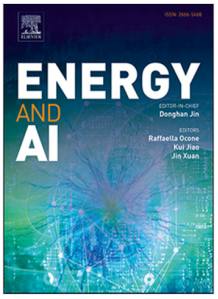
Journal Pre-proof



Federated reinforcement learning for sustainable and cost-efficient energy management

J. Sievers, P. Henrich, M. Beichter, R. Mikut, V. Hagenmeyer, T. Blank, F. Simon

PII: S2666-5468(25)00053-9
DOI: https://doi.org/10.1016/j.egyai.2025.100521
Reference: EGYAI 100521

To appear in: *Energy and AI*

Received date : 9 September 2024
Revised date : 16 April 2025
Accepted date : 27 April 2025

Please cite this article as: J. Sievers, P. Henrich, M. Beichter et al., Federated reinforcement learning for sustainable and cost-efficient energy management. *Energy and AI* (2025), doi: https://doi.org/10.1016/j.egyai.2025.100521.

Federated Reinforcement Learning for Sustainable and Cost-Efficient Energy Management

J. Sievers[a,*], P. Henrich[a], M. Beichter[b], R. Mikut[b], V. Hagenmeyer[b], T. Blank[a] and F. Simon[a]

[a] *Karlsruhe Institute of Technology (KIT), Institute for Data Processing and Electronics (IPE), Hermann-von-Helmholtz-Platz 1, Eggenstein-Leopoldshafen, 76344 , Germany*
[b] *Karlsruhe Institute of Technology (KIT), Institute for Automation and Applied Informatics (IAI), Hermann-von-Helmholtz-Platz 1, Eggenstein-Leopoldshafen, 76344, Germany*

ARTICLE INFO

Keywords:
Reinforcement Learning
Federated Learning
Energy Management
Smart Grid

Abstract

Integrating renewable energy sources into the electricity grid introduces volatility and complexity, requiring advanced energy management systems. By optimizing the charging and discharging behavior of a building's battery system, reinforcement learning effectively provides flexibility, managing volatile energy demand, dynamic pricing, and photovoltaic output to maximize rewards. However, the effectiveness of reinforcement learning is often hindered by limited access to training data due to privacy concerns, unstable training processes, and challenges in generalizing to different household conditions. In this study, we propose a novel federated framework for reinforcement learning in energy management systems. By enabling local model training on private data and aggregating only model parameters on a global server, this approach not only preserves privacy but also improves model generalization and robustness under varying household conditions, while decreasing electricity costs and emissions per building. For a comprehensive benchmark, we compare standard reinforcement learning with our federated approach and include mixed integer programming and rule-based systems. Among the reinforcement learning methods, deep deterministic policy gradient performed best on the Ausgrid dataset, with federated learning reducing costs by 5.01 % and emissions by 4.60 %. Federated learning also improved zero-shot performance for unseen buildings, reducing costs by 5.11 % and emissions by 5.55 %. Thus, our findings highlight the potential of federated reinforcement learning to enhance energy management systems by balancing privacy, sustainability, and efficiency.

1. Introduction

As climate change drives the transition to renewable energy sources (RES), their intermittent nature frequently causes imbalances between energy supply and demand [1]. Cebulla et al. [2] indicate that to maintain grid stability with RES penetration exceeding 80 %, Europe and the United States may require an additional 0.2 TWh to 3 TWh of storage capacity. Furthermore, Europe alone spent 2.25 billion $ on congestion management in 2021 [3], highlighting the need for additional flexibility solutions.

In this context, residential battery energy storage systems (BESS) are essential to maintain grid stability, enhance the integration of RES, or prevent grid expansion [4]. When combined with dynamic electricity pricing and smart meters, advanced BESS scheduling algorithms can predict energy demand peaks and optimize BESS usage based on future electricity prices to reduce electricity costs and emissions [5].

As shown in Figure 1, we propose a novel federated reinforcement learning (FRL) approach to optimize the charging and discharging operations of BESSs across geographically distributed buildings. In contrast to conventional reinforcement learning (RL) approaches relying on isolated building data, our federated method incorporates a global aggregation phase, enabling the model to learn from diverse building conditions without direct data exchange. Our federated approach reduces overall energy costs, lowers emissions, and enhances model robustness, while also ensuring improved generalization to previously unseen buildings. By

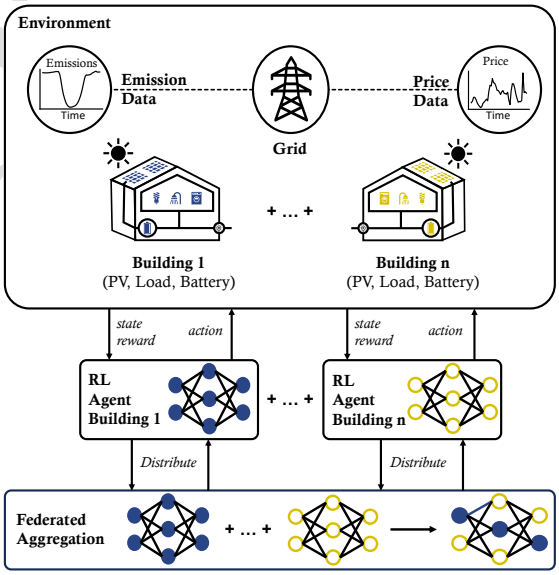


Figure 1: Federated Reinforcement Learning Architecture for Residential Battery Scheduling to Reduce Costs and Emissions

This work was funded by the German Research Foundation (DFG) as part of the Research Training Group 2153: "Energy Status Data - Informatics Methods for its Collection, Analysis, and Exploitation".

*Corresponding author

Nomenclature

Energy Management

β	Emission weighting factor
$\mathbb{P}_{bess}$	Electrical power of battery energy storage system
$\mathbb{P}_{grid}$	Electrical power from the grid
$\mathbb{P}_{load}$	Electrical power demand of the building (load)
$\mathbb{P}_{net}$	Net electrical power (prosumption)
$\mathbb{P}_{pv}$	Electrical power from photovoltaic generation
c	Operational electricity cost
c_{pen}	Penalty cost of the battery energy storage system
e	CO_2 equivalent emissions
p_{dyn}	Dynamic electricity price
p_{feedin}	Grid feed-in tariff
SoE	State of energy

Federated Reinforcement Learning

δ	Clipping parameter for weight updates
ϵ	Random noise term
γ	Discount factor for future rewards
$\mathbb{E}^{\pi}$	Expected value given policy π
$\mathcal{A}$	Action space, the set of all possible actions
$\mathcal{H}$	Entropy term
$\mathcal{M}$	Markov Decision Process tuple
$\mathcal{N}$	Normal (Gaussian) distribution
$\mathcal{P}$	State transition probabilities
$\mathcal{R}$	Reward function
$\mathcal{S}$	State space, the set of all possible states
μ	Actor network
ϕ	Parameters of critic network
π	Policy mapping states to actions
ψ	Parameters of value network
σ	Standard deviation of noise
θ	Parameters of actor network
υ	Model performance metric
a	Action taken by the agent
b	Scaling parameter
d_{dtw}	Dynamic time warping distance
d_{euc}	Euclidean distance
g	Discounted cumulative reward
Q	Action-value function
Q_{ϕ}	Critic network
r	Reward received by the agent
s	State of the environment
V	State-value function
V_{ψ}	Value network
w	Neural network weights

Indices and Superscripts

$\hat{y}$	Estimated value of variable y
b	Index of alignment in dynamic time warping
i, j	Index of client, building, or model
k	Index of cluster
t	Discrete time step

maintaining training stability and privacy protection, our FRL method outperforms conventional solutions, establishing FRL as an effective approach for decentralized energy systems.

Traditional methods for BESS scheduling, such as rule-based systems [6], mathematical modeling [7], and model predictive control [8], rely on detailed technical knowledge, thus limiting automation and scalability. In contrast, RL offers a data-driven approach utilizing historical and real-time data to dynamically adapt to complex environments [9], leading to reductions in operational expenses and increased grid resilience [10].

By optimizing BESS charging and discharging cycles, RL can enhance photovoltaic (PV) self-consumption and benefit from dynamic electricity pricing, thereby reducing costs and emissions [11]. The effective implementation of RL for BESS scheduling necessitates high-resolution data, typically obtained through smart metering infrastructure [12]. However, relying only on local data may limit data availability and diversity. One approach is to aggregate data from multiple buildings on a central server for model training, although this strategy raises significant privacy concerns [13]. Here, research indicates that smart meter data can disclose sensitive details such as users' habits, locations [14], and absences [15], sometimes leading to resistance to smart meter installation.

To ensure data privacy while learning from other buildings data, federated learning (FL) has been proposed. In FL [16], models are trained on individual devices with private data, and only the model parameters are shared and aggregated on a central server, thus enhancing data privacy, reducing latency and improving bandwidth efficiency within the communication network [17].

1.1. Related Work

For a detailed understanding of the current research and existing challenges, we introduce related work for RL and FRL in BESS scheduling. Selected publications are summarized in Table 1.

RL-based BESS scheduling has been explored in various real-world applications, including residential buildings, energy communities, microgrids, and industrial facilities. Each of these domains presents unique challenges in terms of

Table 1
Concept Matrix for the Literature on Reinforcement Learning and Federated Reinforcement Learning for Battery Scheduling

Ref.	Year	Focus	RL	Single obj.	Multi obj.	FL	Cluster	Agg.	FRL vs. RL
[23]	2024	Cost saving (DQN, TD3, DDPG)	✓	✓					
[24]	2023	Cost saving (Model-based DDPG, DQN, Q-learning)	✓	✓					
[25]	2024	PV self-consumption (DDPG, TRPO, Actor-Critic)	✓	✓					
[26]	2023	Cost savings (Actor-Critic)	✓	✓					
[27]	2023	Cost savings and PV self-consumption (PPO, DQN, DDPG, TD3)	✓	✓	✓				
[28]	2024	Cost savings and Comfort (Q-Learning)	✓	✓	✓				
[18]	2024	Peak-load and self-sufficiency (A2C, PPO, TD3, and SAC)	✓	✓	✓				
[29]	2024	Cost saving and PV self-consumption (DQN, Rainbow, PPO)	✓	✓	✓				
[30]	2023	FRL for energy management (DQN)	✓	✓		✓			
[31]	2023	FRL for demand side management (A2C)	✓	✓		✓			
[32]	2023	FRL for reduced energy consumption (Actor-critic)	✓	✓		✓			
This paper	2024	FRL for Battery Scheduling (DDPG, TD3, SAC, PPO)	✓	✓	✓	✓	✓	✓	✓

Abbreviations: Deep Q-Network (DQN), Twin Delayed DDPG (TD3); Deep Deterministic Policy Gradient (DDPG); Advantage Actor Critic (A2C); Proximal Policy Optimization (PPO); Soft Actor-Critic (SAC); Trust Region Policy Optimization (TRPO)

control. In residential systems, RL necessitates the capacity to adapt to rapidly evolving load patterns [18] to reduce electricity costs, improve PV integration, or reduce emissions [19]. In energy communities, the presence of multiple interacting subsystems, such as PV, flexible loads, and BESS, expands the control space and introduces coordination requirements across entities [20]. Microgrids introduce additional complexity by requiring real-time balancing of supply and demand, especially during islanded operation when external grid support is unavailable [21]. In industrial contexts, RL is used for dynamic scheduling of large-scale equipment to reduce peak loads and integrate renewable sources, thereby improving both operational resilience and sustainability [22]. These application-specific characteristics highlight the need for customized RL formulations that address the operational constraints of each domain.

Considering the different RL algorithms applied in the literature, value-based, policy-based, and actor-critic methods exist for both model-free and model-based RL [33]. Value-based methods, such as State-action–reward state–action (SARSA) [34] and Deep Q-Network (DQN) [35], focus on estimating state or action values to guide decisions. However, they are rarely used for battery scheduling in buildings due to their limitations in high-dimensional spaces [36]. Sultana, Ma, Hu, and Wang utilize SARSA for power scheduling of a BESS connected to a PV plant and an electric vehicle (EV) charging station [37]. Similar, Liu, Tang, Matsui, Takanokura, Zhou, and Gao apply SARSA to minimize electricity cost in residential buildings [38].

Policy-based methods like REINFORCE [39], Proximal Policy Optimization (PPO) [40], and Trust Region Policy Optimization (TRPO) [41], learn the best policy, which is the probability of taking action a given a state s, making them effective in continuous action environments. While some publications exist using PPO [18] or TRPO [42] for battery scheduling, most publications consider actor-critic methods [43]. Kang et al. utilize PPO for residential BESS scheduling with PV systems, demonstrating superior performance in maximizing self-sufficiency and economic benefits under real-world uncertainties [18]. Bollenbacher and Rhein apply TRPO, to optimize Energy Hub configurations and control strategies in multi-carrier energy systems, minimizing costs while meeting electrical and heat demands [42].

