands by providing its stored energy. In

addition, temperature sensors provide indications of the indoor temperature experienced inside the residence, as well as the outdoor (ambient) temperature.

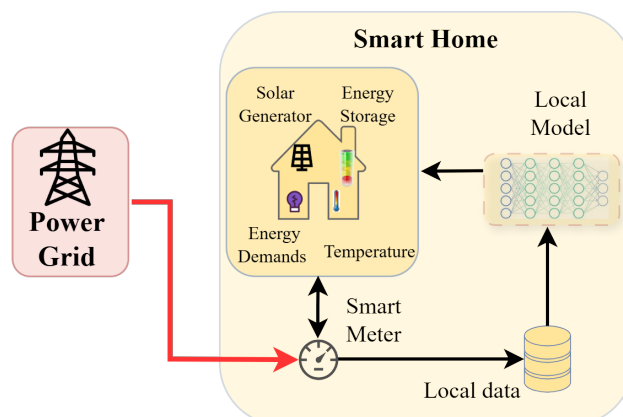


Figure 1. Smart home environment including the energy consumption demands, the energy generated from solar panels, the ESS, and the indoor and outdoor temperature, as well as the electricity price, which are included in the TD3 local model for training (the red arrow illustrates the power flow, whereas the black arrows the data flow).

The aforementioned data constitute the smart home environment and are conveyed to the smart meter in order to calculate the energy balance of the smart home. In case of a deficit (the local energy is not adequate for the loads of the smart home), the smart meter requests that the utility grid provide the missing energy at the current market price, which is usually a costly option. The temporal data gathered from the smart home can be stored in a local database, to be used for training a DRL model to act on the environment, i.e., regulate the input energy into the HVAC system (one of the most significant contributors to controllable loads), as well as the online operation of the ESS, i.e., draw or provide energy depending on the current market price, the energy demands of the household, the state of charge level of the storage system, etc.

In the context of mathematically formulating the optimization problem, we can denote the time-varying energy that is gathered by the solar panels as p_t and the fluctuating energy demands of the smart home (the sum of non-shiftable, non-interruptible and controllable requests apart from the HVAC and ESS control) as ω_t . It is worth noting that t is the time duration that is associated with the period of the provided measurements. Although the duration of each time slot is typically in the order of one hour, the described modeling can be applied to shorter time scales, depending on the required control granularity. Furthermore, the temporal evolution of the indoor temperature in a smart building environment can be expressed by [53–55]:

$$T_{t+1}^{in} = \epsilon T_t^{in} + (1 - \epsilon) \left(T_t^{out} - e_t \frac{\eta}{A} \right), \quad (1)$$

where T_t^{out} is the ambient temperature that is provided by the temperature sensors; ϵ is the inertia factor, which provides a trade-off between the indoor and the outdoor temperatures; η is the coefficient that represents the performance of the HVAC system (different for cooling and heating modes); and A is the thermal conductivity of the smart building measured in kW/°C. The mathematical symbols that are used in the present work are summarized in Table 1.

Table 1. Mathematical symbols.

Symbol	Description	Symbol	Description
ω_t	Smart home energy demands [kWh]	T_t^{out}	Ambient temperature [$^{\circ}$ C]
A	Thermal conductivity [kW/ $^{\circ}$ f]	ϵ	Inertia factor
η	HVAC performance coefficient	T_t^{in}	Indoor temperature [$^{\circ}$ C]
e_t	HVAC provided power [kW]	e^{max}	Peak HVAC power [kW]
T^{min}	Lower bound of T_t^{in} [$^{\circ}$ C]	T^{max}	Upper bound of T_t^{in} [$^{\circ}$ C]
C_t	ESS charge level [kWh]	δ_{σ}	Charging coefficient
σ_t	Charging rate [kW]	λ_t	Discharging rate [kW]
δ_{λ}	Discharging coefficient	C^{min}	Lower bound of C_t [kWh]
C^{max}	Upper bound of C_t [kWh]	σ^{max}	Upper bound of σ_t [kW]
λ^{max}	Upper bound of λ_t [kW]	p_t	Power from renewables [kW]
γ_t	Power from/to the grid [kW]	κ_t	Buying electricity price [\$/kW]
β_t	Selling electricity price [\$/kW]	ζ	Price scaling factor
v	ESS depreciation coefficient [\$/kW]	s_t	Smart home state
α_t	TD3 agent action	$\hat{e}_t$	Scaled HVAC input power
$\hat{\zeta}_t$	Scaled ESS charging/discharging variable	r_t	Reward function
ρ	Trade-off coefficient	$Q(s_t, \alpha_t)$	Action value function
$\mathcal{R}$	Replay memory	w^Q	Critic network weights
w^{ψ}	Actor network weights	M	Training episodes
N	Within-episode time slots	ν	Action exploration noise of deviation σ_n^2
Y	Mini-batch size	ν_c	Clipping limit of exploration noise
θ	Discount factor	N_{ψ}	Dimensions of the actor networks
N_{Q_1}	Dimensions of the critic networks	$\mathbf{w}$	Global FL policy
μ	Regularization coefficient for the personalization	L	Number of clients in the FL scheme
$\mathcal{L}$	TD3 loss for pFedMe	F	Number of federated rounds

Equation (1) indicates that the outdoor temperature, the power provided by the HVAC system e_t (scaled by the building parameters) and its previous value affect the current indoor temperature. Although more complex equations of the indoor temperature that accurately describe the building thermal dynamics can be used to provide a more realistic representation, it is worth noting that they can be effortlessly adapted in the presented approach by replacing this equation [56]. Taking into account the capability of the HVAC system, the provided power is constrained by a maximum value e^{max} . The restriction is expressed by:

$$0 \leq e_t \leq e^{max}, \quad (2)$$

where e^{max} is the peak power in Watts consumed by the HVAC system during the time slot t (typically one hour). An additional constraint that represents the comfort experienced by the smart home residents is related to the indoor temperature T_t^{in} , which must be within specific limits:

$$T^{min} \leq T_t^{in} \leq T^{max}. \quad (3)$$

To this end, the HEMS controls the HVAC power e_t (conforming to the relevant constraint) in the current time slot t through Equation (1) in order to regulate the indoor temperature T_{t+1}^{in} in the next time slot within the desired limits of Equation (3). Concerning the modeling of the ESS dynamics, the charge level in time slot t can be expressed by [7]

$$C_{t+1} = C_t + (\delta_{\sigma}\sigma_t + \frac{\lambda_t}{\delta_{\lambda}})\Delta t, \quad (4)$$

where $0 < \delta_\sigma, \delta_\lambda \leq 1$ are the charging and discharging coefficients, respectively, that reflect the quality of these operations. In practice, these coefficients measure the ratio of energy stored in a battery to the energy consumed/provided from/to the grid, accounting for energy losses (heat, conversion) and typically ranging from 85% to 95% for modern chargers. Moreover, the variables σ_t and λ_t denote the charging and discharging rates during the time slot t , measured in Ah or Wh. The charge level at a particular time slot is bounded within specific limits (adjusted by the capacity of the ESS) as follows:

$$C^{\min} \leq C_t \leq C^{\max}. \quad (5)$$

The ESS charging from the grid is bounded by the following constraint:

$$0 \leq \sigma_t \leq \sigma^{\max}, \quad (6)$$

while the discharging operation to the grid follows a similar constraint, as described in the following equation:

$$-\lambda^{\max} \leq \lambda_t \leq 0. \quad (7)$$

In Equation (6), we have denoted the charging variable as positive-valued with respect to the charging state, whereas in Equation (7) the discharge is a negative-valued parameter, since the latter operation reduces the current charge level of the ESS. Moreover, we assume that the ESS can be used either to charge or to discharge during a time slot t , i.e.,

$$\sigma_t \cdot \lambda_t = 0. \quad (8)$$

Aggregating the energy consumed by the residential loads, the HVAC and the ESS (in case of charge), as well as the energy produced by the solar panels and the ESS (in case of discharge), the household-level time-varying power balance equation can be expressed by:

$$p_t - \lambda_t + \gamma_t = \omega_t + \sigma_t + e_t. \quad (9)$$

The introduced variable γ_t in Equation (9) represents the power in kW that must be drawn from the utility grid in cases where the smart home exhibits a deficit for the current time slot ($\gamma_t > 0$), or the power that can be provided back to the grid in case of energy surplus ($\gamma_t < 0$). It should be noted that in the former scenario, the fluctuating market price (assuming that the customer is charged according to the online market price and not with a fixed rate) for buying the missing energy, denoted as κ_t , is typically higher than the selling electricity price β_t , exhibiting a linear relation:

$$\kappa_t = \xi \beta_t, \quad (10)$$

where ξ is the scaling factor and $\xi \geq 1$. To this end, the energy balance of the smart home indicates whether power will be bought from the grid at a higher price, inherently encouraging the use of local renewable sources in smart home energy consumption. It should be noted that this modeling approach considers the grid as an infinite source that always provides the missing energy to the energy community. In addition, a linear relationship is assumed between the buying and selling energy price for simplicity; more complex equations (e.g., non-linear) can be introduced, without affecting the proposed algorithm.

