IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 March 2026, accepted 15 March 2026, date of publication 20 March 2026, date of current version 25 March 2026.

Digital Object Identifier 10.1109/ACCESS.2026.3676334

RESEARCH ARTICLE

Cost Minimization in Energy Communities With Multi-Agent Deep Reinforcement Learning and Linear Programming

MATIC POKORN[1], ANDREJ ČAMPA[1,2], MIHA SMOLNIKAR[1,2], MIHAEL MOHORČIČ[1], (Senior Member, IEEE), AND JERNEJ HRIBAR[1], (Member, IEEE)

[1]Jožef Stefan Institute, 1000 Ljubljana, Slovenia
[2]ComSensus, 1233 Dob, Slovenia

Corresponding author: Jernej Hribar (jernej.hribar@ijs.si)

This work was supported in part by the HORIZON-MSCA-PF Project Timeliness of Information in Smart Grids Networks (TimeSmart) under Grant 101063721; in part by the Horizon Europe Research and Innovation Programme through the Data-driven Residential Energy Carrier-agnostic Demand Response Tools and Multi-value Services (DEDALUS) Project under Grant 101103998, through the Streaming flexibility to the power system (STREAM) Project under Grant 101075654, and through the Energy Activated Citizens and Data-Driven Energy-Secure Communities for a Consumer-Centric Energy System (ENPOWER) Project under Grant 101096354; and in part by Slovenian Research Agency under Grant P2-0016 and Grant MN-0009.

ABSTRACT With energy costs on the rise and with ever growing concern for environmental impact, energy providers and regulatory bodies have been pushing for dynamic energy prices as a means to encourage load shifting to reduce daily energy demand variance. Coupled with recent advancements in photovoltaics (PV) power generation and Battery Energy Storage System (BESS) technology, this has encouraged the development of energy communities with one of the goals to mitigate the effect of dynamic prices on homeowners' energy bills without sacrificing comfort, while at the same time utilizing aggregation of Distributed Energy Resources (DER) to contribute to grid flexibility. In this paper, we present Mathematical Optimization and Deep Reinforcement learning for Energy Cost minimization (MODREC), a decentralised Community Energy Management System (CEMS). MODREC leverages Multi-Agent Deep Reinforcement Learning (MADRL) coupled with Linear Programming (LP) to minimize cost in an energy community by intelligently charging and discharging household BESSs while assuming non-elastic consumer loads. MODREC follows an LP-guided training pipeline, where an optimal strategy computed with LP on historical data is employed to train a set of Deep Reinforcement Learning (DRL) agents, each assigned to a household in the community, that minimize a common cost function. Our main contribution lies in the system-level integration of LP-derived expert supervision with decentralized multi-agent control for community energy cost minimization under dynamic pricing. With MODREC, we manage to save up to 30% of energy costs compared to conventional approaches and efficiently shift energy load to off-peak hours.

INDEX TERMS Demand response, energy management system, linear programming, multi-agent deep reinforcement learning, energy community.

I. INTRODUCTION

In the past decade, the trend of implementing market-driven energy prices into power grids has steadily been on the rise worldwide [1]. Coupled with the widespread adoption of smart meters and increasing deployment of photovoltaics (PV) systems, battery energy storage system (BESS), Electric Vehicle (EV) chargers, and Heat Pumps (HP) this spurred an intense research in Demand Response (DR) mechanisms to support flexibility on the demand side. DR comprises a set of strategies used in energy management where consumers adjust their electricity usage in response to supply conditions, such as high demand or price changes. The goal is to balance electricity demand with supply, especially during peak periods or when the grid is under stress. Unpredictable environmental and energy market conditions

The associate editor coordinating the review of this manuscript and approving it for publication was Fazel Mohammadi.

and consumer behavior pose a considerable challenge to finding efficient DR strategies. However, benefits are present both on the demand side, where consumer energy costs can be substantially lowered with a price-responsive approach, and on the supply side, where DR on a larger scale aids at stabilizing the energy grid.