Actor-critic methods combine aspects of both approaches, utilizing a policy network (the actor) to select actions and a value network (the critic) to estimate action values, enhancing stability and efficiency. Common actor-critic methods are Deep Deterministic Policy Gradient (DDPG) [44], Twin Delayed DDPG (TD3) [45] and Soft Actor-Critic (SAC) [46]. Within buildings and electrical grids, different objectives are considered, including cost minimization [47], PV self-consumption maximization [48], or frequency stabilization [49]. Other objectives include optimizing energy efficiency, operational and investment costs, reducing battery degradation, or maximizing user comfort [50]. Chen et al. utilize DQN, TD3, and DDPG to evaluate the economic impacts of battery capacity, PV output, and price volatility [23]. Similarly, Xu et al. use model-based DDPG, DQN, and Q-learning to enhance cost savings in residential energy systems [24], while Real et al. reduce costs by increasing PV self-consumption in residential systems with Double DQN, Dueling DQN, Rainbow, and PPO [29]. Cheng et al. present a RL algorithm for optimizing the scheduling of multi-BESS, aiming to reduce electricity costs for consumers through adaptive control strategies [26]. Addressing grid objectives beyond economic performance, Kang et al. demonstrate the reduction of peak loads and improved self-sufficiency through the use of Advantage Actor Critic (A2C), PPO, TD3, and SAC [18]. Meanwhile, Yan, Xu, Wang, and Feng apply DDPG to control BESS resources to provide frequency support in power grids, aiming to enhance system stability and efficiency through data-driven decision-making [51]. Focusing on PV self-consumption, Dou et al. assess the performance of TRPO,

DDPG, Twin Actor-Critic, and Asynchronous Advantage Actor-Critic in scheduling residential BESS with PV systems, finding TRPO to be particularly effective in handling uncertainties and improving self-sufficiency [25]. Minimizing energy costs and maximizing PV self-consumption, Xu et al. introduce a model-based RL strategy specifically for optimizing residential PV-BESS systems, demonstrating the effectiveness of TD3 [27]. Meanwhile, Felicetti et al. present a hybrid framework that combines Q-learning for BESS operation with integer programming for load scheduling, successfully optimizing both electricity bills and mitigating the discomfort induced by demand response programs [28].

Few publications also exist, applying FRL to energy management in buildings. Here, Lee et al. use a DQN for cost minimization [30], while Lee and Choi employ an A2C for demand side management [31], and Lee et al. utilize an A2C to reduce energy consumption [32].

Besides the different RL algorithms, recent research increasingly focuses on developing scalable control architectures that can be applied across multiple buildings, distributed energy resources, and flexible loads [52]. Furthermore, multi-objective optimization formulations are explored to simultaneously address cost, emissions, and user comfort, providing a more comprehensive representation of the complex dynamics of modern energy systems [53]. The increased adoption of edge and distributed computing further highlights the need for control algorithms that balance computational efficiency with privacy protection [54].

Despite recent advancements, several fundamental challenges persist in the field of RL-based BESS scheduling. One of the key challenges is the lack of effective transfer learning strategies, which would enable models to learn from data across multiple buildings and generalize to environments with varying characteristics. Most current approaches focus on individual buildings and struggle with the inherent variability in user behavior, building dynamics, and control objectives. This variability hinders the development of scalable and robust solutions that can maintain high performance despite limited data, while also ensuring stable convergence and meeting computational constraints. Additionally, privacy concerns pose significant barriers to centralized learning methods, highlighting the need for decentralized approaches that can preserve privacy while maintaining model effectiveness. Addressing these challenges is crucial for developing high-performing, adaptive, scalable, and deployable BESS control solutions in real-world applications.

1.2. Paper Contribution and Organization

As shown in Figure 1, we introduce a FRL framework for BESS scheduling across multiple buildings, addressing the challenges of effective transfer learning, privacy preservation, and the reduction of electricity costs and emissions under variable conditions.

Despite selected advances in FRL, no publications exist benchmarking FRL against locally trained models, Mixed Integer Programming (MIP) or rule-based systems. However, such comparative analyses are essential to validate whether the FL framework indeed offers performance improvements. Additionally, no literature exists on state-of-the-art RL algorithms in federated environments, nor have clustering techniques been applied to leverage data heterogeneity, and advanced aggregation mechanisms beyond simple averaging have not been explored in previous studies.

Therefore, further research is required to improve FRL performance and data privacy. To the best of our knowledge, we are the first to demonstrate that incorporating a FL architecture into the RL training process can effectively lower the electricity costs or emissions compared to locally trained models and thus enable transfer learning. Consequently, our main contributions are as follows:

- We introduce a novel FRL framework that seamlessly integrates multiple state-of-the-art RL algorithms (DDPG, SAC, TD3, PPO) into a federated architecture, enabling decentralized, privacy-preserving training, and comprehensively benchmark its capabilities against corresponding locally optimized RL approaches, traditional rule-based methods, and MIP-based models.
- We propose advanced federated aggregation techniques, incorporating differential privacy, gradient clipping, weighted updates, and clustering-based approaches to enhance robustness, convergence efficiency, and resilience to data heterogeneity.
- We present a detailed sensitivity analysis that quantifies the effects of prosumption and emission forecast accuracy, as well as forecast horizon, on decision quality, emphasizing the importance of reliable predictions. Further, we demonstrate the zero-shot learning capabilities of our FRL framework, demonstrating improved generalization to unseen conditions.

The remainder of the paper is organized as follows: Section 2 introduces our methodology, while Section 3 outlines our experimental setup. Building on this, Section 4 presents our results, Section 5 discusses our results and limitations, and Section 6 provides our conclusion and future work.

2. Methodology

In this section, we provide a concise overview of our methodology, including RL and FL to reduce electricity costs and emissions.

2.1. Reinforcement Learning

As shown in Figure 2, a RL problem involves an agent interacting with an environment. This interaction is characterized by the agent observing states s, performing actions a, and receiving rewards r based on the performance. The underlying mathematical structure of this interaction is formalized as a Markov Decision Process (MDP) [55], represented by the tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where S represents the state space, A denotes the action space, P defines the transition probabilities between states, R is the

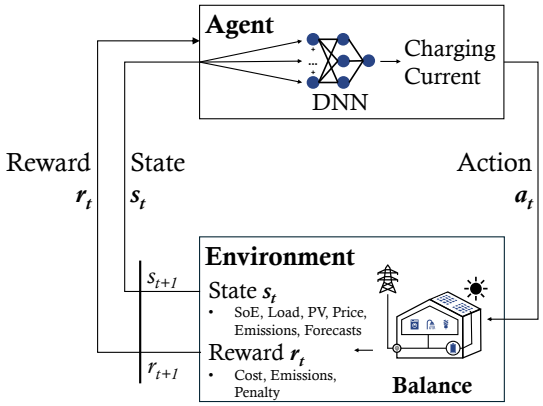


Figure 2: Reinforcement Learning Method for Residential Battery Scheduling

reward function, and γ is the discount factor that balances the importance of immediate versus future rewards ($0 \leq \gamma \leq 1$).

The agent aims to optimize the total discounted rewards from any time step t, formulated as Equation 1 [56].

$$g_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{n=0}^{\infty} \gamma^n r_{t+n+1} \quad (1)$$

To guide decision-making, the agent utilizes a policy π, which specifies a probability distribution over possible actions for a given state. This policy effectively represents the agent's strategy for selecting actions based on the observed state, balancing exploration and exploitation. The agent's objective is to select actions that maximize the expected cumulative future reward. The effectiveness of a state under policy π is measured by the state value function $V^\pi(s)$ [56], which is formally defined in Equation 2.

$$V^\pi(s) = \mathbb{E}^\pi \left[\sum_{n=0}^{\infty} \gamma^n r_{t+n+1} \mid s_t = s \right], \quad (2)$$

where s is the state at time t, r_{t+n+1} denotes the reward received at time $t+n+1$, and $\gamma \in [0, 1)$ is the discount factor that determines the importance of future rewards. The expectation $\mathbb{E}^\pi$ is taken over all possible trajectories that follow the policy π starting from state s.

In addition to evaluating states, it is often useful to evaluate state-action pairs. The action value function $Q^\pi(s, a)$ provides the expected return when taking action a in state s at time t, and thereafter following the policy π. It is formally defined in Equation 3

$$Q^\pi(s, a) = \mathbb{E}^\pi \left[\sum_{n=0}^{\infty} \gamma^n r_{t+n+1} \mid s_t = s, a_t = a \right], \quad (3)$$

where the expectation is again taken over all future trajectories generated by following π after taking action a in state s [56].

We use single-agent RL, where each building has an independent energy management system, instead of implementing a multi-agent RL framework, where each building requires inter-agent coordination. This approach reduces complexity, avoids communication overhead, and improves both scalability and stability by preventing coordination failures.

2.1.1. Environment

As energy infrastructures become increasingly decentralized, individual buildings equipped with PV systems and BESSs are transforming from passive consumers into prosumers, actively participating in the grid. We model the building's operating conditions as a simulated environment that incorporates PV production, volatile load profiles, storage constraints, and dynamic price signals. Within this environment, an RL agent learns to operate a BESS and iteratively refining its control policies through performance-driven feedback.

Based on the agent's decision to charge or discharge the BESS, the environment assesses the impact on the electricity grid, ensures adherence to physical constraints, and provides rewards to guide the agent's actions. As shown in Figure 3, at each time step t, the building's net power $\mathbb{P}_{net,t}$ is determined by balancing the electric load of the building $\mathbb{P}_{load,t}$, the power generated by the PV system $\mathbb{P}_{pv,t}$, and the charging or discharging power of the BESS $\mathbb{P}_{bess,t}$. The sign of $\mathbb{P}_{\text{bess},t}$ indicates the operating mode of the BESS, with positive values representing discharging and negative values charging. This power balance is formalized in Equation 4. A positive net power ($\mathbb{P}_{net,t} > 0$) implies that electricity must be purchased from the grid at the dynamic market price $p_{dyn,t}$, whereas a negative net power ($\mathbb{P}_{net,t} < 0$) indicates surplus electricity is fed into the grid and compensated at a fixed feed-in tariff $p_{feedin,t}$, as expressed in the cost function Equation 5. A positive value of c_t represents a cost, whereas a negative value indicates a profit:

$$\mathbb{P}_{net,t} = \mathbb{P}_{load,t} - \mathbb{P}_{pv,t} - \mathbb{P}_{bess,t} \quad (4)$$

$$c_t = \begin{cases} \mathbb{P}_{net,t} \cdot p_{dyn,t} & \text{if } \mathbb{P}_{net,t} > 0, \\ \mathbb{P}_{net,t} \cdot p_{feedin,t} & \text{if } \mathbb{P}_{net,t} < 0. \end{cases} \quad (5)$$

To ensure the BESS operates within its physical constraints, the agent's actions $a \in \mathcal{A}$ are limited to the maximum charging or discharging power of the BESS.

Based on the previously defined cost structure, the agent's objective is formalized through the reward function shown in Equation 6. This function integrates both economic and environmental considerations by weighting electricity costs and grid-related carbon emissions. Specifically, c_t denotes the electricity cost or revenue as defined in Equation 5, while e_t represents the greenhouse gas emissions (in kg CO_2-equivalent) associated with electricity consumption at time step t. The trade-off between economic and ecological objectives is controlled by a weighting parameter $\beta \in [0, 1]$, with $\beta = 0$ corresponding to a purely cost-driven optimization and $\beta = 1$ to an emission-minimization strategy.

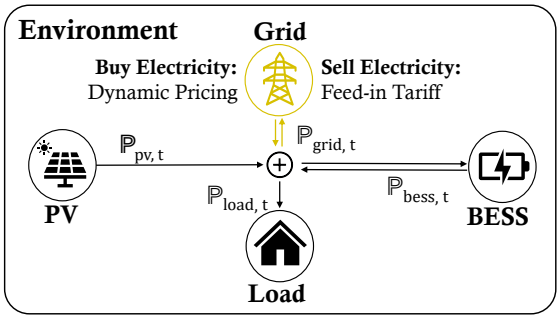


Figure 3: Power Flows within the Residential Battery Scheduling Environment

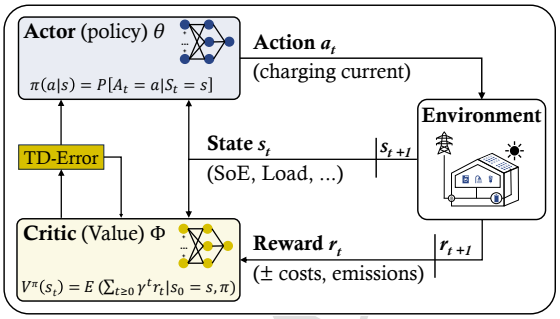


Figure 4: Actor-Critic Reinforcement Learning Architecture for the Use Case of Battery Scheduling

$$r_t = -c_t \cdot (1-\beta) - e_t \cdot \beta - c_{pen,t} \tag{6}$$

Furthermore, the reward function includes a penalty term $c_{pen,t}$ for exceeding operational limits, such as charging beyond rated capacity or operating outside the state-of-charge range (10 % to 90 %), which accelerates battery degradation. To facilitate convergence during training, all components of the reward function are normalized to ensure similar magnitudes.

The state s_t provided by the environment (Equation 7) is defined as an n-tuple comprising the current State of Energy (SoE) of the BESS (SoE_t), the net electrical load of the building ($\mathbb{P}_{net,t}$), the dynamic electricity price ($p_{dyn,t}$), and the associated emissions of the grid electricity (e_t). In addition to these current values, s_t includes forecast vectors of the dynamic price ($\hat{p}_{dyn}$), emissions ($\hat{e}$), and net load ($\hat{\mathbb{P}}_{net}$) with varying horizons to enable the agent to make more informed decisions based on anticipated future conditions.

$$s_t = (SoE_t, \mathbb{P}_{net,t}, \hat{\mathbb{P}}_{net}, p_{dyn,t}, \hat{p}_{dyn}, e_t, \hat{e}) \tag{7}$$

Accordingly, the reward associated with taking action a_t in state s_t is defined as the expected immediate reward, as shown in Equation 8:

$$R = \mathbb{E}[r_{t+1} \mid s_t = s, a_t = a] \tag{8}$$

Based on this reward formulation, the decision-making problem is modeled as a MDP $\mathcal{M} := (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the state transition function, $\mathcal{R}$ the reward function, and $\gamma \in [0, 1)$ the discount factor.