3.2. Energy Cost Minimization Problem

Following the presented modeling process, an optimization problem is formulated for each individual smart home, i.e., the minimization of the energy cost, respecting the various constraints that are imposed from the environment and the equipment. Thus, the minimization problem involves the cost paid directly from the grid when drawing energy (either to charge the ESS or to provide energy to the HVAC in order to control the indoor temperature), as well as the indirect equipment cost of the ESS, since its continuous operation incurs a depreciation cost. Formally, the energy cost that is bought from the grid after the internal balance of the smart home is $\kappa_t \gamma_t$ (in case $\gamma_t > 0$) and the cost related to the ESS operation is $v(\sigma_t + |\lambda_t|)$, where v is defined as a depreciation coefficient with units \$/kW. The latter term reflects the depreciation cost due to the continuous operation of the ESS system, i.e., the frequent charging or discharging operations incur a implied cost that is taken into consideration in the formulation of the energy cost minimization problem. The optimization problem can be expressed as:

$$(P_1) : \min_{e_t, \sigma_t, \lambda_t} \sum_{t=1}^S \mathbb{E}\{\kappa_t \gamma_t + v(\sigma_t + |\lambda_t|)\} \quad (11)$$

subject to (1)–(10), $\gamma_t > 0$,

where the expectation operator $\mathbb{E}$ is calculated for a temporal horizon of $t = 1, 2, \dots, S$ slots and the relevant time-varying environmental parameters, since their statistical distribution cannot be accurately described and can be considered random.

In principle, (P_1) is a very complex problem with multiple challenges: (i) the thermal dynamics inside the smart home environment are, in general, very complex and cannot be accurately modeled or described; (ii) the control variables for the minimization of the energy cost $(e_t, \sigma_t, \lambda_t)$ are not independent, since the ESS system can operate in one mode (either charging or discharging), influencing the power balance of the system and, thus, the power required for the HVAC; (iii) the power balance of the building involves multiple system parameters that cannot be known beforehand. For instance, the power gathered from renewable sources and the energy demands of a residential smart home typically follow periodical temporal patterns. However, the future values of these variables can be predicted with limited accuracy. Moreover, the energy market price exhibits strong temporal fluctuations and marginal forecasting abilities; (iv) the set of control actions that are decided in a time slot $(e_t, \sigma_t, \lambda_t)$ have an impact on future control decisions of the system, potentially leading to sub-optimal solutions. For instance, charging the ESS on a day with increased market prices may be the optimal solution in cases where, in the following days, the energy demands of the household are increased, and also reduced sunlight prohibits the collection of renewable energy. To this end, the optimization problem should be solved over a finite time horizon, where the variables describing the smart home environment are statistically unstable and not for individual time instances. Finally, dynamic parameters of the environment exhibit complex inter-dependencies that cannot be easily modeled. For instance, high energy market prices can lead to reduced energy consumption and vice versa (these variables are not controlled by the model).

(P_1) also involves conflicting optimization objectives, since the target is to minimize the energy cost, while at the same time respecting the indoor temperature constraints. The set of controlling parameters to be determined in each time slot t is the triplet $(e_t, \sigma_t, \lambda_t)$ that has a direct impact on the smart home indoor environment and (P_1) can be approximated as a decision-making Markov Decision Process (MDP) problem [7]. Specifically, most of the key smart home parameters of the $t + 1$ time slot depend on their values in the previous time step t and the performed actions, e.g., the ESS level C_{t+1} is calculated based only on

the current level C_t and the discharging/charging action (σ_t or λ_t). Similarly, the indoor temperature at the next time instance $t + 1$ depends on the indoor temperature at time t , the HVAC power input, the ambient temperature T_t^{out} and the building's characteristics. It is worth noting that some environmental parameters are not Markovian, i.e., they may follow specific patterns (e.g., increased energy consumption requests from the smart home during the afternoon) or long-range temporal correlations (e.g., renewable power generation surges during the summer months). However, (P_1) can be still approximated as an MDP and solved by reinforcement learning algorithms [57,58].

4. Reinforcement Learning Model and Proposed Algorithm

This section presents the key parameters of the RL-based actor-critic model that is incorporated in the HEMS of each smart home, as shown in Figure 1, and describes the PFDRL algorithm that is applied inside the energy community.

4.1. Twin-Delayed Deep Deterministic Policy Gradient Model

In order to solve the optimization problem (P_1) , the intelligent HEMS agent regulates two key parameters: the power provided to the HVAC system and the power that is charged/discharged to/from the ESS. When training the RL algorithm, the TD3 agent observes the smart home parameters (environment state) in each time slot, decides upon the input power to HVAC and ESS charge/discharge (action) and then receives positive or negative feedback when the selected action is applied to the smart home (reward). To this end, the intelligent agent is trained on a trial-and-error basis, slowly learning the most beneficial actions towards the optimization objective (minimization of the energy cost) [7,54,58]. The specific DRL agent is selected since it can deal with continuous states and actions (the state and action spaces are not required to be discrete) that are described below, along with the reward function.

4.1.1. Environment State

The state of the smart home that is observed by the intelligent agent, as shown in Figure 1, includes the dynamic environmental and system parameters and can be expressed as follows:

$$s_t = (p_t, \omega_t, C_t, T_t^{out}, T_t^{in}, \kappa_t, t'), \quad (12)$$

where the variables $p_t, \omega_t, C_t, T_t^{out}, T_t^{in}$ can be acknowledged to the agent through the smart meter and installed sensors, since they reflect on smart home-specific variables (energy consumption, energy generation, ESS level, ambient and indoor temperature), the current market price κ_t can be retrieved online (<https://data.nordpoolgroup.com/auction/day-ahead/prices?deliveryDate=latest¤cy=EUR&aggregation=Hourly&deliveryAreas=AT> accessed on 27 February 2026) and the variable t' denotes the time slot index in a day $t' = \text{mod}(t, 24)$.

4.1.2. Action of the TD3 Agent

As aforementioned, the purpose of the TD3 agent is to adjust the power provided to the HVAC system and the power that is charged/discharged to/from the ESS. The action can be described by:

$$\alpha_t = (\hat{e}_t, \hat{\zeta}_t), \quad (13)$$

where $0 \leq \hat{e}_t \leq 1$ is the scaled HVAC input power (divided by e^{max}) and $-1 \leq \hat{\zeta}_t \leq 1$ is the scaled variable for the ESS charging/discharging that can be computed as follows:

$$\begin{cases} \hat{\zeta}_t = \sigma_t / \sigma^{max}, & \text{if } \hat{\zeta}_t \geq 0 \\ \hat{\zeta}_t = -\lambda_t / \lambda^{max}, & \text{if } \hat{\zeta}_t < 0. \end{cases} \quad (14)$$

To this end, the suggested TD3 actions can be unscaled and implemented in the smart home environment. The HVAC input power can be unscaled according to (13) and the single variable $\hat{\zeta}_t$ that is required to denote the charging/discharging of the ESS can be unscaled according to (14).

4.1.3. Reward Function

Once the intelligent agent recommends the action vector α_t , it is applied to the smart home, leading to new environment state s_{t+1} in the next time slot, according to the building dynamics and the constraints described by (1)–(9). To quantify whether the selected action indeed provides the thermal comfort and cost benefits of the HEMS, a reward is calculated as:

$$r_t(s_t, \alpha_t) = -\left([T_{t+1}^{in} - T^{max}]^+ + [T^{min} - T_{t+1}^{in}]^-\right) - \rho(\kappa_t \gamma_t + v(\sigma_t + |\lambda_t|)). \quad (15)$$

In (15), the first term reflects the experienced comfort originating from the indoor temperature, i.e., the penalty (negative reward) is zero if it is within the specified limits and equals the deviation when the indoor temperature exceeds T^{max} or is below T^{min} . The second term represents the cost of the energy that is directly bought from the grid $\kappa_t \gamma_t$, in case that $\gamma_t > 0$ after the power budget calculation. In addition, we penalize the energy cost according to the optimization objectives of the problem and we do not provide a positive reward in case of $\gamma_t < 0$, i.e., the smart home exhibits energy surplus that can be sold back to the grid. Similarly, the third term of (15) is related to the ESS, since the charging/discharging operations incur a depreciation cost. Finally, ρ is a trade-off variable that balances the penalty due to the temperature deviation (typically in units of °C) and the energy cost negative reward (expressed in \$).