While distributed energy resources (DER) may have fragmented and individually insignificant role in flexibility, their aggregation and coordinated operation is recognised as one of the key enablers for flexible and resilient energy grids. This gave rise to the concept of energy communities where individual prosumers (i.e. producers and consumers) join forces to collectively pursue common goals such as self-consumption, self-sufficiency and coordinated investment, as well as lower bills and even generate revenue. To empower the effective operation of energy communities and the active engagement of their members, several community energy management system (CEMS) have emerged, however, they mostly rely on simple, static rule-based logic, day-ahead scheduling, or manual intervention. The most popular appears to be load/appliance scheduling, where CEMS schedules energy-intensive tasks with the aim to meet consumer's cost or comfort level requirements or in response to the request from the energy provider. While elastic load massively contributes to efficient DR, it has been reported that consumers are often annoyed by the fact that they are unable to use their electrical devices at the desired time [2], [3]. Many studies have put an emphasis on the reliance on predicting energy consumption and solar power generation, the latter heavily dependent on weather forecasts. Such methods apparently struggle with the dynamic nature of modern grids and energy markets, limiting their responsiveness and efficiency, leaving valuable flexibility from DER largely untapped.

In contrast, we developed a solution that does not require predictions that may be imprecise for instance due to local cloud cover influencing solar energy production. In our work, we consider a local energy community setting, where households can cooperate to lower their shared energy cost, and no solar power, consumption or price forecast is available. Individual households will be incentivized to join such a community through direct cost savings, the ability to benefit from price-responsive load shifting, and the non-intrusive nature of our approach, which preserves users' comfort. Fairness is ensured by proportional cost and benefit sharing based on each household's contribution (e.g., BESS capacity or flexibility).

To this end, in this paper, we propose an energy cost reduction solution for a virtual community of smart homes equipped with non-elastic loads, BESSs, PV, and dynamic energy pricing. We also assume that the community is able to sell energy in the event of PV generation surplus. For such a community, we propose a novel approach utilizing mathematical optimization and deep reinforcement learning for energy cost minimization (MODREC), essentially a MADRL-based CEMS. Trained on post-processed historical data and imitating optimal behaviour engineered with LP, MODREC lowers the community's energy costs and aids the energy provider at stabilising the energy grid.

In MODREC, we combine two well-known approaches, linear programming (LP) and Reinforcement learning (RL), for energy management, resulting in a unique solution. In this solution, we use LP to perform deterministic optimization with low computational burden and improve the training process by providing a reward signal that is more aligned with the problem [4], and RL for its adaptability in dynamic and complex systems along with the ability of real-time decision making. While many other works focus on leveraging one of the two approaches for home energy management, none, to the best of the authors' knowledge, have investigated a scenario, where both approaches would be combined to minimize energy costs of an energy community. We define the energy community as a set of households with the option of sharing energy resources, such as PV and BESSs, and where economic benefits are shared by the members of the community [5]. In the first step, we employ LP to build an ''expert'', i.e., a solution that optimizes the performance on historical data, and behaves optimally with respect to the desired cost function within a given environment while satisfying system constraints. In the second step, we use the ''expert'' to train the proposed solution, where each of the participating households' BESS is managed by a deep reinforcement learning (DRL) agent, which operates in real time and controls charging / discharging of BESSs in a manner that closely aligns with the behavior of the ''expert''. Such an approach is commonly used in fields like robotics and autonomous driving and is known as imitation learning [6].

The main contributions of this work are as follows:

1) We propose MODREC, a cooperative cost minimization CEMS control framework that combines LP with DRL. In MODREC, an LP-derived expert strategy is used to supervise training of decentralized agents (one per household) using a standard Deep-Q Network (DQN) approach, enabling imitation of near-optimal behavior while operating under environmental uncertainties;
2) We show that the proposed CEMS is lightweight in practice and readily configurable for different energy community settings (e.g., varying BESS capacities and pricing schemes), while requiring minimal resources and maintenance. The evaluation is based on data with a 15-minute resolution spanning two full years (January 2022–December 2023), including household consumption modeled via the open-source `pyloadprofilegenerator` [7], solar generation profiles obtained from Photovoltaic Geographical Information System (PVGIS) [8], and Slovenian market price data (energy and transmission charges). Under these conditions, the proposed framework provides superior performance, with savings of up to 29% versus a simple baseline policy on the considered case studies.

3) The MODREC-based CEMS is also non-intrusive, as it avoids appliance scheduling and provides unconstrained comfort to homeowners;
4) A key feature of MODREC is its resilience to information loss, maintaining high performance with minimal degradation even when certain system details are missing due to communication channel losses;
5) The proposed MODREC solution is price-responsive, and is consequently able to shift community load to off-peak hours of the day.

The remainder of this paper is organized as follows. Section II reviews related work and positions our contribution within the existing literature. Section III describes the considered local energy community setting, followed by the problem formulation in Section IV. Sections V and VI-A detail our proposed solution, while Section VI-B presents the real-time deployment of MODREC. Section VII outlines the evaluation methodology, and Section VIII discusses the experimental results. Finally, Section IX concludes the paper.