2.1.2. Algorithms

Effective scheduling of BESS demands robust RL algorithms capable of adapting to the dynamic and complex energy system environments. These algorithms must handle continuous action and observation spaces, reflecting real-world conditions where power dynamics are not discretely quantified. Moreover, they need to be compatible with FL settings to facilitate decentralized learning while ensuring data privacy. Based on these criteria and our literature review (Section 1), we select four state-of-the-art RL algorithms: DDPG, SAC, TD3, and, PPO. By employing all four algorithms, we not only enable a comprehensive comparison of their respective performance metrics but also demonstrate their seamless integration into our FL framework.

The DDPG agent [44] is an actor-critic method designed to operate in continuous action spaces. As shown in Figure 4, the architecture comprises two primary components: the actor and critic networks. The actor network maps the observed state space to a specific action, effectively functioning as a deterministic policy function. It takes the current state s as input and outputs an action $a = \mu(s \mid \theta)$, where θ are the parameters of the actor network μ. The critic network evaluates the action taken by the actor by computing the Q-value $Q(s, a \mid \phi)$, representing the expected cumulative reward for taking that action in the given state, where ϕ are the critic network parameters. Here, the Temporal Difference (TD) error, defined as the difference between the predicted Q-value and the target Q-value, is used to update the critic network's parameters, thereby refining the accuracy of the Q-value estimates and improving the policy. This evaluation helps the actor network to refine its policy. The agent interacts with the environment by selecting actions based on the current policy and receiving feedback from the environment in the form of rewards, which are used to update the networks according to the gradients of the policy and value functions [57].

The TD3 agent [45] builds on the DDPG framework, incorporating additional strategies to enhance stability and performance. Similar to the DDPG, the actor network in TD3 determines the action based on the observed state s, outputting $a = \mu(s \mid \theta)$. TD3 uses two critic networks to estimate Q-values $Q_1(s, a \mid \phi_1)$ and $Q_2(s, a \mid \phi_2)$, mitigating the issue of overestimation bias by taking the minimum value from the two critics $Q_{min}(s, a) = \min(Q_1(s, a), Q_2(s, a))$. To further stabilize training, TD3 introduces noise $\epsilon \sim \mathcal{N}(0, \sigma)$ to the target action. Here, ϵ represents the noise term, which follows a normal distribution $\mathcal{N}$ with a mean of 0 and a standard deviation of σ. This process, known as target policy

smoothing, reduces the likelihood of the policy overfitting to narrow peaks in the Q-function [45].

The SAC agent [58] is designed to maximize the trade-off between exploration and exploitation by optimizing a stochastic policy. This agent includes an actor network that outputs a probability distribution over actions, rather than a single deterministic action, represented as $\pi(a \mid s)$. This stochastic policy enables the agent to explore a broader range of actions. The SAC architecture employs two critic networks to compute Q-values $Q_1(s, a \mid \phi_1)$ and $Q_2(s, a \mid \phi_2)$, providing more robust value estimates and reducing the risk of overestimation. These networks evaluate the quality of actions taken based on the policy. A key feature of SAC is the entropy term $\mathcal{H}(\pi(\cdot \mid s))$ in the reward function, controlled by a temperature parameter, which adjusts the balance between exploration and exploitation.

PPO [59] is a policy gradient method known for its simplicity and effectiveness in optimizing stochastic policies. Here, the actor network outputs a distribution over possible actions $\pi(a \mid s)$, balancing exploration and exploitation. The value network estimates the state value $\hat{s} = V(s \mid \psi)$, where ψ are the value network parameters. This value indicates the expected return from the current state, helping to reduce variance in policy gradient estimates and improving the stability of the learning process. PPO introduces a clipping mechanism to the policy update preventing drastic changes to the policy and ensuring more stable and reliable learning.

2.2. Training Architectures

Next, we present both local learning and FL to train our RL agents.

In traditional local learning approaches for residential BESS scheduling, each building independently trains its own RL agent using only its individual dataset. Although this methodology results in models specialized for each building, it restricts the agent's ability to recognize broader patterns, such as diverse occupant behaviors or seasonal demand variations observed across different buildings. To address this limitation, horizontal FL is employed, enabling multiple clients with similar data types to collaboratively train a shared RL model without exchanging raw data. Unlike vertical FL, where different features of the same dataset are maintained by separate clients, horizontal FL allows clients to share the same feature space across different data samples. However, aggregating all clients to train a single global model requires extensive generalization to address critical differences in operational environments.

In FL, the server starts by initializing a global model for each cluster. A cluster refers to a subset of clients with similar data characteristics, allowing for more specialized model training. Each global model comprises various neural networks used by the RL agent, including but not limited to the value and policy network, depending on the specific RL algorithm employed. Subsequently, the global model is distributed to the clients within each cluster. Each client i then trains these networks locally using its specific dataset

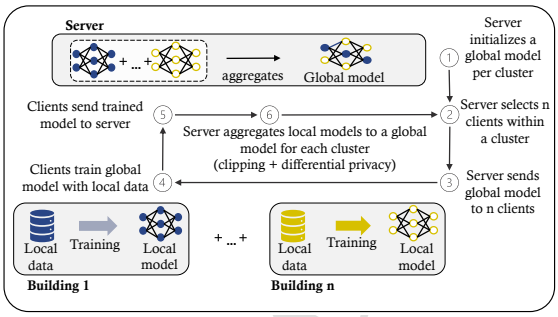


Figure 5: Federated Learning Architecture for Reinforcement Learning based Energy Management [62]

D_i. After local training, clients send their updated model parameters back to the server. The server then aggregates these parameters using a predefined method, enhancing the RL agent's networks by integrating diverse model parameters from similar clients, leading to a more robust and generalized model. The server repeats this iterative procedure over t training rounds, progressively enhancing the global model's performance [60]. An illustration of the FL architecture is shown in Figure 5 [61].

To update model parameters from selected clients, we consider four of the most promising approaches: Average Aggregation [16], Clipped Average Aggregation [63], Differential Privacy Average Aggregation [64], and Weighted Average Aggregation [65].

In Average Aggregation the mean of each client's model weights w_i are calculated, as defined in Equation 9 [66].

$$w = \frac{1}{N} \sum_{i=1}^{N} w_{\mathrm{i}} \tag{9}$$

Although Average Aggregation is simple to implement, it has several limitations. Considering FRL for BESS scheduling, this approach can be sensitive to outliers and malicious clients, potentially leading to a decrease in model performance. If a building with incorrect data or an adversarial intent submits biased model updates, these can disproportionately affect the aggregated model. Additionally, Average Aggregation may not effectively handle diverse data distributions, such as varying battery usage patterns across different facilities, potentially leading to suboptimal scheduling outcomes.

Clipped Average Aggregation (Equation 10) enhances robustness by constraining model weights w_i to a predefined range $\pm\delta$ before averaging, thus mitigating the influence of outliers and malicious updates [66].

$$w = \frac{1}{N} \sum_{i=1}^{N} clip(w_i, \delta) \tag{10}$$

$$w = \frac{1}{N} \sum_{i=1}^{N} (w_i + b \cdot \epsilon_i) \tag{11}$$

Table 2
Federated Learning Procedure with Clustered Aggregation

Input: Get data from each client i
Server clusters clients into k clusters based on similarity
for each cluster c^k with $k = 1, 2, \ldots, K$
  Initialize cluster-specific model weights, $w_i \leftarrow w_{rand}$
end for
for each cluster c^k with $k = 1, 2, \ldots, K$
  for each communication round $t = 1, 2, \ldots, T$
    for each client i in cluster C_k, $k = 1, 2, \ldots, K$
      Get new server model and update local model $w_i \leftarrow w_{rand}$
      Train local model w_i^k on local data
      Transmit updated w_i^k back to the server
    end for
    Update cluster-specific weights by aggregation
  end for
end for
Output: Cluster-specific weights: w_k, $k = 1, 2, \ldots, K$

In Differential Privacy Average Aggregation, each model update is modified by adding noise ϵ_i, scaled by a factor b, before computing the global average, as outlined in Equation 11. The additional noise term is used to obscure individual model weights and can improve model generalization, while potentially slowing convergence during training [66].

The Weighted Average Aggregation algorithm improves upon simple averaging by assigning a performance-based weight v_i to each model i, based on its performance on a client-specific evaluation dataset. The global model is then computed as a weighted sum, as shown in Equation 12:

$$w = \frac{\sum_{i=1}^{N} v_i w_i}{\sum_{i=1}^{N} v_i} \tag{12}$$

By ensuring that better-performing clients have a greater influence on the global model, the weighted average aggregation enhances overall model performance and reduces the impact of outliers and malicious clients [65].

For our FL architecture, we combine both clipping, differential privacy, and weighted average aggregation to achieve scalable and efficient BESS scheduling. This combination ensures robust energy management across distributed storage systems, enhances model accuracy by weighting diverse data contributions, and maintains high data privacy.

In FRL, clustering is essential to address challenges with non-independent and identically distributed (non-IID) data, where the client's data are heterogeneous and not uniformly distributed [67]. As different buildings have varying load and PV patterns, clustering similar buildings can improve model performance and convergence during training. Therefore, in clustered FRL each cluster of buildings trains their own global model, as shown in Table 2 [62].

Common clustering methods, such as K-Means clustering, typically rely on Euclidean distance to assess the similarity between data points. However, this metric is inadequate for clustering energy time series due to their inherent temporal variability, which involves significant differences in timing and speed. To overcome this limitation, we utilize K-Means clustering combined with Dynamic Time Warping (DTW). DTW offers a more precise similarity measure for time series data by flexibly aligning sequences, thereby accommodating non-linear temporal variations. This alignment enables the identification of similar patterns despite differences in timing or speed, ensuring that clusters accurately reflect underlying behavioral trends [68]. The Euclidean distance between two points x and y in an n-dimensional space is given by Equation 13 [69]:

$$d_{euc}(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{13}$$

In contrast, DTW calculates the distance between two time series $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ by finding the optimal alignment that minimizes the cumulative distance. The DTW distance is defined as Equation 14 [68]:

$$d_{dtw}(X, Y) = \sqrt{\min_{i_b, j_b} \left(\sum_{b=1}^{B} (x_{i_b} - y_{j_b})^2 \right)}, \tag{14}$$

where (i_b, j_b) are the indices of the points being aligned at the b-th step, and B is the total number of steps in the alignment path. The minimum is taken over all possible alignments of the two time series. DTW systematically explores all possible pairings of points between X and Y, selecting the alignment that minimizes the total cumulative distance. This process effectively handles time shifts and speed variations, allowing DTW to identify similarities between the time series despite misalignments.

3. Experimental Setup

Building on our methodology, we describe our experimental setup, including data analysis, RL hyperparameters, and training scenarios.

3.1. Datasets

We utilize the Ausgrid dataset [70], which contains smart meter readings from 2010 to 2013 from 300 residential buildings in Australia (Figure 6).

The dataset includes controllable load, general load and PV measurements in half-hourly resolution, indicated in kilowatt hours (kWh). Based on these time series, we extend the dataset by the total load (controllable and general load) and the prosumption (load minus PV output). For computational efficiency, our analysis focuses on a randomly selected subset of the first 30 households, referred to as clients. Additionally, the buildings 31 to 60 are utilized to assess zero-shot learning capabilities. Within the dataset, we define

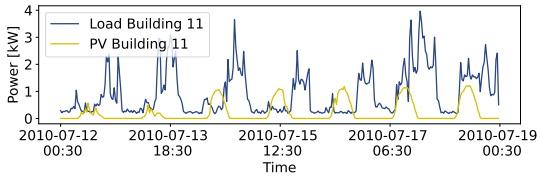


Figure 6: One Week of Load and PV Data of Building 11

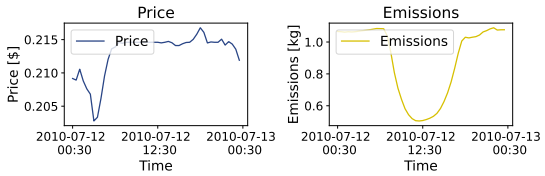


Figure 7: One Day of Price and Emission Data

outliers as (i) negative values (physical limitation for PV and load) or that (ii) deviate more than two standard deviations from the last value (inertia on building level). However, within the dataset no outliers or missing data points are present [62].

Further, we collect time and location-matching data for the electricity prices from the Australian Energy Market Operator [71]. We delete five negative price values, as the general curve of the price does not seem to support negative values. The missing five values are then replaced by linear interpolation. It is worth noting that the price dataset contains few price peaks that correlate with dates of weather extremes and are thus not corrected. Our final price dataset combines the stock market price and an offset of 0.1891 $ added to the net costs. This offset represents the taxes, levies, charges, and grid usage fees. To sell electricity to the grid, we apply a fixed feed-in tariff of 0.076 $ according to the current tariff system in New South Wales [72].