To this end, the learning process can be tuned to be more sensitive to the indoor comfort at the expense of the energy cost or vice versa. It should be noted that additional metrics that are involved in the indoor comfort can be included in the smart home model and the training process, such as the relative humidity. The dynamics of the humidity can be modeled similarly to the indoor temperature and an additional action of the TD3 agent can be added to regulate the humidity by configuring the HVAC system [56]. It is also worth noting that the energy cost penalty term of the reward function inherently boosts the use of renewable energy that is generated locally by each smart home, eliminating the dependency from the power grid.

As is evident from the action of the TD3 agent and the reward function during the learning process, the aim of the intelligent HEMS agent is to adjust the power level provided to the HVAC system, as well as the charging or discharging power for the ESS operation in each time instance to receive the maximum reward, as expressed in (15), also called the immediate reward. Specifically, the reward is maximized (ideally zero) when the DRL agent maintains the indoor temperature within the required limits and the energy cost that is related to the HVAC and the ESS operation (depreciation cost) are both minimized. To this end, the DRL formulation completely aligns with the optimization problem under investigation and its constraints, enabling the TD3 agent to provide beneficial solutions for (P_1). It is worth noting that the TD3 agent also inherently boosts the utilization of

the generated renewable energy to cover the individual demands of the smart home, minimizing the grid energy exchange and the associated cost.

4.2. Individual Smart Home Algorithm

This subsection details the algorithmic process of the individual TD3 agent training. As aforementioned, the TD3 algorithm belongs to the categories of actor/critic models that deal with problems involving continuous state and action spaces. Through the continuous interaction between the actor and critic components, the policy and value function of the Td3 algorithm are updated [59]. Specifically, the role of the actor neural network is to output the action α_t solely based on the environment state s_t , providing the most beneficial decision during the inference phase of the model. On the other hand, the utilization of the twin critic networks are limited to the training phase, receiving the state vector from the smart home environment and the suggested action from the actor network. Their role is to evaluate the action recommended by the actor through an action value function $Q_1(s_t, \alpha_t)$ and $Q_2(s_t, \alpha_t)$, respectively. The mechanisms of clipped double Q-learning (the minimum of the target action value functions is selected) and delayed policy updates (actor is not updated in each step) are used to calculate the policy gradient and update the weights of the actor network.

During the training phase, the individual TD3 agents interact with the smart home parameters on a trial-and-error basis for multiple episodes (different instantiations of the environment), i.e., observing the state, performing actions and receiving rewards. These tuples are stored in a memory experience replay which, along with the actor and critic target networks mechanisms, are used to update the weights of the networks [60]. The algorithmic process for the training of an individual smart home TD3 agent is illustrated in Algorithm 1. The required input data are the historical datasets of renewable power generation from the solar panels p_t , the non-shiftable and non-interruptible energy demands of the household ω_t , the time series of the outdoor ambient temperature T_t^{out} and the buying electricity price κ_t , data that concern the same temporal period.

The algorithm begins by initializing the replay memory $\mathcal{R}$ that hosts the pool of transition tuples $(s_t, \alpha_t, r_t, s_{t+1})$, the twin critic networks $Q(s_t, \alpha_t | w^{Q_1})$, $Q(s_t, \alpha_t | w^{Q_2})$ and the actor network $\psi(s_t | w^\psi)$. In addition, the target critic and actor networks are initialized by copying the weights from the original networks $w^{Q'_1} \leftarrow w^{Q_1}$, $w^{Q'_2} \leftarrow w^{Q_2}$ and $w^{\psi'} \leftarrow w^\psi$. The loop of the algorithm is then executed for M training episodes, where the environment is reset, i.e., a random time instance is selected in the dataset, representing different $p_t, \omega_t, T_t^{out}, \kappa_t$ variables. A random ESS level C_t and an indoor temperature T_t^{in} are also selected to constitute the state space s_t . For each training episode, the TD3 agent interacts with the environment in within-episode time slots N in order to become familiarized with the time-varying dependencies of the related parameters (e.g., patterns of outdoor temperature, generation of renewable energy). To this end, the agent selects an action α_t expressed by:

$$\alpha_t = \psi(s_t | w^\psi) + v, v \sim \mathcal{N}(0, \sigma_n^2). \quad (16)$$

According to (16), the action selected by the agent contains the exploitation term $\psi(s_t | w^\psi)$, which is the vector suggested by the actor network. This term is deviated by v , following a normal distribution with a variance of σ_n^2 , to introduce exploration noise to the selected action related both to the HVAC input power and the ESS charging/discharging. The selected action α_t is applied in the environment by calculating the dependent dynamic parameters using (1), (4), and (9) and the next state s_{t+1} can be acknowledged to the agent, along with the reward r_t according to (15). This transition tuple is then stored in the replay memory $\mathcal{R}$, as described in Algorithm 1.

Algorithm 1 Training of an individual TD3 agent

Input: Time series of renewable power generation, non-shiftable and non-interruptible energy demands, outdoor ambient temperature and buying electricity price.

Output: Actor network w^ψ and twin critic networks w^{Q_1}, w^{Q_2} of the TD3 model.

```

1: Random initialization of replay memory  $\mathcal{R}$  of size  $K$ .
2: Random initialization of actor network  $\psi(s_t|w^\psi)$  and twin critic networks  $Q_1(s_t, \alpha_t|w^{Q_1}), Q_2(s_t, \alpha_t|w^{Q_2})$ .
3: Initialize target networks:
4:  $w^{\psi'} \leftarrow w^\psi, w^{Q'_1} \leftarrow w^{Q_1}, w^{Q'_2} \leftarrow w^{Q_2}$ .
5: for  $ep = 1, 2, \dots, M$  do
6:   Random initialization of parameters  $C_t, T_t^{in}$  and time instance  $t$ , assemble state:
7:    $s_t = (p_t, \omega_t, C_t, T_t^{out}, T_t^{in}, \kappa_t, t')$ .
8:   for  $t = 1, 2, \dots, N$  do
9:     Select action with exploration noise:
10:     $\alpha_t = \psi(s_t|w^\psi) + \nu, \nu \sim \mathcal{N}(0, \sigma^2)$ 
11:    Apply action using (1), (4), (9), observe  $s_{t+1}$ .
12:    Compute reward  $r_t$  using (15).
13:    Store transition  $(s_t, \alpha_t, r_t, s_{t+1})$  in  $\mathcal{R}$ .
14:    Sample a random mini-batch of  $Y$  tuples  $(s_i, \alpha_i, r_i, s_{i+1})$  from  $\mathcal{R}, 1 \leq i \leq Y$ .
15:    Add clipped noise to target action:
16:     $\tilde{\alpha}_{i+1} = \psi'(s_{i+1}|w^{\psi'}) + \text{clip}(\nu, -c, c)$ 
17:    Compute target value:
18:     $y_i = r_i + \theta \min_{j=1,2} Q'_j(s_{i+1}, \tilde{\alpha}_{i+1})$ 
19:    Update both critics by minimizing:
20:     $\frac{1}{Y} \sum_i (Q_1(s_i, \alpha_i) - y_i)^2 + (Q_2(s_i, \alpha_i) - y_i)^2$ 
21:    if  $t \bmod d = 0$  then
22:      Update actor policy:
23:       $\nabla_{w^\psi} \frac{1}{Y} \sum_i Q_1(s_i, \psi(s_i|w^\psi))$ 
24:      Soft update target networks:
25:       $w^{Q'_j} \leftarrow \chi w^{Q_j} + (1 - \chi) w^{Q'_j}, j = 1, 2$ 
26:       $w^{\psi'} \leftarrow \chi w^\psi + (1 - \chi) w^{\psi'}$ 
27:    end if
28:  end for
29: end for