II. RELATED WORK

The proposed solution for community energy cost minimization through energy trading merges LP and DRL, two popular approaches used for DR. In the past, RL [9], [16], [17], [18], [19], [22], [23], [27], [32] and LP [11], [12], [20], [26] have proved useful in resolving a variety of problems in the home energy management domain. However, these approaches are rarely combined in such a way as proposed in our work.

Important aspects of home energy management, and in particular of demand-side management, are appliance scheduling [9], [10], [11], [12], [19], [20], [22], [26] and energy consumption forecasting in buildings [33]. Such problems are typically formulated as energy efficiency and consumption-side energy bill optimization problems, as also considered in our work. However, the problem we are solving is only loosely related to appliance scheduling as the focus of our solution is strictly to optimize the usage of BESS, i.e., deciding on when to charge or discharge, as opposed to turning appliances on and off.

LP has long been a widely used tool in various fields where optimal decision making is applied [34], [35]. In the home energy management domain, Mixed Integer Linear Programming (MILP) [11], [12], [20], [26] is a particularly popular approach for solving appliance scheduling tasks as home appliance on/off times generally require to be encoded as binary or integer variables. In contrast to these solutions, we only optimize BESS charging and discharging intensities, thus avoiding the need for integer variables and theoretically achieving lower time complexity (MILP is known to be NP-hard while LP problems can be solved in polynomial time) [36], [37].

Recently, RL has gained significant traction for smart grid control [38]. In [16], [17], and [18], authors propose a RL solution to control a single-home BESS without appliance scheduling, but they do not consider a community or Multi-Agent Reinforcement Learning (MARL). In [23], they consider a system of competing microgrids in a dynamic price environment, but they are not focusing on BESS or appliance scheduling. In [22], they use an RL algorithm for cost minimization with dynamic prices, but they do not deploy multiple agents and assume elastic load. In [19], they focus on air conditioner control and find an optimal solution with genetic algorithms. In [9], they use MARL for appliance scheduling in a dynamic pricing environment, but they do not address the multiple household problem or use a stationary BESS. In [27], they study a problem where a community shares an energy generation unit and a common BESS, using Fuzzy Q-learning to find an optimal energy trading strategy. They disregard appliance scheduling and use dynamic prices, however they do not use MARL as a means to solve their problem. In [39], they use both RL and LP, but their approach leverages both techniques in parallel where LP and RL are used to optimize their respective parts of the systems. Conversely, we use LP to find an optimal solution for the entire system, and then utilize it to train DRL agents to be able to perform in real time. In [29], they propose a MARL-based approach to reduce community energy costs, however they assume elastic load and use a more direct approach to training the RL agents, whereas we pre-engineer the training environment with LP to facilitate better convergence to the optimal policy.

Several studies have been published in recent years using more advanced DRL algorithms such as Deep Deterministic Policy Gradient (DDPG) [40] and Proximal Policy Optimization (PPO) [30]. While not used to tackle the same problem as here, these algorithms provide the advantage of accommodating continuous actions, which are also applicable to our study and may be investigated as part of our future work.

Lastly, many other techniques have been proposed for demand-side energy management such as LP [20], heuristics approaches (genetic algorithms [24] and particle swarm optimization [13]), model predictive control [14], [21], machine learning (including RL) [15], and game theory [25], [28], [41]. These approaches are typically used separately. However, we combine the strengths of LP and DRL to minimize the energy costs. While LP provides optimality, it is impractical in real time due to its inability to handle future uncertainties. In contrast, DRL addresses uncertainty but it can converge to local optima. Thus, we leverage mathematical optimization on historical data to create an ''expert'' policy, which is then used to train a DRL policy for real-time operation, imitating the behavior of the ''expert''. Some approaches of learning from an ''expert'' in energy management exist, however they mainly use combinations of MILP and Artificial Neural Network (ANN). In [31], they use this approach for appliance scheduling and in [42] for heat pump control.

Table 1 compares all relevant studies with our work. In summary, the proposed MODREC solution uniquely integrates MADRL with LP in a community environment featuring BESS and dynamic energy pricing. Unlike the

TABLE 1. Overview and comparison of existing work with the proposed MODREC solution.