To obtain greenhouse gas emissions data for grid electricity, we utilize information from the OpenNEM platform [73] for the geographical zone (Sydney) aligning with the Ausgrid dataset. Since historical emission data for our specific time range is not available at half-hourly resolution, we utilize emission data from 2023 to model daily and seasonal variations. We scale this data to fit the annual emissions from the corresponding years in the Ausgrid dataset, ensuring accurate reflection of temporal and seasonal patterns. The price and emission patterns are illustrated in Figure 7.

To obtain our final datasets, we use the first year for training, the second year for evaluation, and the third year for testing.

3.2. Optimization algorithms

The energy management environment is initialized with default parameters to emulate a Tesla Powerwall, featuring a 13.5 kWh battery capacity and a 4.6 kW power limit for both charging and discharging. To ensure seamless integration of the Ausgrid dataset [70], we adjust our environment to align it with the same 48-slot partitioning of the day. Consequently, we define a continuous action space ranging from -4.6 kW to 4.6 kW, thus limiting the energy transfer to a maximum of 2.3 kWh per half an hour. The observation space consists of a 58-dimensional vector including the SoE, prosumption, grid emissions, electricity price, and forecasts. We incorporate forecasts for prosumption, price, and emissions over the next 9 h (18 values), using both perfect foresight and noisy forecasts. Perfect foresight facilitates the assessment of the theoretical performance of the RL algorithms by eliminating the impact of forecast inaccuracies, while noisy forecasts are used to evaluate the robustness and practical performance. The results of further forecast horizons (0-24 hours) and the integration of different energy forecasts (load, PV) are summarized in Appendix A.

Within our FRL architecture, we employ DDPG, SAC, TD3, and PPO algorithms, each optimized for our energy management system. A detailed summary of our algorithm parameters is provided in Appendix B. For all RL algorithms in our study, the following parameters are consistently applied: a discount factor of 0.99, network architectures comprising two fully connected layers with 400 and 300 units respectively, ReLU activation functions, and Huber loss for TD errors. Algorithms that incorporate target networks, specifically SAC, DDPG, and TD3, implement a soft update parameter τ of 0.05 with an update period of 5. These algorithms also employ an actor-critic architecture, with the actor learning rate set at 1×10^{-4} and the critic learning rate at 1×10^{-3}, both optimized using the Adam optimizer.

Further, the DDPG utilizes Ornstein-Uhlenbeck noise, characterized by a standard deviation of 0.2 and a damping factor of 0.15. The TD3 algorithm builds on the DDPG architecture by incorporating twin critics and policy smoothing techniques, which help mitigate overestimation bias in value function predictions. The SAC leverages a stochastic policy framework with an actor network and two critic networks. It optimizes the entropy temperature, with a target entropy of -1, to effectively balance exploration and exploitation. The PPO algorithm includes a value network and refines both policy and value functions over 20 training epochs per update cycle. Key parameters include a lambda value of 0.95 for generalized advantage estimation, entropy regularization of 0.1 to encourage policy exploration, and an unclipped importance ratio, ensuring that updates are controlled without excessive constraint on the policy changes.

To comprehensively benchmark our RL algorithms, we include a conventional rule-based BESS scheduling approach and a near-optimal solution derived from a MIP model. The rule-based system implements a straightforward strategy where BESS charging and discharging are determined by predefined price and emission thresholds. The full methodology of the rule-based system is shown in Appendix C. Additionally, the MIP approach leverages all historical and future data to minimize costs or emissions while ensuring compliance with all constraints. Detailed

information of the MIP formulation is provided in Appendix D.

Together, the four RL algorithms and the two benchmark approaches provide a comprehensive evaluation framework, ensuring a detailed analysis of performance across different scenarios.

3.3. Training Architecture

Our training architecture integrates a customized energy management environment with FL to enhance data privacy and model performance. The environment simulates crucial parameters of a BESS, while the FL setup coordinates training across multiple agents.

All RL agents are trained using a batch size of 128 and a replay buffer capacity of 20000 samples to store experiences. Note, that batch sizes smaller 128 resulted in unstable performance. Initial exploration is facilitated by collecting 2000 random steps, followed by 30 steps per iteration during training. We run the training for a total of 5000 iterations and evaluate the agent's performance at the last iteration (Appendix B).

For the FL setup, we distribute the training across 30 simulated buildings, grouped into 9 clusters. To determine the optimal cluster size, we evaluate various numbers of clusters ranging from 1 to 30, as shown in Appendix E. During our FL process, we repeat the local model training and global model aggregation for 3 rounds. Here, additional federated rounds did not yield further improvements. For our federated model aggregation, we employ Weighted Average Aggregation without clipping or noise, as this configuration provided the best results. The evaluation of different aggregation methods, noise levels (0-1), and clipping values (0-30) is presented in Appendix F.

All experiments are run in the TensorFlow 2 deep learning framework and all time related measurements are calculated on a simulation server using an Intel UHD Graphics 630 GPU with 16 GB memory attached to an Intel Core i9-9900K CPU at 4.6 GHz, with 8 kernels and 32 GB memory. The distributed training scenarios are all simulated on a single machine.

4. Results

In this section, we present our results demonstrating the effectiveness of our FRL approach compared to locally trained RL. To ensure a comprehensive evaluation, four distinct methods are benchmarked. The MIP model serves as an upper bound for the optimization problem, representing a near-optimal solution and illustrating the theoretical performance limit. However, the MIP approach is not applicable in real-time applications. Conversely, the traditional rule-based model establishes the lower bound, highlighting the performance gap between simple algorithms and more advanced approaches. We describe the rule-based system in Appendix C, while the MIP approach is detailed in Appendix D. Each RL algorithm (DDPG, PPO, SAC, and TD3) is evaluated under both local and federated training to demonstrate the effectiveness of our federated methodology.

Table 3
Average Annual Electricity Costs for Buildings 1 to 30. Here, a positive Diff. indicates an improvement of FL over LL.

	Local Learning		Federated Learning		
	Mean ($)	Std. ($)	Mean ($)	Std. ($)	Diff. (%)
MIP	1016.40	±0			
DDPG	1146.53	±28	1088.57	±0.6	**+5.06**
SAC	1214.27	±38	1199.92	±0.1	**+1.18**
TD3	1204.08	±47	1185.58	±0.3	**+1.54**
PPO	1207.55	±28	1193.50	±2	**+1.16**
RuleB	1728.33	±0			

We start by analyzing the potential cost savings achieved by the algorithms (Subsection 4.1), followed by an assessment of emission reductions (Subsection 4.2). To further validate the robustness of our RL algorithms, we implement a zero-shot optimization approach. Here, we test our RL algorithms on a set of 30 new and previously unseen buildings, without any preliminary retraining. By directly applying the pre-trained models to these new buildings, we assess their generalization capabilities and adaptability in real-world scenarios (Subsection 4.3). While these initial evaluations use perfect foresight for their forecasts, a subsequent performance analysis evaluates cost and emission savings using noisy predictions (Subsection 4.4). Next, the trade-off between costs and emissions is analyzed by systematically varying the prioritization in the objective function (Subsection 4.5). Finally, we briefly evaluate the training times for each RL algorithm (Subsection 4.6).

Each model is trained 3 times per scenario to consider statistical variations. If not stated otherwise, the metrics are averaged over all buildings, clusters, or training rounds and the results are achieved on the test dataset.

4.1. Analysis of Electricity Cost Savings

Optimizing electricity costs is essential for enhancing both economic efficiency and the adoption of RES. In this section, we present our findings on electricity cost savings, demonstrating that our best FRL algorithm (DDPG) can reduce annual electricity expenses per building from 1146.53 $ to 1088.57 $ (a 5.06 % decrease) compared to the locally trained model. For reference, the near-optimal costs calculated by the MIP amount to 1016.40 $, while the rule-based system results in 1728.33 $ of yearly expenses. Detailed results are provided in Table 3, where the reward function is optimized solely for electricity costs ($\beta = 0$).

It is important to note that the RL algorithms and the rule-based system can be deployed in real-time, whereas the MIP approach has access to the ground truth of all historical and future data patterns, thus offering a theoretical upper limit that is not achievable with real-time forecasting-based solutions.

The TD3 algorithm ranks second, with annual costs of 1204.08 $ for local learning and 1185.58 $ for FL. This reflects a 1.54 % improvement when using FL. However, the DDPG still outperforms the TD3 algorithm by 8.91 %.

Similarly, the PPO algorithm shows a 1.16 % cost reduction in the FL setting, decreasing expenses from 1207.55 $ to 1193.50 $. Despite this improvement, PPO remains 9.64 % less effective than DDPG. Among all RL algorithms evaluated, SAC incurs the highest electricity costs, with 1214.27 $ under local learning and 1199.92 $ under federated training.

When evaluating cost savings against the upper and lower bounds, the MIP approach achieves the lowest average electricity cost per building at 1016.40 $. This represents a 6.63 % improvement over federated DDPG, an 11.35 % improvement over local DDPG, and a 41.19 % reduction compared to the rule-based system. Additionally, federated DDPG demonstrates a 37.02 % cost reduction relative to the simple rule-based system, with costs ranging from 1088.57 $ to 1728.33 $.

While the rule-based and MIP methods yield deterministic results with zero standard deviation, the RL algorithms slightly vary in their performance caused by the random initializations. Here, the federated training of the DDPG agent reduces the standard deviation from 28 $ to 0.6 $, per building, enhancing the robustness and reliability of the training process.

In addition to the average cost savings across all buildings and evaluation rounds, Figure 8 presents the cost savings for each of the 30 buildings and training rounds individually. Here, each building has three values per algorithm, indicating the three evaluation rounds. The area marked as gain indicates the improvement of FL compared to local learning for a single evaluation round and building. As the greatly differing values are hard to compare, we connected the results of the individual buildings with lines (although they are independent) to improve clarity. Across the 30 buildings, the federated DDPG architecture achieves additional total savings of 1738.72 $, in electricity costs compared to the locally trained versions, which translates to an average of 57.96 $ per building and year. As indicated by the gain and loss areas in Figure 8, the FRL achieved the best results in 27 buildings, while in 3 buildings the local learning performed similar or better. Furthermore, when considering the best result from the three evaluation rounds for both local and FL approaches instead of the average, the FL architecture demonstrates improvements for all buildings, with savings ranging from a minimum of 0.51 $ to a maximum of 300.51 $.

4.2. Analysis of Emission Savings

Addressing the critical challenge of reducing greenhouse gas emissions is essential to mitigating climate change and enhancing sustainability. Next, we present our results for emission savings, highlighting that our federated DDPG can reduce the average annual emissions per building from 3514.34 kg to 3352.55 kg (a 4.60 % decrease) compared to the locally trained version.

Detailed results are shown in Table 4, where the reward function is optimized solely for emissions ($\beta = 1$). Similar to Subsection 4.1, we compare local and federated trained RL algorithms to the rule-based and MIP solutions.

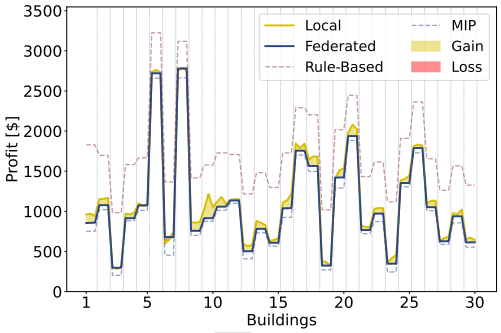


Figure 8: Electricity Cost Results for all 30 Buildings, where Gain indicates an improvement with Federated Learning. Note that the connecting lines are for orientation only, as each building is independent.

Both the SAC and TD3 algorithms achieve similar emission reductions. In the local learning setting, SAC results in slightly lower emissions (4745.93 kg) compared to TD3 (4749.27 kg). However, in the federated training setting, TD3 demonstrates lower emissions (4615.88 kg) than SAC (4638.73 kg). Notably, both algorithms show improvements in the federated setting, reducing emissions by 2.81 % for TD3 and 2.26 % for SAC.

While SAC shows the highest costs among the RL algorithms, the PPO algorithm results in the highest emissions savings, with 4861.73 kg in local learning and 4799.32 kg in FL. Furthermore, when comparing the FL results of different RL algorithms, DDPG outperforms SAC by 27.73 %, TD3 by 27.37 %, and PPO by 30.15 %.

The MIP approach provides the lower bound with an average of 2761.84 kg CO_2 per building. This is 17.62 % lower than the federated DDPG, 21.41 % lower than the local DDPG, and 53.78 % lower than the rule-based system. Moreover, the federated DDPG approach reduces emissions by 43.89 % compared to the rule-based system. Similar to the cost results, the FL of the DDPG decreases the standard deviation from 92 kg to 12 kg per building, indicating a more consistent performance across different buildings.

Similar to the cost savings, Figure 9 illustrates the emission reduction for each of the 30 buildings and training rounds individually. While each building has three values per algorithm indicating the performance within the three evaluation rounds, the connecting lines between the building results are for orientation only.

The FL architecture achieves additional emissions savings of 4853.41 kg compared to the locally trained version, averaging 161.78 kg per building and year. Furthermore, considering the best result from the three rounds for both local and FL approaches, the FL architecture demonstrates improvements for most buildings. While four buildings (1, 7, 23, 29) showed slight performance decreases, the remaining 26 buildings reduced emissions by up to 769.53 kg (Building 6).