```

Following the algorithmic process, a mini-batch of Y transition tuples is sampled from the replay memory and used to calculate the expected state action values. To this end, the next action is calculated by:

$$\tilde{\alpha}_{i+1} = \psi'(s_{i+1}|w^{\psi'}) + \text{clip}(\nu, -\nu_c, \nu_c), \quad (17)$$

where the target actor network is used to provide the feedforward operation and the noise is clipped within the limits $[-\nu_c, \nu_c]$. Then, the target Q-value can be computed for both critic networks based on the Bellman equation [59,61]:

$$y_i = r_i + \theta \min_{j=1,2} Q'_j(s_{i+1}, \tilde{\alpha}_{i+1}), \quad (18)$$

where the target critic networks are used to compute the expected state action values and $\theta \in [0, 1]$ is the discount factor that balances between the immediate and future rewards. Notably, the future rewards of the Bellman Equation (18) provide the quality of the current

actions of the TD3 agent considering upcoming time instances of the complex environment. The weights of the critic networks are updated based on

$$\frac{1}{Y} \sum_i (Q_1(s_i, \alpha_i) - y_i)^2 + (Q_2(s_i, \alpha_i) - y_i)^2 \quad (19)$$

and the sampled policy gradient is calculated once every policy delay time, denoted d for each update of the Q-networks:

$$\nabla_{w^\psi} \frac{1}{Y} \sum_i Q_1(s_i, \psi(s_i|w^\psi)) \quad (20)$$

is used to update the actor policy. Finally, the target critic and actor networks are updated by using $w^{Q'_j} \leftarrow \chi w^{Q_j} + (1 - \chi)w^{Q'_j}$, $j = 1, 2$ and $w^{\psi'} \leftarrow \chi w^\psi + (1 - \chi)w^{\psi'}$, while the update variable χ is typically small ($\chi \ll 1$) to ensure the soft update of the target networks. The architectures of the actor and critic networks used in this work are adopted from the DDPG implementation in [38], including two hidden layers for the actor network and four hidden layers for the critic networks.

The computational complexity of the proposed TD3-based HEMS algorithm is primarily driven by the forward and backward passes of the neural networks during training. At each time step, the agent performs inference through the actor and twin critic networks, followed by gradient-based updates using a mini-batch of size Y . Since, in this work, we use fully connected networks with N_ψ , N_{Q_1} , N_{Q_2} parameters for the actor and the two critic networks respectively, the per-update complexity is $\mathcal{O}(Y(N_\psi + 2N_Q))$, assuming that the two critic networks are identical and $N_{Q_1} = N_{Q_2} = N_Q$. Since the TD3 framework employs twin critics and delayed policy updates, the critic networks are updated at every step, while the actor and target networks are updated every d steps, slightly reducing the overall computational burden compared to standard actor–critic methods with synchronous updates. Over a full training process consisting of M episodes and N time steps per episode, the total computational complexity scales as $\mathcal{O}(MNY(N_\psi + 2N_Q))$. The use of experience replay ensures that updates are independent of the trajectory length, while the clipping and noise injection operations introduce negligible overhead.

4.3. Federated Learning Algorithmic Process

As aforementioned, the trained ML models of the individual TD3 agents are fused in a global ML model using FL. The architecture of the FL process is shown in Figure 2, where multiple smart homes that are inside the same energy community train their TD3 agents. Evidently, all community members (smart homes) have direct access to the power grid that covers their immediate energy deficit. It should be noted that the missing energy provided by the grid is typically the most expensive option, since it is the highest order of energy providers. Furthermore, a special logical entity is the Community Orchestrator (CO) that enables energy exchange between different communities through horizontal communication, and also serves as a message bus that conveys information between the various smart homes within the community. This information includes, for instance, energy plans or energy surpluses/deficits that are experienced by the community members. In the proposed scenario, we consider that the information that is conveyed from/to the community households to the CO are the updated neural network weights of the TD3 agents.

As an FL mechanism, pFedMe is adopted to deal with heterogeneous data originating from different smart homes. pFedMe enables a client-specific personalized control policy, using Moreau envelopes as client's regularized loss, while maintaining coordination through a shared global model [52]. At each federated round, each smart home locally

optimizes its TD3 policy using a proximal objective that penalizes deviation from the global policy. The server residing in the CO then aggregates the personalized models to update the parameters of the global model that are acknowledged back to the FL participants. In our implementation, the pFedMe proximal term is applied to both actor and critic parameters, yielding personalized policies and value functions while maintaining coordination through the global model. The global FL policy for all smart home clients is described by the vector $\mathbf{w} = \{w^\psi, w^{Q_1}, w^{Q_2}\}$ that includes the global actor network and the twin critic networks of the TD3 model.

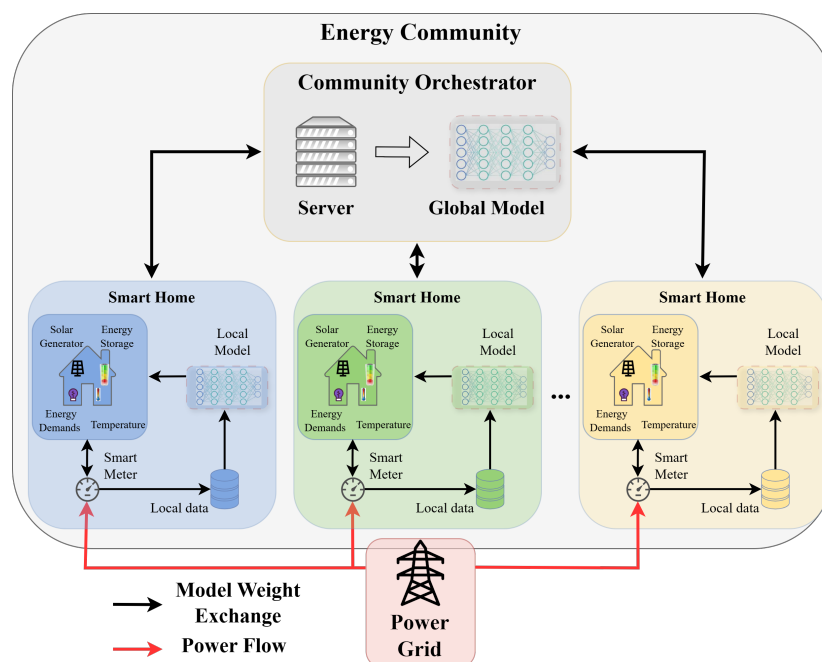


Figure 2. FL scheme among the smart homes inside an energy community, illustrating the fusion of ML model weights in the Community Orchestrator.

The FL algorithm is shown in Algorithm 2 and requires as input the smart home-specific data that are involved in Algorithm 1 from all the FL participants. After the initialization of the global policy $\mathbf{w}^0$, the algorithm is executed for $f = 1, 2, \dots, F$ federated rounds. In each FL round, each smart home/FL client receives the global model $\mathbf{w}^{f-1}$ and then performs model training based on the local dataset, i.e., executes Algorithm 1 and updates the local TD3 model, aiming to minimize a Moreau envelope of the client objectives, allowing for controlled personalization via the regularization coefficient μ :

$$\mathbf{w} \leftarrow \arg \min_{\mathbf{w}} \mathcal{L}_\ell(w_\ell^\psi, w_\ell^{Q_1}, w_\ell^{Q_2}) + \frac{\mu}{2} \left(\|w_\ell^\psi - w^\psi\|^2 + \|w_\ell^{Q_1} - w^{Q_1}\|^2 + \|w_\ell^{Q_2} - w^{Q_2}\|^2 \right), \quad (21)$$

where $\ell = 1, 2, \dots, L$ are the number of clients in the FL scheme and $\mathcal{L}_\ell(w_\ell^\psi, w_\ell^{Q_1}, w_\ell^{Q_2})$ is the TD3 loss for the ℓ^{th} client. Finally, the average of the actor and critic network parameters is utilized to calculate the global policy for this federated round. In this context, each smart home learns its own policy, while the global model acts as a soft regularizer and the parameter μ controls the personalization vs. federation trade-off. It is worth noting that the overall complexity in the federated setting increases linearly with the number of participating clients $\mathcal{O}(L)$, although communication costs are limited to periodic model aggregation. Overall, the proposed approach remains computationally tractable for real-time HEMS applications, as the dominant operations are matrix multiplications that can be efficiently accelerated on modern hardware.