Paper	Method	Environment	Use of BESS	Load type	Model type	Pricing
Xu et al. [9]	MARL	single household	yes	elastic	/	dynamic
Alfaverh et al. [10]	RL	single household	yes	elastic	/	dynamic
Silvente et al. [11]	MILP	single household	yes	elastic	/	dynamic
Yu et al. [12]	MILP	single household	yes	elastic	/	dynamic
Ma et al. [13]	PSO	single household	yes	elastic	/	dynamic
Chen et al. [14]	MPC	single household	yes	elastic	/	dynamic
Matallanas et al. [15]	Genetic alg.	single household	yes	elastic	/	dynamic
Pokorn et al. [16], [17]	RL	single household	yes	fixed	/	dynamic
Mbuwir et al. [18]	RL	single household	yes	fixed	/	dynamic
Hu et al. [19]	Genetic alg.	single household	no	elastic	/	dynamic
Yahia et al. [20]	MILP	single household	no	elastic	/	dynamic
Yu et al. [21]	MILP	single household	no	elastic	/	dynamic
Kim et al. [22]	RL	community	no	elastic	cooperative	dynamic
Wang et al. [23]	RL	community	no	fixed	competitive	dynamic
Neves et al. [24]	Genetic alg.	community	no	elastic	cooperative	fixed
Luan et al. [25]	Game theory	community	no	fixed	cooperative	dynamic
Barbato et al. [26]	ILP	community	yes	elastic	cooperative	dynamic
Zhou et al. [27]	RL	community	yes	fixed	cooperative	dynamic
Atzeni et al. [28]	Game theory	community	yes	fixed	cooperative	dynamic
Charbonnier et al. [29]	MARL	community	yes	elastic	cooperative	dynamic
Peirelinck et al. [30]	RL	single household	no	elastic	/	fixed
Dinh et al. [31]	MILP, ANN	single household	yes	fixed	/	dynamic
Proposed MODREC	MADRL, LP	**community**	**yes**	**fixed**	**cooperative**	**dynamic**

related works listed in Table 1, the proposed approach is the only one that combines these elements. This results in a robust and adaptable solution for managing community-level energy resources while minimizing overall community costs. The drawback of this approach, however, is that MODREC requires full observability of the community members' loads, PV outputs and BESS states, potentially raising privacy concerns and requiring participants to agree on sharing their anonymized household data with CEMS.

III. SYSTEM MODEL

In this section, we first describe the considered community power grid model, followed by the energy pricing scheme and a mathematically rigorous definition of the objective function and system constraints. We present the considered system graphically in Fig. 1 as a flow graph, with the relevant variables listed in Table 2.

A. ENERGY COMMUNITY

In this study we consider an energy community comprised of m households with or without PV and/or BESS units and investigate their energy generation and consumption over a given time interval divided into n equally-spaced discrete time steps. In time step j, each household's fixed consumption (c_{ij}), PV generation (g_{ij}) and BESS output / input (δ_{ij}) are measured. Additionally, the energy price for step j is reported and the cost for step j is computed from those values. An example 3-day snipped of these values is presented in Fig. 2. We assume that not every household in the community has a PV generation unit, and the number of households equipped with a PV system is $m_{PV} \leq m$. Similarly, we assume that only $m_{BESS} \leq m$ number of households have a BESS. Additionally, we assume the same BESS capacity to ensure fairness and comparability across households with BESS.

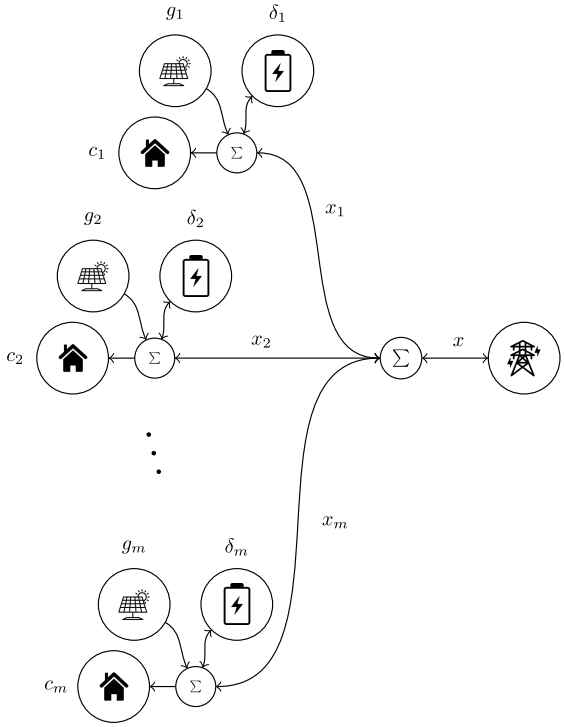


FIGURE 1. Illustration of the proposed system with *m* households. Note that not every household needs to have a BESS or PV.