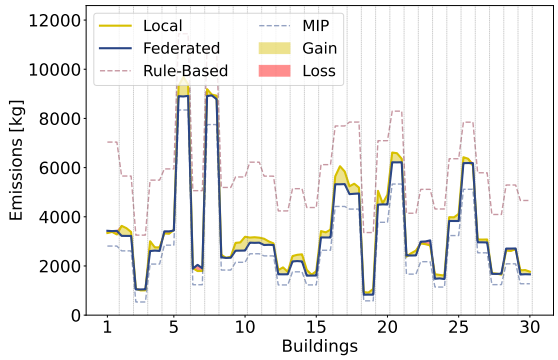


Figure 9: Emission Results for all 30 Buildings, where Gain indicates an improvement with Federated Learning. Note that the connecting lines are for orientation only, as each building is independent.

Table 4
Average Annual Emissions for Buildings 1 to 30. Here, a positive Diff. indicates an improvement of FL over LL.

	Local Learning		Federated Learning		
	Mean (kg)	Std. (kg)	Mean (kg)	Std. (kg)	Diff. (%)
MIP	2761.84	±0			
DDPG	3514.34	±92	3352.55	±12	**+4.60**
SAC	4745.93	±244	4638.73	±1	**+2.26**
TD3	4749.27	±104	4615.88	±7	**+2.81**
PPO	4861.73	±107	4799.32	±0	**+1.28**
RuleB	5975.37	±0			

4.3. Analysis of Zero-Shot Learning Capabilities

Scalability and efficiency in dynamic environments depend on a model's ability to perform effectively in novel scenarios. Zero-shot learning measures this ability by assessing the generalizability of models from learned experience to unseen environments.

This section evaluates the zero-shot learning capabilities of our FRL approach, demonstrating that the FRL approach can achieve cost reductions of up to 5.11 % and emissions savings of up to 5.55 % compared to locally trained models.

In zero-shot learning, the RL agents are evaluated on new buildings using unseen data, without any prior training or refitting. For simplicity, we only utilize the best-performing RL algorithm (DDPG) from Subsection 4.1 and Subsection 4.2. To obtain comparable DDPG agents, in local learning we train our models only on the data from building 1, while in FL we select the cluster containing building 1. Subsequently, we apply the local and federated DDPG agents to the buildings 31 to 60 without any retraining. The results for cost savings are summarized in Table 5, while the emissions reductions are detailed in Table 6.

The MIP provides a near-optimal solution of 963.88 $ outperforming the rule-based system by 42.86 %, which achieves costs of 1687.01 $ (Table 5).

The federated DDPG agent successfully reduces the electricity costs from 1080.33 $ to 1025.07 $ outperforming the local version by 5.11 % and the rule-based system by 39.24 %. However, the federated DDPG stays 5.96 % below the upper limit provided by the MIP. Comparing our zero-shot learning results to the cost savings from Subsection 4.1, the difference between the federated DDPG and the near-optimal MIP solution decreased from 6.63 % to 5.96 %, indicating a slight increase in performance.

Table 5
Average Annual Electricity Costs for Buildings 31 to 60 with Zero-Shot Learning. Here, a positive Diff. indicates an improvement of FL over LL.

	Local Learning		Federated Learning		
	Mean ($)	Std. ($)	Mean ($)	Std. ($)	Diff. (%)
MIP	963.88	±0			
DDPG	1080.33	±4	1025.07	±0	**+5.11**
RuleB	1687.01	±0			

Table 6
Average Annual Emissions for Buildings 31 to 60 with Zero-Shot Learning. Here, a positive Diff. indicates an improvement of FL over LL.

	Local Learning		Federated Learning		
	Mean ($)	Std. ($)	Mean ($)	Std. ($)	Diff. (%)
MIP	2520.83	±0			
DDPG	3301.30	±12	3118.23	±7	**+5.55**
RuleB	5806.26	±0			

Considering the emission savings in Table 6, the MIP provides a near-optimal solution of 2520.83 kg, outperforming the rule-based system with 5806.26 kg CO_2 emissions by 56.58 %. The federated DDPG agent reduces the emissions from 3301.30 kg to 3118.23 kg, outperforming the local version by 5.55 % and the rule-based system by 46.30 %. However, the federated DDPG stays 19.16 % below the upper limit provided by the MIP. Comparing our zero-shot learning results to the cost savings from Subsection 4.2, the difference between the federated DDPG and the near-optimal MIP solution increased from 17.62 % to 19.16 %, indicating a reduction in performance.

Besides the average zero-shot capabilities over all buildings, we show the results for each building and evaluation round for cost savings (Figure 10) and emission savings (Figure 11). The federated DDPG can improve the costs for all buildings compared to the local trained DDPG, saving a total of 1657.74 $, averaging 55.26 $ per building and year. The minimal improvement of 5.21 $ is realized for building 16, while building 23 can lower the electricity costs by 129.55 $ (Figure 10).

Figure 11 highlights, that the federated DDPG can improve the emissions for most buildings compared to the local trained DDPG, saving a total of 5492.03 kg CO_2 over all 30 buildings per year, which averages to 183.07 kg. While 4 buildings slightly decreased their performance, the remaining 26 buildings could decrease their emissions by up to 439.41 kg (Building 10).

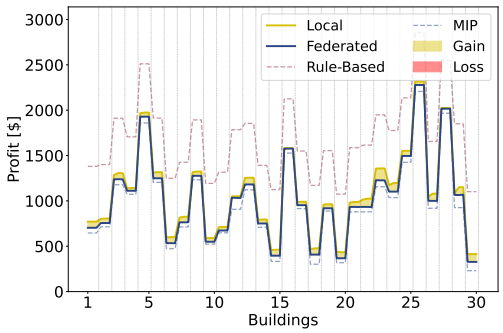


Figure 10: Cost Savings for all 30 Buildings within Zero-Shot Learning, where Gain indicates an improvement with Federated Learning. Note that the connecting lines are for orientation only, as each building is independent.

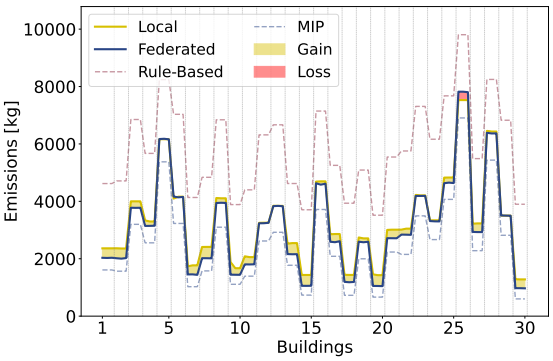


Figure 11: Emission Savings for all 30 Buildings within Zero-Shot Learning, where Gain indicates an improvement with Federated Learning. Note that the connecting lines are for orientation only, as each building is independent.

4.4. Analysis of Forecasting Accuracy on Emission and Cost Savings

While the previous result chapters used perfect foresight for their forecasts, in this section we evaluate the impact of decreasing forecast accuracy on the cost and emission savings. By including both perfect and imperfect forecasting setups in our evaluation, we first highlight the potential of RL without the influence of forecast errors. The results show decreasing performance with increasing noise scales for both local and federated RL, underlining the importance of accurate energy forecasts.

Within our RL architecture, we include prosumption, price and emission forecasts with a forecast horizon of 9 h (18 timeslots). As the dynamic price data for the next day is openly available to incentivize demand response, we only add noise to the prosumption and emission forecasts. Therefore, we create random noise following a normal distribution with a mean of zero and varying standard deviations (0 to 2). Here, the increase of the standard deviation relates to a decrease in forecasting accuracy. Each noise value represents a kilowatt deviation from the actual forecast, with higher standard deviations corresponding to larger forecast errors and reduced accuracy.

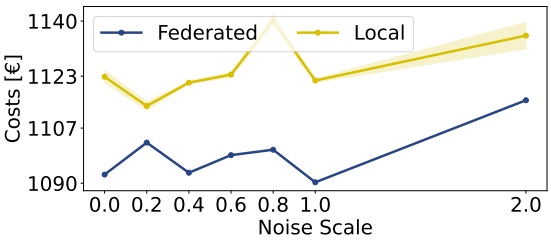


Figure 12: Impact of Forecasting Accuracy on the Annual Electricity Costs of Building 1 to 30

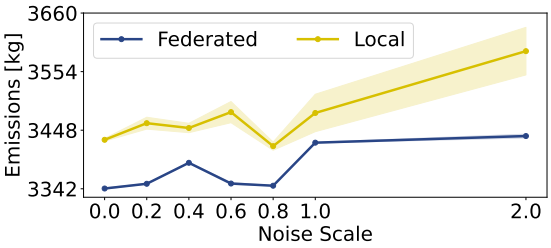


Figure 13: Impact of Forecasting Accuracy on the Annual Emissions of Building 1 to 30

For cost savings, we evaluate the effect of decreasing forecasting accuracy by only adding noise to the prosumption forecast, as the emissions do not affect the cost calculations ($\beta = 0$). The results are shown in Figure 12. To improve the visibility of the performance changes, we do not include the rule-based and MIP results.

As depicted in Figure 12, costs remain relatively similar yet volatile for noise scales between 0 and 1. As we repeat each experiment three times to account for statistical variations, the transparent shading around the line shows the standard deviation between the experiments. With an increasing noise scale up to 2, the performance starts decreasing more clearly. Here, a noise scale of 2 increases the costs by 2.05 % and 22.01 $ for the federated DDPG and by 1.12 % and 12.69 $ for the local DDPG compared to perfect foresight.

In parallel to the cost savings, we evaluate the effect of decreasing prosumption and emission forecast accuracy, only for emission reduction ($\beta = 1$). The findings in Figure 13 indicate a similar trend to the cost savings. The emissions remain relatively constant for noise scales between 0 and 1, but as noise scales increase from 1 to 2, the local DDPG shows a decline in performance, whereas the federated DDPG maintains more consistent emission levels. In detail, a noise scale of 2 increased the emissions by 2.76 % and 94.95 kg for the federated DDPG and by 6.92 % and 248.59 kg for the local DDPG compared to perfect foresight.

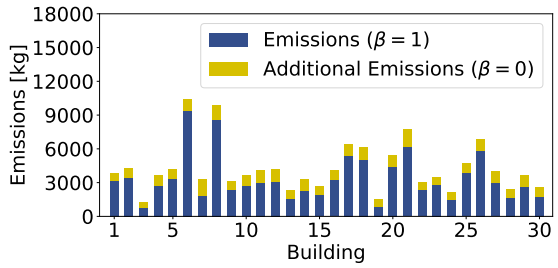


Figure 14: Emission Cost Trade-off for Building 1 to 30, where $\beta = 0$ optimizes only Costs and $\beta = 1$ considers only Emissions

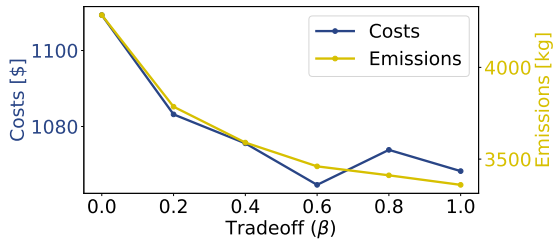


Figure 15: Emission Cost Trade-off for Buildings 1–30, where $\beta = 0$ optimizes only Costs and $\beta = 1$ considers only Emissions

4.5. Analysis of Emission and Cost Trade-off

While in the previous result sections, we only optimize costs or emissions, effectively setting β to 0 or 1, in this section we evaluate the trade-off between emissions and costs by systematically increasing β from 0 to 1. For simplicity, we only show the results utilizing the federated DDPG agent.

Figure 14 shows the average annual emissions per building. While the blue bars represent the emissions that occur if the target function only includes emissions ($\beta = 1$), the yellow bars represent the additional emissions if only costs are reduced ($\beta = 0$).

For each building the emissions increase when only optimizing for costs, which results in a difference of 27 763.86 kg of CO_2 per year over all buildings. Considering the results on building level, building 3 has the highest cost emission trade-off, as the emissions increase from 807.29 kg to 1260.97 kg (35.98 %). Conversely, building 21 has the lowest trade-off, where the emissions increase from 6175.19 kg to 7714.08 kg (19.45 %).

In Figure 15 we present the cost and emission trade-off on building level. The results indicate that varying β affects electricity costs from 1109.33 $ to 1068.26 $ and emissions from 4285.12 kg to 3359.66 kg. Emissions consistently decrease as their prioritization in the reward function increases. In contrast, electricity costs show an initial decrease as β rises, reaching a minimum at $\beta = 0.6$, after which the costs slightly increase. This trend is particularly interesting as it indicates that optimizing for both costs and emissions simultaneously, rather than focusing solely on minimizing costs, results in lower electricity costs up to a certain threshold. It is important to note, that the scales of emission and cost values vary, meaning that a $\beta = 0.5$ does not imply an equal weighting of both factors. While the reasons for this trend are discussed in Section 5, the findings suggest, that including emissions in the reward function (until $\beta = 0.6$) is always beneficial to reduce both costs and emissions.

4.6. Analysis of Total Training Time

In addition to the electricity and emission reduction of our RL models, we briefly evaluate the required training time only within local learning, as the training time for FL is proportional. Here, the total training time per building of an RL algorithm is shown in Appendix G. The PPO is the fastest algorithm with 10 minutes, followed by the SAC with 11 minutes, the TD3 with 12 minutes and the DDPG with 13 minutes.