Algorithm 2 pFedMe for Federated Smart Home Control**Input:** Algorithm 1 and related data for all participants/smart homes.**Output:** Global policy $\mathbf{w} = \{w^\psi, w^{Q_1}, w^{Q_2}\}$ that includes the global actor network and the twin critic networks of the TD3 model.

```

1: Random initialization of global policy  $\mathbf{w}^0$ 
2: for each federated round  $f = 1, \dots, F$  do
3:   for each client  $\ell$  in parallel do
4:     Receive  $\mathbf{w}^{f-1} = \{w^\psi, w^{Q_1}, w^{Q_2}\}^{f-1}$ 
5:     Execute Algorithm 1 with loss calculated by Equation (21)
6:   end for
7:    $\mathbf{w}^f \leftarrow \frac{1}{L} \sum_{\ell=1}^L \{w_\ell^\psi, w_\ell^{Q_1}, w_\ell^{Q_2}\}$ 
8: end for

```

5. Results*5.1. Individual TD3 Model Training*

This subsection details the training details of a single smart home TD3 HEMS agent, focusing on the stabilization hyperparameters and the environmental variables. The simulation results presented herein are based on real data obtained from an open household dataset. Specifically, the time series of energy consumption and generation by renewable sources originating from several households, as well as the ambient temperature from the region of Germany, can be retrieved online (https://data.open-power-system-data.org/household_data/ accessed on 27 February 2026), along with the time series of the energy market. The rest of the parameters that are required to model the dynamic smart home environment (e.g., thermal building dynamics) are tabulated in Table 2, according to typical range values found in the literature [7,38]. Without the loss of generality, the ESS maximum capacity is adjusted to 10 kWh, the maximum power provided to the HVAC system is 8 kW, the comfortable temperature bounds are between 19.5 and 22.5 °C, and $\eta = -1$ is used for the value of the thermal conversion efficiency [53,54]. It is worth mentioning that these environmental parameters can be, in principle, different for each smart home, necessitating the use of personalized FL in the training process. In addition, Table 2 tabulates the learning parameters of the TD3 model, as described in Algorithm 1, that have been fine-tuned via simulations.

Table 2. Simulation parameters of the smart home environment and the TD3 learning algorithm.

Environment Parameters	Value	Learning Parameters	Value
ϵ	0.7	σ_n^2	0.2
η	−1	M	2000
A (kW/°f)	0.14	N	24
e^{max} (kW)	8	K	35,000
T^{max} (°C)	22.5	Y	32
T^{min} (°C)	19.5	c	0.5
C^{max} (kWh)	10	θ	0.99
C^{min} (kWh)	0.6	χ	10^{-3}
$\delta_\sigma, \delta_\lambda$	0.95	N_ψ	(300, 600)
$\sigma^{max}, \lambda^{max}$ (kW)	3	N_{Q_1}, N_{Q_2}	(300, 600, 600, 600)
ξ	1.1	F	20
v (\$/kW)	1	μ	10^{-3}

The resulting fine-tuned learning curves are shown in Figure 3 for a training session consisting of $M = 2000$ training episodes and $N = 24$ inter-episode steps. In each episode, the TD3 agent interacts with the dynamic environment for 24 consecutive hours, aiming to learn the temporal dependencies of the associated parameters. The environment is reset in

the next training episode at a random day and the TD3 agent interacts with the environment in a trial-and-error manner. According to the learning curves, the selected values are $L_{r,ac} = 10^{-3}$ for the actor network and $L_{r,cr} = 10^{-4}$ for the critic networks, whereas these results concern a balanced optimization scenario between the thermal comfort and the energy cost ($\rho = 10$). Additional TD3 agents can be trained with different optimization objectives in the reward function for comfort-aware HEMS ($\rho = 0$) and cost-aware HEMS ($\rho = 20$), as depicted in Figure 3, depending on the specific requirements of the household resident. In the former case, the episode reward converges approximately to zero, since the TD3 agent learns to maintain the temperature within the requested limits regardless of the energy cost (HVAC system and ESS). On the other hand, the cost-aware HEMS agent prefers the temperature penalty in favor to the energy cost, converging to lower values. In principle, training the HEMS agent with $\rho = 10$ is considered an adequate trade-off between the indoor comfort and the energy cost. It is worth mentioning that the performance of the TD3 agents are not compared to additional baseline methods herein, since previous studies have consistently reported that actor/critic techniques significantly increase the benefits for HEMS control compared to classical approaches (e.g., cognitive rule-based approaches, as well as DRL methods with discrete action space) in terms of energy cost and thermal comfort [7,38].

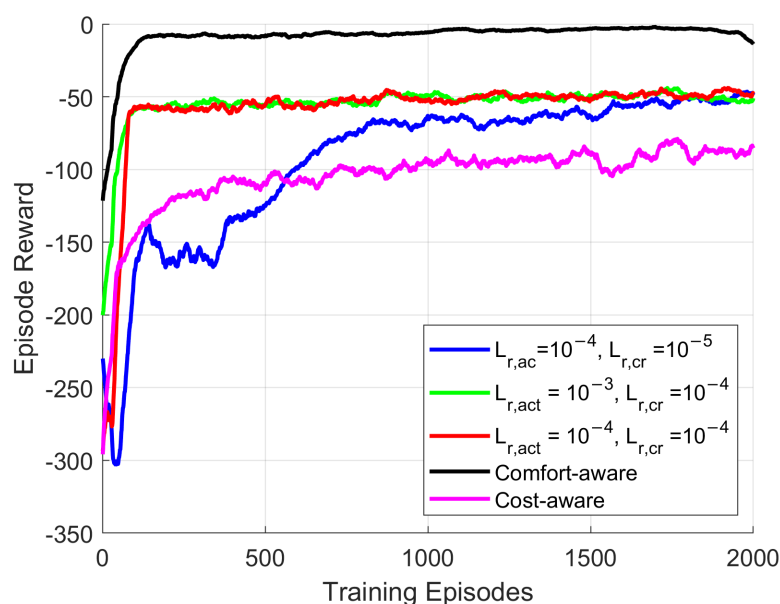


Figure 3. Learning curves for an individual TD3 HEMS agent for different values of the actor and critic learning rates, as well as different optimization objectives (cost-aware, balanced, comfort-aware).

5.2. FL Training of TD3 Agents Within the Energy Community

Regarding the FL configuration of multiple smart home, Table 3 provides a qualitative comparison of the considered FL schemes in this work, highlighting their key characteristics in terms of personalization capability, robustness to data heterogeneity, and computational complexity. First, the FedAvg algorithm relies on simple model averaging and assumes homogeneous data distributions across clients, which limits its ability to adapt to the diverse operating conditions of different smart homes. Moreover, the FedProx approach partially addresses this issue by introducing a proximal regularization term that stabilizes local updates and improves convergence under heterogeneous data, yet it still enforces a single global model. In contrast, the proposed approach is based on the pFedMe method, explicitly enabling personalization by decoupling global and local objectives through Moreau envelope-based regularization, allowing each client to learn a model tailored to its

specific environment while still benefiting from shared knowledge. This flexibility comes at the cost of increased computational complexity due to the additional optimization steps required for model personalization. Overall, pFedMe offers a more suitable framework for energy communities with heterogeneous participants, where individualized control policies are essential for achieving optimal performance.

Table 3. Qualitative comparison of FL schemes.

Property	FedAvg	FedProx	pFedMe
Global model sharing	✓	✓	✓
Personalized models	✗	✗	✓
Handles data heterogeneity	Low	Medium	High
Regularization to global model	✗	✓ (proximal term)	✓ (Moreau envelope)
Decoupling global/local objectives	✗	Partial	✓
Convergence stability	Medium	High	High
Adaptation to local preferences	Low	Medium	High
Communication efficiency	High	High	Medium

In this context, extending the training in the FL configuration of the energy community consisting of three smart homes, the learning curves for different FL algorithms are presented in Figure 4.

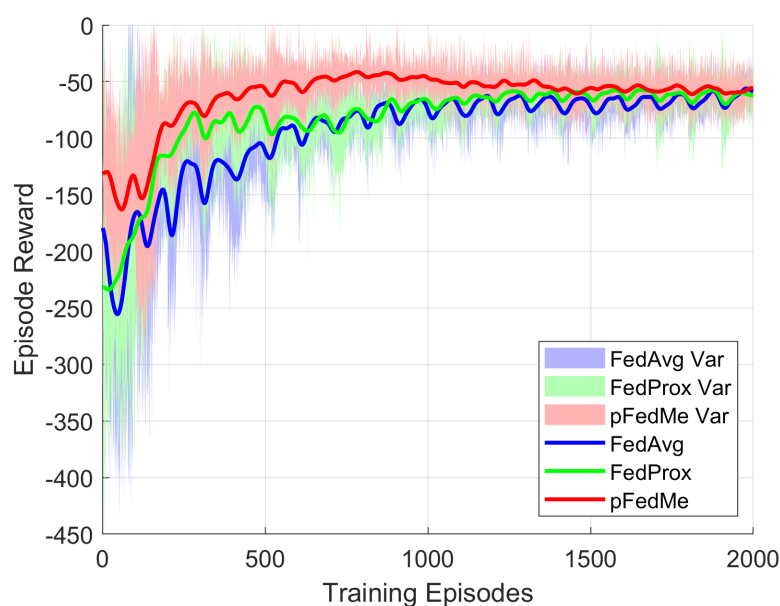


Figure 4. Training convergence for different FL algorithms, namely, FedAvg, FedProx and pFedMe for a configuration of three smart homes. The mean value and the variance of the episode reward between the FL participants are illustrated.