In general, the number of households equipped with BESS (m_{BESS}) and with PV (m_{PV}) can be different and the set of households with a PV system and the set of households with a BESS can be disjoint. The i-th BESS has its own unique characteristics, namely maximal capacity B_i, maximal energy output / input δ_{max_i} and charging / discharging efficiency η_i. In our computations it will be assumed that, if a household i is not equipped with a PV system, then $g_{ij} = 0$ for all $j = 1, \ldots, n$. If it is not equipped with a BESS, it will

TABLE 2. System model variables with descriptions.

Notation	Description	Notation	Description
$m \in \mathbb{N}$	number of households	$i \in 1, \ldots, m$	index for enumerating households
$n \in \mathbb{N}$	number of time steps	$j \in 1, \ldots, n$	index for enumerating time steps
m_{BESS}	number of household with a BESS	m_{PV}	number of households with PV
c_{ij}	energy consumption	x_{ij}	energy output of household i at time step j
g_{ij}	energy generation	x_j	energy output of the community at time step j
B_i	maximal BESS capacity of household i	δ_{ij}	BESS external change
δ'^{*}_{ij}	optimal parameter for δ'_{ij}	y_{ij}	energy cost for household i at time step j
δ'_{ij}	BESS internal change	y_j	energy cost for the community at time step j
SoC_{ij}	BESS state of charge	y	cumulative energy cost of the community
η_i	charging / discharging efficiency (%)	x^{in}_j, x^{out}_j	energy consumption and generation of community
p^e_j	energy price (€/kWh)	$\delta^{in}_{ij}, \delta^{out}_{ij}$	BESS external input and output
p^d_j	transmission charge (€/kWh)	$\delta^{in*}_{ij}, \delta^{out*}_{ij}$	optimal parameters for BESS external input and output
SoC^*_{ij}	optimal parameter for SoC_{ij}		
J	batch size	λ	target network update rate
D	memory capacity	N_{eps}	number of episodes
ϵ	exploration rate	p_ϵ	ϵ-greedy reduction factor
l	number of dense layers	w	step width
d	dense layer size	k	sliding window size
a_{ij}	action taken	f_i	function mapping a_{ij} to δ_{ij}
R^*_{ij}	offline reward	R'_{ij}	online reward
$\mathbf{s}_j$	state at time step j		

be assumed that $B_i = 0$, $\delta_{max_i} = 0$ and $\eta_i = 0$ for all $j = 1, \ldots, n$.

The BESS state of charge (SoC) for a household i at time j is denoted as SoC_{ij} whereas δ'_{ij} is the actual energy being discharged from or charged into the BESS. Note that δ' and SoC are equivalent descriptions of the BESS state and their relationship is as follows:

$$\delta'_{ij} = SoC_{ij} - SoC_{i(j-1)}; \quad i \in \mathbb{N}_{\leq m}, j \in \mathbb{N}_{\leq n}, \tag{1}$$

$$SoC_{ij} = SoC_{i0} + \sum_{k=1}^{j} \delta'_{ik}; \quad i \in \mathbb{N}_{\leq m}, j \in \mathbb{N}_{\leq n}. \tag{2}$$

Additionally, we assume that all BESSs at the beginning of the investigated period are empty:

$$SoC_{i0} = 0; \quad i \in \mathbb{N}_{\leq m}. \tag{3}$$

For the remainder of this paper we will use δ' as the main BESS state descriptor. δ'_{ij} is the only controllable variable in the system and must conform to constraints posed by B_i and δ_{max_i}:

$$0 \leq \sum_{k=1}^{j} \delta'_{ik} \leq B_i; \quad i \in \mathbb{N}_{\leq m}, j \in \mathbb{N}_{\leq n}; \tag{4}$$

$$|\delta'_{ij}| \leq \delta_{max_i}; \quad i \in \mathbb{N}_{\leq m}, j \in \mathbb{N}_{\leq n}. \tag{5}$$

Considering that δ' is the change of BESS SoC within the BESS, due to the BESS charging / discharging efficiency η_i the energy input / output from the system's perspective will not be equal to δ'_{ij}. We define δ_{ij} to represent the energy input / output from the system's perspective and the relationship between δ'_{ij} and δ_{ij} is:

$$\delta'_{ij} = \begin{cases} \eta_i \delta_{ij}; & \delta_{ij} \leq 0, \\ \frac{1}{\eta_i} \delta_{ij}; & \delta_{ij} > 0. \end{cases} \tag{6}$$

Next, we determine the net energy exchange for household i, presented with Eq. (7). Note that positive x_{ij} will mean that more energy is being put out of the household while negative x_{ij} will signify that more energy is taken in the household, i.e., consumed and/or stored than generated:

$$x_{ij} = g_{ij} - c_{ij} - \delta_{ij}. \tag{7}$$

Finally, we can formulate energy exchange x of the entire community in a given time step j as the sum of energy exchange of all households as follows:

$$x_j = \sum_{i}^{m} x_{ij}. \tag{8}$$

This aggregate exchange serves as the foundation for determining the corresponding energy pricing model of the community.