5. Discussion

In this section, we discuss our results, including cost and emission savings, zero-shot optimization, performance under noisy predictions, and the cost-emission trade-off, highlighting key insights and limitations.

Based on the cost savings results in Subsection 4.1, our analysis shows that FRL significantly reduces electricity costs and variance in BESS scheduling. In local learning, models operate independently, which preserves data privacy but limits their learning capacity due to the isolation from other buildings' datasets. Conversely, the FL architecture enables models to utilize peer datasets within the same cluster while maintaining privacy. Our findings highlight that FRL enhances transfer learning, allowing models to leverage peer data to reduce electricity costs.

Among the evaluated RL algorithms, DDPG consistently outperforms the SAC, PPO, and TD3, highlighting the effectiveness of deterministic policies for BESS scheduling compared to the stochastic strategies employed by PPO and SAC. The ability of DDPG to effectively optimize continuous action spaces makes it ideal for energy systems.

The FRL framework also offers advantages in scalability and efficiency. Its decentralized nature enables seamless integration of additional buildings without substantial computational overhead. By clustering buildings based on similarity, new buildings can either form new clusters or join existing ones with high similarity. This scalability is particularly valuable for expanding energy management systems in large-scale deployments.

The analysis of the emission reductions in Subsection 4.2 shows similar results to our cost savings, where the FRL approach improves performance and reduces variance for emission savings. However, unlike cost minimization, the emission values are further from the optimum. This discrepancy may be attributed to the structure of the emission data, which remains constant during the night and has low values during the day. Consequently, the RL algorithm must learn to wait for longer time periods and only act during certain hours, which requires more sophisticated strategies to effectively optimize emissions.

In Subsection 4.3, our FRL architecture demonstrates superior zero-shot capabilities compared to local learning, indicating that FL enhances the robustness and generalizability of our RL algorithms. Such improvements are crucial for real-world applications where data patterns may change due to factors like new machine installations, changing residents, or seasonal variations. Models with better generalization require less retraining and are more resilient to data errors.

Further, pretrained RL algorithms can be efficiently deployed across similar buildings, requiring only minimal retraining to adapt to specific operating conditions, thereby reducing data requirements.

In Subsection 4.4 we analyze the impact of forecast noise on cost and emission savings. Our results show that smaller noise scales (ranging from 0 to 1) lead to only marginal changes in performance, while larger noise scales (1 to 2) cause more noticeable performance degradation. These results suggest that while RL agents are able to learn and incorporate future patterns into their decision making, their performance remains robust to moderate forecast inaccuracies. This behavior can likely be attributed to the relatively low sensitivity of optimal control outputs to small forecast variations and the capability of the RL agent to compensate for moderate noise levels. Consequently, the agent effectively adjusts its decisions, maintaining robust performance despite the presence of less accurate forecasts.

Considering the trade-off between cost and emissions in Subsection 4.5, our results demonstrate that incorporating emissions into the objective function can effectively reduce both emissions and costs simultaneously. However, this cost reduction is only observed up to a certain threshold of β, beyond which further prioritization of emissions results in rising costs. This initial decrease in costs is likely due to the complexities of the dynamic price data used in our study. The electricity price data obtained from the utility's official website exhibits significant variability due to changing market conditions, seasonal base price adjustments, and price spikes caused by extreme weather events. These fluctuations result in different base prices for different months, which complicates the ability of the RL algorithm to identify consistent patterns during training and evaluation. In the test dataset, the price data exhibits even greater variability compared to the training and evaluation datasets, making it increasingly difficult for the RL agent to develop a stable policy when focusing solely on cost minimization. In this context, incorporating emissions into the objective function serves as a stabilizing factor during the training process, providing a regularization effect that mitigates the risk of overfitting to volatile price data and encourages the exploration of more cost-effective solutions. Moreover, the cross-correlation between emissions and cost data could provide additional information. Importantly, while we could have pre-processed the price data to smooth out these fluctuations, we intentionally retained the realistic data patterns to better represent real-world conditions. In practice, RL algorithms must deal with similar levels of data uncertainty and variability, making this an essential aspect of our study. Based on our results, we recommend including emissions in the reward function in the presence of volatile price patterns to stabilize the training of the RL agent while simultaneously reducing emissions.

5.1. Limitations

This subsection addresses the limitations related to data selection, benchmarking, and federated aggregation, while highlighting key challenges for future research.

The electricity price data in our experiments shows significant outliers, likely due to extreme weather events and inconsistencies in base prices, with a noticeable spike in the third year. These factors make the RL training for cost savings more challenging, yet realistic. To enhance the comparability of our optimization results, we assume uniform feed-in tariffs for all households, even though, in reality, each household might select different tariffs. Due to the lack of high-resolution historical emission data, we rely on simulated data, highlighting the need for more openly available datasets to evaluate FRL performance across diverse scenarios.

The diversity of existing BESS scheduling approaches makes comprehensive benchmarking challenging. Therefore, we select only the most promising models from the literature. Further, due to computational constraints, we assume that using similar hyperparameters will yield comparable results, despite minor adjustments.

FRL integrates privacy preservation, cost and emission savings, and scalability, making it an effective method for managing the increasing complexity of energy systems. To further advance its capabilities, future work could investigate how FRL adapts to regulatory changes and evolving market conditions. These adaptations would ensure consistent optimization, even under novel conditions.

Enhancing the robustness of FRL against adversarial threats is another direction for future work. Techniques for mitigating the impact of manipulated data or compromised clients, such as secure aggregation and anomaly detection, ensure the integrity of the collaborative learning process.

In addition, reducing computational and communication overhead is critical for scalability. While FRL distributes computation across clients, further streamlining is essential to maintain high performance and enable deployment in resource-constrained environments. Addressing these challenges will establish FRL as a practical and reliable framework for managing complex and dynamic energy systems.

6. Conclusion

In this paper, we presented a novel federated reinforcement learning framework for battery scheduling, emphasizing enhanced data privacy, improved cost efficiency, and reduced emissions. By enabling decentralized model training and parameter aggregation, this approach addresses key challenges in energy management, including limited data sharing, variability in household conditions, and the need for generalization across diverse environments. Through a

comprehensive evaluation, our federated approach demonstrated significant improvements in both operational cost reductions and emission savings compared to standard reinforcement learning methods. Utilizing the Ausgrid dataset, we evaluated local and federated reinforcement learning algorithms, including Deep Deterministic Policy Gradient, Twin Delayed Deep Deterministic Policy Gradient, Soft Actor-Critic, and Proximal Policy Optimization. For comprehensive benchmarking, we compared our reinforcement learning approaches with a simple rule-based system and a near-optimal solution derived from mixed integer programming. Our results show that the federated reinforcement learning approach reduced costs by 5.01 % and emissions by 4.60 % compared to the standard reinforcement learning approach. Additionally, the federated approach improved zero-shot capabilities for unseen buildings by 5.11 % for cost and 5.55 % for emission savings. We also highlighted the critical role of accurate energy forecasts, showing that reduced forecast accuracy can lead to performance losses of up to 2.05 % for cost savings and 6.92 % for emission reduction. Furthermore, our analysis of the cost-emissions trade-off suggests that emissions should always be included to some extend in the objective function to improve performance. Future work could analyze the effect of different forecasting characteristics on cost and emission savings, as well as different clustering strategies to evaluate the ideal cluster size.

Acknowledgement

We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology. The presented work was funded by the German Research Foundation (DFG) as part of the Research Training Group 2153: “Energy Status Data — Informatics Methods for its Collection, Analysis, and Exploitation”. R. Mikut and V. Hagenmeyer were supported by the Helmholtz Association in the Program Energy System Design.

A. Forecasting in Reinforcement Learning

This section presents the results of integrating various forecasting methods into the RL environment for BESS scheduling. Here, the term "forecasts" denotes the use of true future values, also known as perfect foresight. This approach is used to establish a baseline, demonstrating the RL models' potential under ideal conditions. In later stages, we apply imperfect forecasts with controlled noise to systematically test the robustness and adaptability of the RL models in more practical, uncertain scenarios.

We compare the effectiveness of PV and load forecasts versus prosumption forecasts and assess the impact of including emission forecasts. We also evaluate different forecasting horizons (1 to 24 hours) to determine their influence on the RL agent's decision-making. When different horizons yielded similar results, we chose the shortest horizon due to the increasing difficulty of making accurate long-term forecasts. The results, summarized in Table 7, highlight the best parameters for each scenario in terms of cost savings, emission reductions. In our experiments, including prosumption forecasts generally results in higher cost savings and emission reductions compared to using PV and load predictions. Including emission forecasts further enhances performance, especially in terms of emission reductions. Although longer forecasting horizons initially improve cost savings and emission reductions, performance plateaus after 9 hours. Therefore, we selected a forecast horizon of 18 time steps (9 hours) as it offers the best balance between performance and forecast accuracy in reality.

Table 7
Performance Comparison of Different Forecasting Methods and Horizons in RL Environment

Scenario	Best Parameter
PV (horizon: 0-24 h)	horizon 9 h
PV and Load Forecast (horizon: 0-24 h)	horizon 9 h
Prosumption Forecast (horizon: 0-24 h)	horizon 9 h
PV + Emission Forecast (horizon: 0-24 h)	horizon 9 h
Prosumption + Emission Forecast (horizon: 0-24 h)	**horizon 9 h**

Table 8
Optimization Algorithm Parameters for RL Environment

Parameter Setup	**Parameter Range**	**Ideal Parameter**
Initial Collect Steps	$\{1000, 2000, 5000\}$	2000
Replay Buffer Capacity	$\{20000\}$	20000
Collect Steps Per Iteration	$\{1, 10, 20, 30, 50, 1000\}$	30
Num Iterations	$\{1000, 5000, 10000\}$	5000
Actor Learning Rate	$[1e-5, 1e-3]$	$1e-4$
Critic Learning Rate	$[1e-5, 1e-3]$	$1e-3$
Network Layers	$\{2, 3, 4, 5\}$	2
Network Units	$[10, 1000]$	$(400, 300)$
Gamma	$[0.9, 1]$	0.99
Loss	Huber loss	Huber loss
Optimizer	Adam	Adam
DDPG Actor Target Networks	$\{True, False\}$	$True$
DDPG Critic Target Networks	$\{True, False\}$	$True$
DDPG OU_Stddev Values	$[0.1, 1.0]$	0.2
DDPG OU_Damping Values	$[0.1, 1.0]$	0.15
DDPG Target Update Tau	$[1e-4, 1e-2]$	$5e-2$
DDPG Target Update Period	$\{1, 5, 10\}$	5
SAC actor loss weight	$[0.5, 2]$	1.2
SAC critic loss weight	$[0.5, 2]$	0.5
SAC target entropy	$[-5, 5]$	-1
SAC Alpha learning rate	$[1e-5, 1e-3]$	$1e-3$
SAC Target Networks	$\{True, False\}$	$True$
TD3 Target Networks	$\{True, False\}$	$True$
PPO lambda value	$[0.8, 1]$	0.95
PPO entropy regularization	$[0, 1]$	0.1
PPO number epochs	$[1, 100]$	20

B. Optimization Algorithm Parameters

In this section, we present a detailed overview of our RL parameters to ensure the reproducibility of our results. Table 8 summarizes the setup, parameter ranges, and the optimal values identified through extensive experimentation.

C. Rule-Based Energy Management

This section introduces a rule-based scheduling algorithm for BESS, serving as a benchmark to evaluate advanced optimization methods such as FRL and MIP. The rule-based approach specifically targets two distinct objectives: minimizing either electricity costs or emissions. At each time step t, the algorithm uses a state vector comprising the current SoE of the BESS (SoE_t), the net electrical load of the building ($\mathbb{P}_{net,t}$), the dynamic electricity price ($p_{dyn,t}$), and the associated emissions of the grid electricity (e_t). To enhance decision-making, the algorithm additionally employs forecasts of net load ($\hat{\mathbb{P}}_{net}$), dynamic price ($\hat{p}_{dyn}$), and emissions ($\hat{e}$) for the subsequent 18 time steps (9 hours). Charging and discharging decisions rely on percentile-based thresholds derived from forecasted electricity prices or emissions, depending on the selected optimization objective. After evaluating various thresholds, we achieved the best results using the 10th percentile as the lower threshold and the 90th percentile as the upper threshold. Charging occurs when the current price or emission factor is below the lower threshold, indicating favorable conditions. Conversely, discharging is triggered when these values exceed the upper threshold, signaling periods of high costs or emissions. Final costs and emissions calculations consider dynamic electricity prices and grid emission factors for imported energy, whereas exported surplus generation is compensated via a fixed feed-in tariff. Local PV generation is treated as emission-free. Due to its simplicity and transparency, this rule-based algorithm provides a valuable reference for quantifying the potential advantages of more sophisticated optimization techniques.

D. Mixed Integer Program

To benchmark our FRL approach comprehensively, we include MIP in our analysis. The MIP provides a near-optimal solution for the BESS scheduling problem, serving as an upper bound for our optimization objectives. Notably, the MIP utilizes all available historical and future data to determine optimal actions retrospectively, making real-time implementation not feasible and the comparison theoretical.

This section provides the MIP formulation, including parameters, variables, the objective function, and constraints.

D.1. Parameters

We define the following parameters for our problem.