All the FL algorithms run for $F = 20$ federated learning rounds, while keeping the total number of training episodes $M = 2000$. In addition, the regularization coefficient is fine-tuned to the value of $\mu = 10^{-3}$ for the pFedMe and $\mu = 10^{-5}$ for the FedProx implementation and the three schemes have been trained for the balanced optimization scenario of $\rho = 10$. Evidently, the pFedMe exhibits better learning behavior compared to the two baseline algorithms, since the mean episode reward converges after approximately 500 training episodes and remains close to the reward of the individual TD3 agent.

5.3. Evaluation Results of the FL Schemes

To evaluate the impact of the personalized FL algorithm and assess the performance of the trained HEMS agents residing in the smart homes, the individual TD3 models are tested with real data that were not utilized during the learning process. In this context, the agents are tested with data for 1 month that was not encountered during the training (namely, the ambient outdoor temperature, the renewable power generation, the energy consumption and the buying electricity price time series), while randomly initializing each smart home environment (ESS level, indoor temperature and time instance). It is worth noting that only the actor networks are used during the inference (evaluation) phase, since the output of the critic networks is required only during TD3 agent training to provide feedback related to the action selected by the actor network and update its policy.

The house-specific TD3 actor networks are inferred with the evaluation data in terms of two metrics: the average temperature violation in the inference period of 1 day ($N = 24$) and the average daily energy cost. Regarding the temperature violation key performance indicator (KPI), the simulation results depicted in Figure 5 verify the efficiency of the pFedMe algorithm compared with the baselines of the FedAvg, FedProx and pFedMe global model. Notably, the latter baseline refers to the pFedMe model that has not conducted the final personalization steps during the FL process. Evidently, the pFedMe personalized TD3 agents achieve a temperature violation below $1.5\text{ }^{\circ}\text{C}$ for all three houses, being marginally outperformed by FedAvg in house 1, as well as FedProx and pFedMe global in house 2.

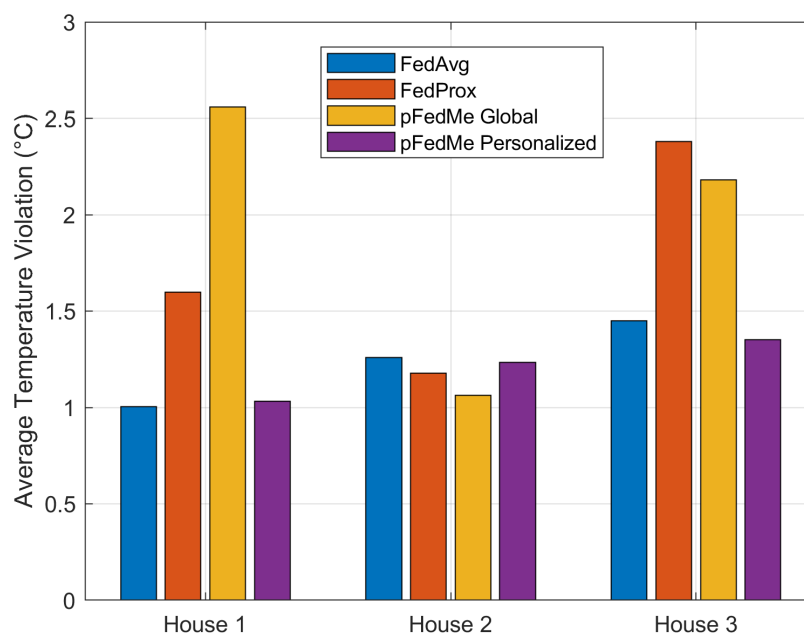


Figure 5. Average temperature violation for the different FL algorithms and the three smart homes inside the energy community.

In addition, Figure 5 illustrates the second KPI, i.e., the average daily energy cost for the different FL algorithms and the different households, where the pFedMe personalized models outperform all the baselines in all three houses. Specifically, the minimum reduction in the energy cost compared to FedAvg is approximately 37% for the first household, approximately 35% for the third smart home compared to Fedprox and less than 10% for the second household compared to the average daily energy cost provided by the pFedMe global model.

Overall, the superior performance of the personalized pFedMe scheme can be attributed to its ability to explicitly account for statistical heterogeneity across smart homes. In contrast to FedAvg, which enforces a single global model and may lead to sub-optimal decisions when local environments differ, and FedProx, which only constrains local updates through a proximal term, pFedMe introduces a Moreau envelope-based regularization that decouples global model learning from local personalization. This allows each agent to converge towards a solution that is both close to the global model and well-adapted to its local dynamics, such as distinct consumption patterns, thermal characteristics, and renewable generation profiles. As a result, the personalized policies learned by pFedMe achieve a better trade-off between energy cost and thermal comfort, leading to improved overall performance compared to non-personalized federated approaches. This behavior is also reflected in the reduced temperature violations and lower energy cost observed in Figures 5 and 6.

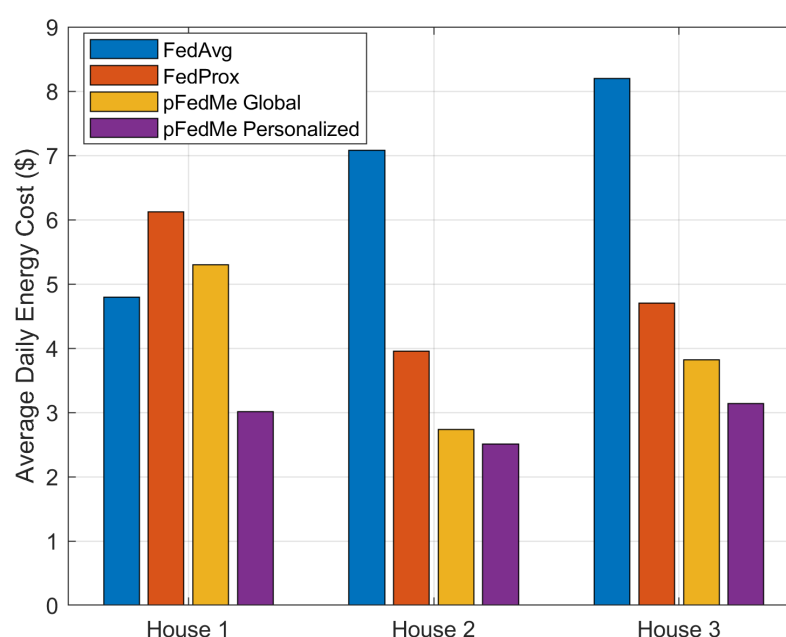


Figure 6. Average daily energy cost for the different FL algorithms and the three smart homes inside the energy community.

6. Discussion and Conclusions

The present work aims to extend the intelligent HEMS agent from a single smart home to an energy community level, where the participating clients/end devices collaborate in an FL scheme to exchange intelligence distilled from the local data, environmental parameters and historical trial-and-error decisions. To this end, we present how the collaborative training of multiple TD3 agents can be conducted in a PFDRL framework, using the Community Orchestrator as the aggregation entity that performs the fusion of the global model. We differentiate from previous studies by proposing a personalized FL algorithm that uses Moreau envelopes (pFedMe), decoupling the personalized model optimization from the global model learning. The introduced pFedMe method manages to maintain an optimal trade-off between the generalization capabilities offered by the typical FL mechanism and personalized model training that may result in overfitting. In other words, the intelligent agent of an individual smart home trained in the presented scheme copes with heterogeneous data that may be encountered due to the FL process and at the same time provides improved performance for the local environment. The simulation

results clearly illustrate the performance capabilities of the TD3 personalized agents trained in the pFedMe scheme compared to the baselines of the FedAvg, Fedprox and pFedMe global model outputs. Specifically, the pFedMe scheme achieves a maximum temperature violation below 1.5 °C for the three houses in the FL scheme, exhibiting similar results with the FedAvg and FedProx frameworks. In addition, the pFedMe personalized agents outperform the baseline FL schemes in all three houses regarding the average daily energy cost, demonstrating an improvement of at least 37%, 35% and 10% for the three residential houses, respectively.