B. ENERGY PRICING

The market-driven energy price in our model is composed of two components: the price p^e of energy consumption and the transmission charge of electricity p^d. We assume different pricing models for buying and selling energy. Since we consider a community of households, we must define a

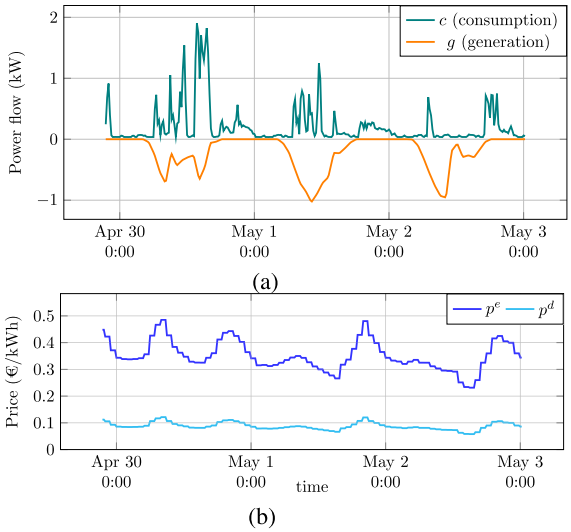


FIGURE 2. A 3-day snippet from the time series dataset for consumption, generation of one particular household equipped with a PV generation unit, along with prices.

pricing model for both the community as a whole and the individual households within it. We are constrained by the requirement that the sum of energy costs for all households at any given time step must equal the energy cost of the entire community.

The cost y_j for the community at time step j is determined as:

$$y_j = \begin{cases} -x_j(p_j^e + p_j^d); & x_j < 0 \\ -x_j p_j^e; & \text{otherwise.} \end{cases} \tag{9}$$

Similarly, cost for a specific household y_{ij} is calculated as follows:

$$y_{ij} = \begin{cases} -x_{ij}(p_j^e + p_j^d); & x_j < 0 \\ -x_{ij} p_j^e; & \text{otherwise.} \end{cases} \tag{10}$$

Note that the case for a specific household depends on the energy exchange of the entire system, not its own. This results in a scenario, where x_{ij} for a particular household i might be negative, but x_j might be positive. This means that household i may be consuming energy while paying a lower price p^e because the community is generating or selling energy thanks to output of other households, as opposed to a scenario where energy is not being shared and would have to be paid at the cost of the sum of energy and transmission prices, i.e., $p^e + p^d$.

If $x_j < 0$, the community as a whole has an energy shortage and it must take power from the energy grid in order to satisfy its energy needs. On the other hand, if the community has an energy surplus it will sell the excess energy to the grid. We assume that transmission charges apply only when buying the energy from the grid. This is due to the fact that energy produced by the community will be consumed elsewhere by a different consumer who will cover the transmission costs; secondly, it must be verified that community-side and household side cost equality holds, i.e., $\sum_{i=1}^{m} y_{ij} = y_j$. In our system this equality is trivial to verify, but it is worth mentioning that Eq. (10) could take a very different form depending on the community agreement for instance to reward or incentivize households with BESS and/or PV for sharing the energy, while still conforming to the cost equality constraint.

Finally, as we are interested in measuring the cumulative cost over the desired time interval, we define cumulative cost y as:

$$y = \sum_{j=1}^{n} y_j = \sum_{j=1}^{n} \sum_{i=1}^{m} y_{ij}. \tag{11}$$

With the system model defined, the next step is to derive the optimization problem.