B_{cap} - Battery capacity
B_{min} - Minimum battery capacity
$\mathbb{P}_{\text{charge, max}}$ - Maximum battery charging power
$\mathbb{P}_{\text{discharge, max}}$ - Maximum battery discharging power
soe_{init} - Initial state of energy in the battery
T - Number of time periods
β - Eco factor
c_t - Buying cost at time period t
s_t - Selling profit at time period t
$\mathbb{P}_{net,t}$ - Net load at time period t
e_t - CO_2 equivalent emissions at time period t

D.2. Variables

Next, we define the variables for our problem.

energy_sell_t - Energy sold at time t
energy_buy_t - Energy bought at time t
s_b_t - Buy/sell decision at time t
p_t - Penalty for buy/sell conflict at time t
ba_t - Battery action at time t
soe_t - State of Energy in battery at time t

D.3. Objective Function

Our objective function maximizes the total profit while considering emission costs and penalties. Note that profits here correspond to negative costs in our previous experiments.

$$\max \quad (1-\beta)\left(\sum_{t=1}^{T}(\text{energy\_sell}_t \cdot s_t - \text{energy\_buy}_t \cdot c_t)\right) - \beta\left(\sum_{t=1}^{T}(\text{energy\_buy}_t \cdot e_t)\right) - \sum_{t=1}^{T}\text{p}_t$$

D.4. Constraints

The following constraints bind the objective function.

State of Energy Constraints:

$$\text{soe}_t \leq B_{\text{cap}}, \quad \forall t$$
$$\text{soe}_t \geq B_{\text{min}}, \quad \forall t > 1$$
$$\text{soe}_1 = SoE_{\text{init}}$$
$$\text{soe}_t = \text{soe}_{t-1} - \text{ba}_{t-1}, \quad \forall t > 1$$

Battery Action Constraints:

$$\text{ba}_t \leq \mathbb{P}_{\text{discharge, max}}, \quad \forall t$$
$$\text{ba}_t \geq -\mathbb{P}_{\text{charge, max}}, \quad \forall t$$

Energy Balance Constraints:

$$\text{energy\_sell}_t = (\mathbb{P}_{net,t} - \text{ba}_t) \cdot \text{s\_b}_t \cdot (-1), \quad \forall t$$
$$\text{energy\_buy}_t = (\mathbb{P}_{net,t} - \text{ba}_t) \cdot (1 - \text{s\_b}_t), \quad \forall t$$

Penalty for Simultaneous Buy and Sell:

$$\text{p}_t = \text{s\_b}_t \cdot (1 - \text{s\_b}_t) \cdot 100, \quad \forall t$$

In summary, the MIP serves as an important benchmark, offering a near-optimal solution essential for assessing the

Table 9
Training time per reinforcement learning algorithm for 30 buildings

	DDPG	SAC	TD3	PPO
Time	105 min	114 min	117 min	40 min

effectiveness of both rule-based and advanced FRL approaches in BESS scheduling.

E. Federated Clustering Results

In the context of FRL, clustering can enhance overall performance by grouping similar buildings together. We conducted experiments with various cluster sizes, ranging from 1 to 30, to assess their impact on both cost and emission savings. For clustering, we used prosumption data from September, October, and November 2010 at a 4-hour resolution. This approach captures load and PV patterns as well as seasonal variations from summer to winter. We also tested clustering based solely on load or PV data, but these methods resulted in reduced performance compared to prosumption-based clustering. Based on our results for emission and cost savings, we selected a cluster size of 9. We employed k-means clustering with DTW to group the buildings. Our experiments indicated that cluster sizes ranging from 5 to 10 yielded similar performance, with the best results observed at a cluster size of 9.

F. Federated Aggregation Results

To improve the performance of FL, different aggregation mechanisms exist. Here we compared simple Average Aggregation to Weighted Average Aggregation and tested different parameters for clipping and noise. In theory clipping could stabilize the training process, while noise can work as a regularization technique to incentivize further exploration. For Simple and Weighted Average Aggregation, we tested clipping parameters $\delta \in [0, 30]$ and noise parameters $\epsilon \in [0, 1]$. The results indicate that Weighted Average Aggregation outperforms Simple Average Aggregation. Additionally, incorporating clipping and noise decrease the performance. We achieved the best results using Weighted Average Aggregation with a no clipping nor noise. These findings highlight the importance of selecting appropriate aggregation mechanisms and tuning parameters to enhance the effectiveness of FL for BESS scheduling.

G. Training Time of the Reinforcement Learning

In this section, we provide an overview of the training times per RL algorithm, as shown in Table 9.

H. Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author(s) used Grammarly, ChatGPT, and DeepL in order to improve the readability and language of the work. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

References

[1] F. Plaum, R. Ahmadiahangar, A. Rosin, J. Kilter, Aggregated demand-side energy flexibility: A comprehensive review on characterization, forecasting and market prospects, Energy Reports 8 (2022) 9344–9362. doi:10.1016/j.egyr.2022.07.038.

[2] F. Cebulla, J. Haas, J. Eichman, W. Nowak, P. Mancarella, How much electrical energy storage do we need? A synthesis for the U.S., Europe, and Germany, Journal of Cleaner Production 181 (2018) 449–459. doi:10.1016/j.jclepro.2018.01.144.

[3] F. Monforti-Ferrario, M. P. Blanco, The impact of power network congestion, its consequences and mitigation measures on air pollutants and greenhouse gases emissions. A case from Germany, Renewable and Sustainable Energy Reviews 150 (2021) 111501. doi:10.1016/j.rser.2021.111501.

[4] J.-F. Toubeau, J. Bottieau, Z. De Grève, F. Vallée, K. Bruninx, Data-Driven Scheduling of Energy Storage in Day-Ahead Energy and Reserve Markets With Probabilistic Guarantees on Real-Time Delivery, IEEE Transactions on Power Systems 36 (2021) 2815–2828. doi:10.1109/TPWRS.2020.3046710.

[5] J. Sievers, T. Blank, A Systematic Literature Review on Data-Driven Residential and Industrial Energy Management Systems, Energies 16 (2023). doi:10.3390/en16041688.

[6] J. de Hoog, K. Abdulla, R. R. Kolluri, P. Karki, Scheduling Fast Local Rule-Based Controllers for Optimal Operation of Energy Storage, in: Proceedings of the Ninth International Conference on Future Energy Systems, e-Energy '18, Association for Computing Machinery, New York, NY, USA, 2018, p. 168–172. doi:10.1145/3208903.3208917.

[7] V. A. Silva, A. R. Aoki, G. Lambert-Torres, Optimal Day-Ahead Scheduling of Microgrids with Battery Energy Storage System, Energies 13 (2020). doi:10.3390/en13195188.

[8] H. Zhang, S. Seal, D. Wu, F. Bouffard, B. Boulet, Building Energy Management With Reinforcement Learning and Model Predictive Control: A Survey, IEEE Access 10 (2022) 27853–27862. doi:10.1109/ACCESS.2022.3156581.

[9] R. Rocchetta, L. Bellani, M. Compare, E. Zio, E. Patelli, A reinforcement learning framework for optimal operation and maintenance of power grids, Applied Energy 241 (2019) 291–301. doi:10.1016/j.apenergy.2019.03.027.

[10] E. Kuznetsova, Y.-F. Li, C. Ruiz, E. Zio, G. Ault, K. Bell, Reinforcement learning for microgrid energy management, Energy 59 (2013) 133–146. doi:10.1016/j.energy.2013.05.060.

[11] B. V. Mbuwir, F. Ruelens, F. Spiessens, G. Deconinck, Battery Energy Management in a Microgrid Using Batch Reinforcement Learning, Energies 10 (2017) 1846. URL: https://api.semanticscholar.org/CorpusID:969221.

[12] E. Mocanu, D. C. Mocanu, P. H. Nguyen, A. Liotta, M. E. Webber, M. Gibescu, J. G. Slootweg, On-Line Building Energy Optimization Using Deep Reinforcement Learning, IEEE Transactions on Smart Grid 10 (2019) 3698–3708. doi:10.1109/TSG.2018.2834219.

[13] M. Savi, F. Olivadese, Short-Term Energy Consumption Forecasting at the Edge: A Federated Learning Approach, IEEE Access 9 (2021) 95949–95969. doi:10.1109/ACCESS.2021.3094089.

[14] A. Taïk, S. Cherkaoui, Electrical Load Forecasting Using Edge Computing and Federated Learning, in: ICC 2020 - 2020 IEEE

International Conference on Communications (ICC), 2020, pp. 1–6. doi:10.1109/ICC40277.2020.9148937.

[15] Y. Zhao, W. Xiao, L. Shuai, J. Luo, S. Yao, M. Zhang, A Differential Privacy-enhanced Federated Learning Method for Short-Term Household Load Forecasting in Smart Grid, in: 2021 7th International Conference on Computer and Communications (ICCC), 2021, pp. 1399–1404. doi:10.1109/ICCC54389.2021.9674514.

[16] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, 2023. doi:10.48550/arXiv.1602.05629.

[17] C. Fang, Y. Guo, N. Wang, A. Ju, Highly efficient federated learning with strong privacy preservation in cloud computing, Computers & Security 96 (2020) 101889. doi:10.1016/j.cose.2020.101889.

[18] H. Kang, S. Jung, H. Kim, J. Jeoung, T. Hong, Reinforcement learning-based optimal scheduling model of battery energy storage system at the building level, Renewable and Sustainable Energy Reviews 190 (2024). doi:10.1016/j.rser.2023.114054.

[19] Z. Rostamnezhad and L. Dessaint, Power Management in Smart Buildings Using Reinforcement Learning, 2023 IEEE Power Energy Society Innovative Smart Grid Technologies Conference (ISGT) (2023) 1–5. doi:10.1109/ISGT51731.2023.10066398.

[20] H. Kang and S. Jung and H. Kim and J. Hong and J.-H. Jeoung and T. Hong, Multi-objective sizing and real-time scheduling of battery energy storage in energy-sharing community based on reinforcement learning, Renewable and Sustainable Energy Reviews (2023). doi:10.1016/j.rser.2023.113655.

[21] M. H. Alabdullah and M. A. Abido, Microgrid energy management using deep Q-network reinforcement learning, Alexandria Engineering Journal (2022). doi:10.1016/j.aej.2022.02.042.

[22] C. Zhang and M. Juraschek and C. Herrmann, Deep reinforcement learning-based dynamic scheduling for resilient and sustainable manufacturing: A systematic review, Journal of Manufacturing Systems (2024). doi:10.1016/j.jmsy.2024.10.026.

[23] Q. Chen, Z. Kuang, X. Liu, T. Zhang, Application-oriented assessment of grid-connected pv-battery system with deep reinforcement learning in buildings considering electricity price dynamics, Applied Energy 364 (2024) 123163. doi:10.1016/j.apenergy.2024.123163.

[24] W. Xu, Y. Li, G. He, Y. Xu, W. Gao, Performance assessment and comparative analysis of photovoltaic-battery system scheduling in an existing zero-energy house based on reinforcement learning control, Energies 16 (2023). doi:10.3390/en16134844.

[25] Z. Dou, C. Zhang, J. Li, D. Li, M. Wang, L. Sun, Y. Wang, Innovative energy solutions: Evaluating reinforcement learning algorithms for battery storage optimization in residential settings, Process Safety and Environmental Protection 191 (2024) 2203–2221. doi:10.1016/j.psep.2024.09.123.

[26] G. Cheng, L. Dong, X. Yuan, C. Sun, Reinforcement learning-based scheduling of multi-battery energy storage system, Journal of Systems Engineering and Electronics 34 (2023) 117–128. doi:10.23919/JSEE.2023.000036.

[27] Y. Xu, W. Gao, Y. Li, F. Xiao, Operational optimization for the grid-connected residential photovoltaic-battery system using model-based reinforcement learning, Journal of Building Engineering 73 (2023) 106774. doi:10.1016/j.jobe.2023.106774.

[28] R. Felicetti, F. Ferracuti, S. Iarlori, A. Monteriù, Peak shaving and self-consumption maximization in home energy management systems: A combined integer programming and reinforcement learning approach, Computers and Electrical Engineering 117 (2024) 109283. doi:10.1016/j.compeleceng.2024.109283.

[29] A. C. Real, G. P. Luz, J. Sousa, M. Brito, S. Vieira, Optimization of a photovoltaic-battery system using deep reinforcement learning and load forecasting, Energy and AI 16 (2024) 100347. doi:10.1016/j.egyai.2024.100347.

[30] J.-H. Lee, J.-Y. Park, H.-S. Sim, H.-S. Lee, Multi-Residential Energy Scheduling Under Time-of-Use and Demand Charge Tariffs With Federated Reinforcement Learning, IEEE Transactions on Smart Grid 14 (2023) 4360–4372. doi:10.1109/TSG.2023.3251956.

[31] S. Lee, D.-H. Choi, Federated Reinforcement Learning for Energy Management of Multiple Smart Homes With Distributed Energy Resources, IEEE Transactions on Industrial Informatics 18 (2022) 488–497. doi:10.1109/TII.2020.3035451.

[32] S. Lee, L. Xie, D.-H. Choi, Privacy-Preserving Energy Management of a Shared Energy Storage System for Smart Buildings: A Federated Deep Reinforcement Learning Approach, Sensors 21 (2021). doi:10.3390/s21144898.

[33] A. Perera, P. Kamalaruban, Applications of reinforcement learning in energy systems, Renewable and Sustainable Energy Reviews 137 (2021) 110618. doi:10.1016/j.rser.2020.110618.

[34] G. Rummery, M. Niranjan, On-Line Q-Learning Using Connectionist Systems, Technical Report CUED/F-INFENG/TR 166 (1994).