Regarding future work, we aim to include different thermal comfort variables in the modeling of the smart home (e.g., humidity), complementing also the control actions of the TD3 agent. Moreover, we plan to extend this approach in energy communities that also exchange directly locally produced energy to achieve a zero-sum energy footprint. Finally, other than including smart homes in the FL scheme, we aim to incorporate different types of participating clients/end devices (e.g., EV chargers) in the FL scheme that may act as prosumers in the energy community.

Author Contributions: Conceptualization, S.S. and A.G.; methodology, S.S. and A.G.; software, S.S.; validation, S.S. and A.G.; formal analysis, S.S. and A.G.; investigation, S.S. and A.G.; resources, S.S. and A.G.; data curation, S.S.; writing—original draft preparation, S.S.; writing—review and editing, S.S.; visualization, S.S. and A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The following open-source datasets were used in this work: (1) Data related to the energy management of residential households can be found here: https://data.open-power-system-data.org/household_data/ accessed on 27 February 2026. (2) Data related to the energy market can be found here: <https://data.nordpoolgroup.com/auction/day-ahead/prices?deliveryDate=latest¤cy=EUR&aggregation=Hourly&deliveryAreas=AT> accessed on 27 February 2026.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-Ghaili, A.M.; Kasim, H.; Al-Hada, N.M.; Jørgensen, B.N.; Othman, M.; Wang, J. Energy management systems and strategies in buildings sector: A scoping review. *IEEE Access* **2021**, *9*, 63790–63813. [\[CrossRef\]](#)
2. Hannan, M.A.; Faisal, M.; Ker, P.J.; Mun, L.H.; Parvin, K.; Mahlia, T.M.I.; Blaabjerg, F. A review of Internet of Energy based building energy management systems: Issues and recommendations. *IEEE Access* **2018**, *6*, 38997–39014. [\[CrossRef\]](#)
3. Jing, T.; Zhao, Y. Optimizing energy consumption in smart buildings: A model for efficient energy management and renewable integration. *Energy Build.* **2024**, *323*, 114754. [\[CrossRef\]](#)
4. Poyyamozi, M.; Murugesan, B.; Rajamanickam, N.; Shorfuzzaman, M.; Aboelmagd, Y. IoT—A promising solution to energy management in smart buildings: A systematic review, applications, barriers, and future scope. *Buildings* **2024**, *14*, 3446. [\[CrossRef\]](#)
5. Constantinou, S.; Costa, C.; Konstantinidis, A.; Chrysanthos, P.K.; Zeinalipour-Yazti, D. A sustainable energy management framework for smart homes. *IEEE Trans. Sustain. Comput.* **2025**, *10*, 70–81. [\[CrossRef\]](#)
6. Yu, L.; Qin, S.; Zhang, M.; Shen, C.; Jiang, T.; Guan, X. A review of deep reinforcement learning for smart building energy management. *IEEE Internet Things J.* **2021**, *8*, 12046–12063. [\[CrossRef\]](#)
7. Yu, L.; Xie, W.; Xie, D.; Zou, Y.; Zhang, D.; Sun, Z.; Zhang, L.; Zhang, Y.; Jiang, T. Deep reinforcement learning for smart home energy management. *IEEE Internet Things J.* **2020**, *7*, 2751–2762. [\[CrossRef\]](#)
8. Aldahmashi, J.; Ma, X. Real-time energy management in smart homes through deep reinforcement learning. *IEEE Access* **2024**, *12*, 43155–43172. [\[CrossRef\]](#)
9. Lee, S.; Choi, D.-H. Reinforcement learning-based energy management of smart home with rooftop solar photovoltaic system, energy storage system, and home appliances. *Sensors* **2019**, *19*, 3937. [\[CrossRef\]](#)

10. Kodama, N.; Harada, T.; Miyazaki, K. Home energy management algorithm based on deep reinforcement learning using multistep prediction. *IEEE Access* **2021**, *9*, 153108–153115. [\[CrossRef\]](#)
11. Yu, L.; Jiang, T.; Zou, Y. Online energy management for a sustainable smart home with an HVAC load and random occupancy. *IEEE Trans. Smart Grid* **2019**, *10*, 1646–1659. [\[CrossRef\]](#)
12. Valladares, W.; Galindo, M.; Gutiérrez, J.; Wu, W.-C.; Liao, K.-K.; Liao, J.-C.; Lu, K.-C.; Wang, C.-C. Energy optimization associated with thermal comfort and indoor air control via a deep reinforcement learning algorithm. *Build. Environ.* **2019**, *155*, 105–117. [\[CrossRef\]](#)
13. Amer, A.A.; Shaban, K.; Massoud, A.M. DRL-HEMS: Deep reinforcement learning agent for demand response in home energy management systems considering customers and operators perspectives. *IEEE Trans. Smart Grid* **2023**, *14*, 239–250. [\[CrossRef\]](#)
14. Anvari-Moghaddam, A.; Monsef, H.; Rahimi-Kian, A. Optimal smart home energy management considering energy saving and a comfortable lifestyle. *IEEE Trans. Smart Grid* **2015**, *6*, 324–332. [\[CrossRef\]](#)
15. Christopoulos, M.; Spantideas, S.; Giannopoulos, A.; Trakadas, P. Deep reinforcement learning for smart home temperature comfort in IoT-edge computing systems. In Proceedings of the 1st International Workshop MetaOS Cloud-Edge-IoT Continuum (MECC'24), Athens, Greece, 22–25 April 2024; pp. 1–7. [\[CrossRef\]](#)
16. Zhang, D.; Li, S.; Sun, M.; O'Neill, Z. An optimal and learning-based demand response and home energy management system. *IEEE Trans. Smart Grid* **2016**, *7*, 1790–1801. [\[CrossRef\]](#)
17. Gupta, A.; Badr, Y.; Negahban, A.; Qiu, R.G. Energy-efficient heating control for smart buildings with deep reinforcement learning. *J. Build. Eng.* **2021**, *34*, 101739. [\[CrossRef\]](#)
18. Gao, G.; Li, J.; Wen, Y. DeepComfort: Energy-efficient thermal comfort control in buildings via reinforcement learning. *IEEE Internet Things J.* **2020**, *7*, 8472–8484. [\[CrossRef\]](#)
19. Zhang, C.; Liu, Y.; Wu, F.; Tang, B.; Fan, W. Effective charging planning based on deep reinforcement learning for electric vehicles. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 542–554. [\[CrossRef\]](#)
20. Tang, X.; Chen, J.; Liu, T.; Qin, Y.; Cao, D. Distributed deep reinforcement learning-based energy and emission management strategy for hybrid electric vehicles. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9922–9934. [\[CrossRef\]](#)
21. Hu, X.; Liu, T.; Qi, X.; Barth, M. Reinforcement learning for hybrid and plug-in hybrid electric vehicle energy management: Recent advances and prospects. *IEEE Ind. Electron. Mag.* **2019**, *13*, 16–25. [\[CrossRef\]](#)
22. Zhang, F.; Yang, Q.; An, D. CDDPG: A deep-reinforcement-learning-based approach for electric vehicle charging control. *IEEE Internet Things J.* **2021**, *8*, 3075–3087. [\[CrossRef\]](#)
23. Saad, A.; Youssef, T.; Elsayed, A.T.; Amin, A.; Abdalla, O.H.; Mohammed, O. Data-centric hierarchical distributed model predictive control for smart grid energy management. *IEEE Trans. Ind. Inform.* **2019**, *15*, 4086–4098. [\[CrossRef\]](#)
24. Zheng, W.; Lu, H.; Zhang, M.; Wu, Q.; Hou, Y.; Zhu, J. Distributed energy management of multi-entity integrated electricity and heat systems: A review of architectures, optimization algorithms, and prospects. *IEEE Trans. Smart Grid* **2024**, *15*, 1544–1561. [\[CrossRef\]](#)
25. Tian, P.; Xiao, X.; Wang, K.; Ding, R. A hierarchical energy management system based on hierarchical optimization for microgrid community economic operation. *IEEE Trans. Smart Grid* **2016**, *7*, 2230–2241. [\[CrossRef\]](#)
26. Yan, L.; Chen, X.; Chen, Y.; Wen, J. A hierarchical deep reinforcement learning-based community energy trading scheme for a neighborhood of smart households. *IEEE Trans. Smart Grid* **2022**, *13*, 4747–4758. [\[CrossRef\]](#)
27. Mahmoud, M.; Slama, S.B. Peer-to-peer energy trading case study using an AI-powered community energy management system. *Appl. Sci.* **2023**, *13*, 7838. [\[CrossRef\]](#)
28. Chen, T.; Su, W. Local energy trading behavior modeling with deep reinforcement learning. *IEEE Access* **2018**, *6*, 62806–62814. [\[CrossRef\]](#)
29. Levis, G.A.; Spantideas, S.T.; Giannopoulos, A.E.; Trakadas, P. A peer-to-peer energy management and exchange framework in energy communities via actor-critic learning. *TechRxiv* **2025**. [\[CrossRef\]](#)
30. Carli, R.; Dotoli, M.; Pellegrino, R. A hierarchical decision-making strategy for the energy management of smart cities. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 505–523. [\[CrossRef\]](#)
31. Hong, Y.-Y.; Alano, F.I. Hierarchical energy management in islanded networked microgrids. *IEEE Access* **2022**, *10*, 8121–8132. [\[CrossRef\]](#)
32. Lee, S.; Choi, D.-H. Federated reinforcement learning for energy management of multiple smart homes with distributed energy resources. *IEEE Trans. Ind. Inform.* **2022**, *18*, 488–497. [\[CrossRef\]](#)
33. Xu, B.; Zhao, H.; Cao, H.; Garg, S.; Alrashoud, M. Long-term energy management empowered hierarchical federated learning for smart consumer electronics. *IEEE Trans. Consum. Electron.* **2024**, *70*, 1155–1166. [\[CrossRef\]](#)
34. Li, Y.; Chen, X.; Wang, Y. EdgeHEM: Sparse federated reinforcement learning for home energy management at the edge. *IEEE Trans. Smart Grid* **2025**, *16*, 5602–5614. [\[CrossRef\]](#)
35. Lu, Z.; Pan, H.; Dai, Y.; Si, X.; Zhang, Y. Federated learning with non-IID data: A survey. *IEEE Internet Things J.* **2024**, *11*, 19188–19209. [\[CrossRef\]](#)