IV. PROBLEM FORMULATION

Assuming that all parameters g_{ij}, c_{ij}, p_j^e and p_j^d are known for $i = 1, \ldots, m$ households and $j = 1, \ldots, n$ time steps, the objective is to determine the values of δ_{ij} that minimize y. Accordingly, the problem is formulated as the following constrained optimization model:

$$\begin{aligned} &\min_{\delta_{ij} \in \mathbb{R}^{m \times n}} \sum_{j=1}^{n} \sum_{i=1}^{m} y_{ij}; \\ &\text{subject to} \quad y_{ij} = \begin{cases} -x_{ij}(p_j^e + p_j^d); & x_j \le 0; \\ -x_{ij} p_j^e; & x_j > 0; \end{cases} \\ &\qquad x_{ij} = g_{ij} - c_{ij} - \delta_{ij}; \\ &\qquad \delta'_{ij} = \begin{cases} \eta_i \delta_{ij}; & \delta_{ij} \le 0; \\ \dfrac{1}{\eta_i} \delta_{ij}; & \delta_{ij} > 0; \end{cases} \\ &\qquad |\delta'_{ij}| \le \delta_{max_i}; \\ &\qquad 0 \le \sum_{k=1}^{j} \delta'_{ik} \le B_i. \end{aligned} \tag{12}$$

The objective function of the problem, defined in Eq. 12, measures the sum of energy costs over all households and the desired time interval. We call a solution, which aims to solve this problem, a cooperative solution; contrary, a competitive solution would put households against one another, focusing on individual costs rather than the cumulative cost of the community.

The objective is inherited by every household in the system, where the cost of energy for each household is computed proportionally to their energy output/input using Eq. 10. This ensures that BESS owners, who are the largest contributors to the community, are the main recipients of the benefits of such CEMS. Furthermore, in this CEMS, cumulative cost can be lowered more than if each household had an individual energy management system, creating incentive for both BESS owners and other households to participate. These claims are reinforced in Section VIII, where per household costs are discussed.

It is important to differentiate two views of the problem:

- **Offline:** g_{ij}, c_{ij}, p_j^e, p_j^d are known for all time interval steps $j = 1, \ldots, n$; our task is to find optimal δ_{ij} for our objective function.
- **Online:** $g_{ij}, c_{ij}, p_j^e, p_j^d$ are known only up to some current step k. Additionally, all δ_{ij} up to k have already been decided. Our task is to choose δ_{ik} such that we believe is the optimal value for our objective function in that time step.

Since CEMS must take decisions in real time, our ultimate goal is to solve the online version considering the current state and future uncertainties. As to the latter, if the offline version of the problem on historical data is solved, i.e. the "expert" policy is found, it can be utilized to learn an online policy. Taking this into account, we designed a solution to resolve the optimization problem defined in Eq. (12) following a three-step procedure, also visually depicted in Fig. 3:

1) First, we compute the optimal solution (δ^*) to the offline problem using LP, as described in Section V.
2) Next, we use the optimal solution (δ^*) to train DRL agents on offline data, as detailed in Section VI-A.
3) Finally, we deploy the trained DRL agents in an online environment and evaluate their performance, as discussed in Section VI-B.

V. OFFLINE SOLUTION

In section IV, we interpreted the problem of cost minimization as a constrained optimization problem. In this section we show that this problem is equivalent to LP, and how it can be interpreted as such. LP, apart from most other forms of deterministic global optimization, has guaranteed polynomial time complexity, allowing such problems to be solved more quickly.

The standard form of a linear program is written as:

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & Ax \geq b, \\ & x \geq 0; \end{aligned} \tag{13}$$

where c is the vector of coefficients in the cost function, x is the vector of variables, A is the matrix of constraint coefficients and b is the vector of bounds on constraints.

Alternatively, we can present our problem in the form:

$$\begin{aligned} \min \quad & c^T x \\ \text{subject to} \quad & l_1 \leq Ax \leq u_1, \\ & l_2 \leq x \leq u_2; \end{aligned} \tag{14}$$

where l_1, u_1, l_2 and u_2 are vectors of either real numbers or positive or negative infinities. They represent lower and upper bounds on the linear constraints and variables. It is known that the form in Eq. (14) can easily be translated into the form in Eq. (13) [43].

In the following we introduce a translation of our problem to a linear program for only a single household, after which we derive a linear program for a community of homes.