[35] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with Deep Reinforcement Learning, 2013. arXiv:1312.5602.

[36] G. Chenxiao and Y. Wang and L. Xue and S. Nazarian and M. Pedram, Reinforcement learning-based control of residential energy storage systems for electric bill minimization, in: 2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), 2015, pp. 637–642. doi:10.1109/CCNC.2015.7158054.

[37] A. Sultana, X. Ma, R. Q. Hu, H. Wang, Power Scheduling and Cost Optimization of a Grid Integrated PV and BESS Fast Charging using SARSA Reinforcement Learning, in: 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall), 2024, pp. 1–6. doi:10.1109/VTC2024-Fall63153.2024.10756801.

[38] J. Liu, H. Tang, M. Matsui, M. Takanokura, L. Zhou, X. Gao, Optimal management of energy storage system based on reinforcement learning, in: Proceedings of the 33rd Chinese Control Conference, 2014, pp. 8216–8221. doi:10.1109/ChiCC.2014.6896376.

[39] R. J. Williams, Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning, Mach. Learn. 8 (1992) 229–256. doi:10.1007/BF00992696.

[40] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, ArXiv abs/1707.06347 (2017). URL: https://api.semanticscholar.org/CorpusID:28695052.

[41] J. Schulman, S. Levine, P. Abbeel, M. Jordan, P. Moritz, Trust Region Policy Optimization, in: F. Bach, D. Blei (Eds.), Proceedings of the 32nd International Conference on Machine Learning, volume 37 of *Proceedings of Machine Learning Research*, PMLR, Lille, France, 2015, pp. 1889–1897. URL: https://proceedings.mlr.press/v37/schulman15.html.

[42] J. Bollenbacher, B. Rhein, Optimal configuration and control strategy in a multi-carrier-energy system using reinforcement learning methods, in: 2017 International Energy and Sustainability Conference (IESC), 2017, pp. 1–6. doi:10.1109/IESC.2017.8167476.

[43] Y. Gao, Y. Matsunami, S. Miyata, Y. Akashi, Operational optimization for off-grid renewable building energy system using deep reinforcement learning, Applied Energy 325 (2022) 119783. doi:10.1016/j.apenergy.2022.119783.

[44] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, CoRR (2019). doi:10.48550/arXiv.1509.02971.

[45] S. Fujimoto, H. van Hoof, D. Meger, Addressing Function Approximation Error in Actor-Critic Methods, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 1587–1596. URL: https://proceedings.mlr.press/v80/fujimoto18a.html.

[46] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, 2018. arXiv:1801.01290.

[47] G. Muriithi, S. Chowdhury, Optimal Energy Management of a Grid-Tied Solar PV-Battery Microgrid: A Reinforcement Learning Approach, Energies 14 (2021). doi:10.3390/en14092700.

[48] B. V. Mbuwir, F. Ruelens, F. Spiessens, G. Deconinck, Battery Energy Management in a Microgrid Using Batch Reinforcement Learning, Energies 10 (2017). doi:10.3390/en10111846.

[49] L. Xi, L. Zhou, Y. Xu, X. Chen, A Multi-Step Unified Reinforcement Learning Method for Automatic Generation Control in Multi-Area Interconnected Power Grid, IEEE Transactions on Sustainable Energy 12 (2021) 1406–1415. doi:10.1109/TSTE.2020.3047137.

[50] R. Subramanya, S. A. Sierla, V. Vyatkin, Exploiting Battery Storages With Reinforcement Learning: A Review for Energy Professionals, IEEE Access 10 (2022) 54484–54506. doi:10.1109/ACCESS.2022.3176446.

[51] Z. Yan, Y. Xu, Y. Wang, X. Feng, Deep reinforcement learning-based optimal data-driven control of battery energy storage for power system frequency support, IET Generation, Transmission & Distribution 14 (2020) 6071–6078. doi:10.1049/iet-gtd.2020.0884.

[52] F. Charbonnier and T. Morstyn and M. Mcculloch, Scalable multi-agent reinforcement learning for distributed control of residential energy flexibility, ArXiv abs/2203.03417 (2022). doi:10.1016/j.apenergy.2022.118825.

[53] S. Touzani and A. K. Prakash and Z. Wang and S. Agarwal and M. Pritoni and M. Kiran and R. E. Brown and J. Granderson, Controlling distributed energy resources via deep reinforcement learning for load flexibility and energy efficiency, Applied Energy (2021). doi:10.1016/j.apenergy.2021.117733.

[54] T. Zheng and J. Wan and J. Zhang and C. Jiang, Deep Reinforcement Learning-Based Workload Scheduling for Edge Computing, Journal of Cloud Computing 11 (2022). doi:10.1186/s13677-021-00276-0.

[55] R. Bellman, A Markovian Decision Process, Journal of Mathematics and Mechanics 6 (1957) 679–684. doi:10.1007/BF00992696.

[56] F. Rezazadeh, N. Bartzoudis, A Federated DRL Approach for Smart Micro-Grid Energy Control with Distributed Energy Resources, 2022. arXiv:2211.03430.

[57] A. K. Shakya, G. Pillai, S. Chakrabarty, Reinforcement learning algorithms: A brief survey, Expert Systems with Applications 231 (2023) 120495. doi:10.1016/j.eswa.2023.120495.

[58] T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor, in: J. Dy, A. Krause (Eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of *Proceedings of Machine Learning Research*, PMLR, 2018, pp. 1861–1870. URL: https://proceedings.mlr.press/v80/haarnoja18b.html.

[59] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, CoRR abs/1707.06347 (2017). URL: http://arxiv.org/abs/1707.06347. arXiv:1707.06347.

[60] Y. L. Tun, K. Thar, C. M. Thwal, C. S. Hong, Federated Learning based Energy Demand Prediction with Clustered Aggregation, in: 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), 2021, pp. 164–167. doi:10.1109/BigComp51126.2021.00039.

[61] J. Sievers, T. Blank, Secure short-term load forecasting for smart grids with transformer-based federated learning, in: 2023 International Conference on Clean Electrical Power (ICCEP), 2023, pp. 229–236. doi:10.1109/ICCEP57914.2023.10247363.

[62] J. Sievers, T. Blank, F. Simon, Advancing Accuracy in Energy Forecasting using Mixture-of-Experts and Federated Learning, in: Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems, e-Energy '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 65–83. doi:10.1145/3632775.3661945.

[63] X. Zhang, X. Chen, M. Hong, Z. S. Wu, J. Yi, Understanding Clipping for Federated Learning: Convergence and Client-Level Differential Privacy, 2021. URL: https://arxiv.org/abs/2106.13673. arXiv:2106.13673.

[64] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, H. Vincent Poor, Federated Learning With Differential Privacy: Algorithms and Performance Analysis, IEEE Transactions on Information Forensics and Security 15 (2020) 3454–3469. doi:10.1109/TIFS.2020.2988575.

[65] D. Wang, N. Zhang, M. Tao, Clustered federated learning with weighted model aggregation for imbalanced data, China Communications 19 (2022) 41–56. doi:10.23919/JCC.2022.08.004.

[66] M. Moshawrab, M. Adda, A. Bouzouane, H. Ibrahim, A. Raad, Reviewing Federated Learning Aggregation Algorithms; Strategies, Contributions, Limitations and Future Perspectives, Electronics 12 (2023). doi:10.3390/electronics12102287.

[67] A. Moradzadeh, H. Moayyed, B. Mohammadi-Ivatloo, A. P. Aguiar, A. Anvari-Moghaddam, A Secure Federated Deep Learning-Based Approach for Heating Load Demand Forecasting in Building Environment, IEEE Access 10 (2022) 5037–5050. doi:10.1109/ACCESS.2021.3139529.

[68] D. J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series, in: Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, AAAIWS'94, AAAI Press, 1994, p. 359–370.

[69] S. Aghabozorgi, A. Seyed Shirkhorshidi, T. Ying Wah, Time-series clustering – A decade review, Information Systems 53 (2015) 16–38. doi:10.1016/j.is.2015.04.007.

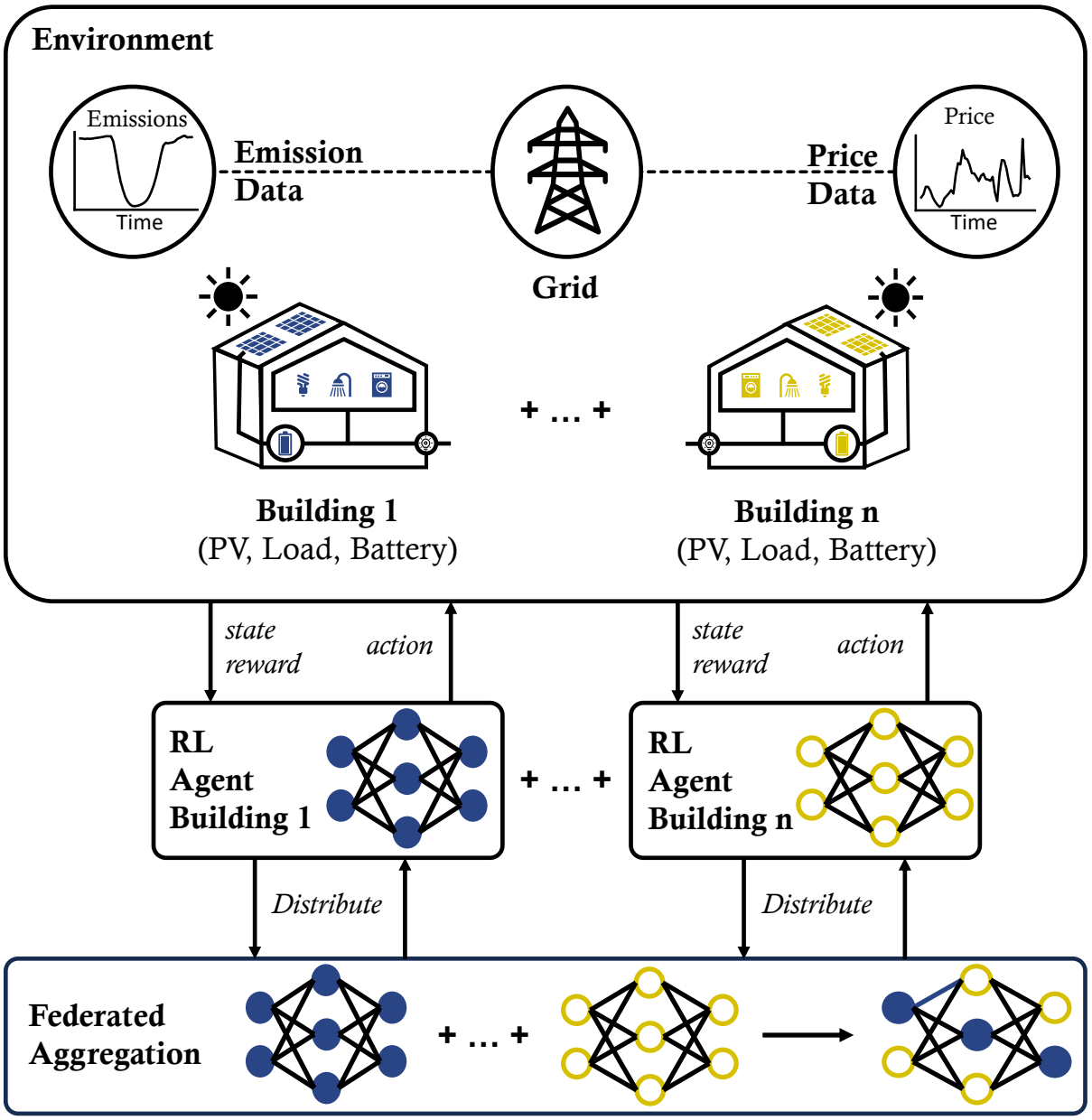
[70] Ausgrid, Ausgrid Electricity Consumption Data, 2018. URL: https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data, accessed: 2024-05-28.

[71] AEMO, Aggregated price and demand data, 2024. URL: https://aemo.com.au/energy-systems/electricity/national-electricity-market-nem/data-nem/aggregated-data, accessed: 2024-05-28.

[72] Australian Energy Council, Feed-in Tariffs State by State, 2024. URL: https://www.energycouncil.com.au/media/12974/feed-in-tariffs-state-by-state.pdf, accessed: 2024-08-28.

[73] OpenNEM, OpenNEM: An Open Platform for National Electricity Market Data, 2024. URL: https://opennem.org.au/, accessed: 2024-05-28.

- Novel federated framework for reinforcement learning in energy management.
- Federated reinforcement learning enhanced privacy, generalization, and robustness.
- Federated learning reduced costs by 5.01% and emissions by 4.60% in buildings.
- Improved zero-shot performance for unseen household conditions by over 5%.
- Highlights federated learning's potential for efficient energy management systems.


Environment
Emissions
Time
Emission Data
Grid
Price Data
Price
Time
+ … +
Building 1
(PV, Load, Battery)
Building n
(PV, Load, Battery)
state
reward
action
RL Agent Building 1
+ … +
RL Agent Building n
Distribute
Federated Aggregation
+ … +

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☒ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Jonas Sievers reports financial support was provided by Karlsruhe Institute of Technology. Jonas Sievers reports financial support was provided by German Research Foundation. R. Mikut and V. Hagenmeyer reports financial support was provided by Helmholtz Association in the Program Energy System Design. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.