36. Giannopoulos, A.E.; Spantideas, S.T.; Zetas, M.; Nomikos, N.; Trakadas, P. FedShip: Federated over-the-air learning for communication-efficient and privacy-aware smart shipping in 6G communications. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 19873–19888. [\[CrossRef\]](#)
37. Latoń, D.; Grela, J.; Ożadowicz, A. Applications of deep reinforcement learning for home energy management systems: A review. *Energies* **2024**, *17*, 6420. [\[CrossRef\]](#)
38. Spantideas, S.T.; Giannopoulos, A.E.; Trakadas, P. Autonomous price-aware energy management system in smart homes via actor–critic learning with predictive capabilities. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 15018–15033. [\[CrossRef\]](#)
39. Ding, X.; Cerpa, A.; Du, W. Exploring deep reinforcement learning for holistic smart building control. *ACM Trans. Sens. Netw.* **2024**, *20*, 70. [\[CrossRef\]](#)
40. Christensen, M.H.; Ernewein, C.; Pinson, P. Demand response through price-setting multi-agent reinforcement learning. In *Proceedings of the 1st International Workshop Reinforcement Learning for Energy Management in Buildings & Cities (RLEM'20)*, Online, 17 November 2020; ACM: New York, NY, USA, 2020; pp. 1–5. [\[CrossRef\]](#)
41. Wei, Q.; Liao, Z.; Shi, G. Generalized actor–critic learning optimal control in smart home energy management. *IEEE Trans. Ind. Inform.* **2021**, *17*, 6614–6623. [\[CrossRef\]](#)
42. Rezazadeh, F.; Bartzoudis, N. A federated DRL approach for smart micro-grid energy control with distributed energy resources. In *Proceedings of the IEEE 27th International Workshop Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, Paris, France, 2–3 November 2022. [\[CrossRef\]](#)
43. Lee, S.; Xie, L.; Choi, D.-H. Privacy-preserving energy management of a shared energy storage system for smart buildings: A federated deep reinforcement learning approach. *Sensors* **2021**, *21*, 4898. [\[CrossRef\]](#)
44. Chen, Z.; He, J.; Yu, L.; Zhao, Q. Personalized federated deep reinforcement learning for smart home energy management. *IFAC-PapersOnLine* **2025**, *59*, 193–198. [\[CrossRef\]](#)
45. Liu, X.; Xu, J.; Zheng, K.; Zhang, G.; Liu, J.; Shiratori, N. Throughput maximization with an AoI constraint in energy harvesting D2D-enabled cellular networks: An MSRA-TD3 approach. *IEEE Trans. Wirel. Commun.* **2025**, *24*, 1448–1466. [\[CrossRef\]](#)
46. Wang, C.; Zhang, J.; Wang, A.; Wang, Z.; Yang, N.; Zhao, Z.; Lai, C.S.; Lai, L.L. Prioritized sum-tree experience replay TD3 DRL-based online energy management of a residential microgrid. *Appl. Energy* **2024**, *368*, 123471. [\[CrossRef\]](#)
47. Zhang, H.; Liu, F.; Ma, B.; Zhang, S.; Peng, Y.; Gu, R.; Miao, J. Energy-efficient power scheduling policy for cloud-assisted distributed PV system: A TD3 approach. *Appl. Sci.* **2023**, *13*, 11611. [\[CrossRef\]](#)
48. Zhou, J.; Xue, S.; Xue, Y.; Liao, Y.; Liu, J.; Zhao, W. A novel energy management strategy of hybrid electric vehicle via an improved TD3 deep reinforcement learning. *Energy* **2021**, *224*, 120118. [\[CrossRef\]](#)
49. Pasha, S.A.; Vali Mohamad, N.M. Energy efficient PSO integrated TD3 reinforcement learning with nonlinear energy harvesting in UAV assisted D2D networks. *Int. J. Netw. Distrib. Comput.* **2026**, *14*, 12. [\[CrossRef\]](#)
50. Wang, Q.; Wang, Y.; Chen, Z.; Soares, J. Multi-agent system consistency-based cooperative scheduling strategy of regional integrated energy system. *Energy* **2024**, *295*, 130904. [\[CrossRef\]](#)
51. Fallah, A.; Mokhtari, A.; Ozdaglar, A. Personalized federated learning: A meta-learning approach. *arXiv* **2020**, arXiv:2002.07948. [\[CrossRef\]](#)
52. Dinh, C.T.; Tran, N.H.; Nguyen, T.D. Personalized federated learning with Moreau envelopes. In *Advances in Neural Information Processing Systems (NeurIPS 2020)*; Curran Associates: Red Hook, NY, USA, 2020; pp. 1–12.
53. Pilloni, V.; Floris, A.; Meloni, A.; Atzori, L. Smart home energy management including renewable sources: A QoE-driven approach. *IEEE Trans. Smart Grid* **2018**, *9*, 2006–2018. [\[CrossRef\]](#)
54. Constantopoulos, P.; Schweppe, F.C.; Larson, R.C. Estia: A real-time consumer control scheme for space conditioning usage under spot electricity pricing. *Comput. Oper. Res.* **1991**, *18*, 751–765. [\[CrossRef\]](#)
55. Thatte, A.A.; Xie, L. Towards a unified operational value index of energy storage in smart grid environment. *IEEE Trans. Smart Grid* **2012**, *3*, 1418–1426. [\[CrossRef\]](#)
56. Sun, Y.; Zhang, Y.; Guo, D.; Zhang, X.; Lai, Y.; Luo, D. Intelligent distributed temperature and humidity control mechanism for uniformity and precision in the indoor environment. *IEEE Internet Things J.* **2022**, *9*, 19101–19115. [\[CrossRef\]](#)
57. Zhang, Z.; Chong, A.; Pan, Y.; Zhang, C.; Lam, K.P. Whole building energy model for HVAC optimal control: A practical framework based on deep reinforcement learning. *Energy Build.* **2019**, *199*, 472–490. [\[CrossRef\]](#)
58. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [\[CrossRef\]](#)
59. Wang, D.; Hu, M. Deep deterministic policy gradient with compatible critic network. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 4332–4344. [\[CrossRef\]](#)

60. Tan, H. Reinforcement learning with deep deterministic policy gradient. In Proceedings of the 2021 International Conference Artificial Intelligence, Big Data and Algorithms (CAIBDA), Xi'an, China, 28–30 May 2021. [[CrossRef](#)]
61. Giannopoulos, A.; Spantideas, S.; Kapsalis, N.; Karkazis, P.; Trakadas, P. Deep reinforcement learning for energy-efficient multi-channel transmissions in 5G cognitive HetNets: Centralized, decentralized and transfer learning based solutions. *IEEE Access* **2021**, *9*, 129358–129374. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.