1) LP FORMULATION FOR A SINGLE HOUSEHOLD

Since we are considering only one household, we will temporarily omit index i from the variables for simpler notation. For our problem to become a linear program, we introduce several new variables: x_j^{in}, x_j^{out}, δ_j^{in}, δ_j^{out}, where:

$$x_j = x_j^{in} + x_j^{out}, \tag{15}$$

$$x_j^{in} \leq 0, x_j^{out} \geq 0, \tag{16}$$

and

$$\delta_j = \delta_j^{in} + \delta_j^{out}, \tag{17}$$

$$\delta_j^{in} \geq 0, \delta_j^{out} \leq 0. \tag{18}$$

Together they form the vector x and bound vectors l_2 and u_2 from Eq. (14):

$$x = \begin{bmatrix} x^{in} & x^{out} & \delta^{in} & \delta^{out} \end{bmatrix}^T, \tag{19}$$

$$l_2 = \begin{bmatrix} -\infty & 0 & 0 & -\infty \end{bmatrix}^T, \tag{20}$$

$$u_2 = \begin{bmatrix} 0 & \infty & \infty & 0 \end{bmatrix}^T. \tag{21}$$

The objective function then becomes:

$$y(x^{in}, x^{out}, \delta^{in}, \delta^{out}) = -\sum_{j=1}^{n} x_j^{in}(p_j^e + p_j^d) - \sum_{j=1}^{n} x_j^{out} p_j^e; \tag{22}$$

$$c = \begin{bmatrix} -(p^e + p^d) & -p^e & 0 & 0 \end{bmatrix}^T. \tag{23}$$

Note that neither δ^{in} nor δ^{out} explicitly affect the objective function, however they are needed as they play a role as constraints. Additionally, x_j^{in} and x_j^{out} should not be both non-zero for the same j. This relationship is not explicitly stated in the constraints as it requires multiplication of variables. However, this issue is resolved because of the nature of the objective function; p^e and p^d are both non-negative while x^{in} is non-positive and x^{out} is non-negative, rendering the case where x_j^{in} and x_j^{out} are both nonzero a sub-optimal solution. Similarly, δ_j^{in} and δ_j^{out} should also not be both non-zero for the same j in order to conform to the constraint described in Eq. 6 and one can show that this is the case at optimum.

Crucially, the following equality must hold for a single household for every step $j = 1, \ldots, n$:

$$g_j - c_j - \delta_j^{in} - \delta_j^{out} = x_j^{in} + x_j^{out}. \tag{24}$$

In block matrix form, where $\mathbf{I}$ is the identity matrix of size n:

$$\begin{bmatrix} \mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} x^{in} \\ x^{out} \\ \delta^{in} \\ \delta^{out} \end{bmatrix} = \begin{bmatrix} g - c \end{bmatrix}. \tag{25}$$

We must also address the constraint posed by the BESS maximal capacity. Here we introduce another variable δ_j', which will represent the change of SoC, whereas δ_j denotes

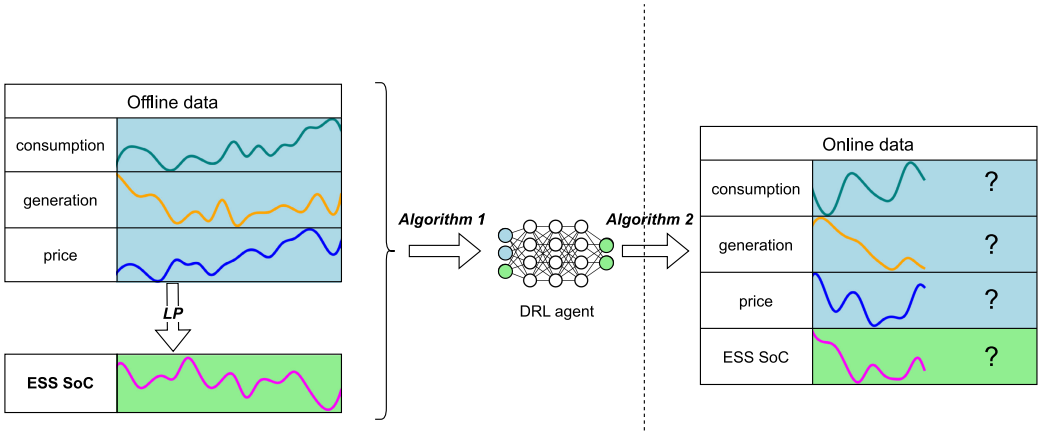


FIGURE 3. Illustration of the proposed training and deployment process of the MODREC solution. First, an optimal solution is obtained from offline data using LP. Then, agents are trained on this solution via Algorithm 1. Finally, the trained agents are deployed in an online environment using Algorithm 2.

the energy fed into the BESS (these values are different due to BESS charging losses):

$$\delta'_j = \frac{1}{\eta}\delta_j^{out} + \eta\delta_j^{in}, \tag{26}$$

where η is the charging efficiency with a value between 0 and 1 (usually slightly below 1). Note that for $\eta = 1$ (no loss) it follows that $\delta_j = \delta'_j$.

Constraint from Eq. (4) can thus be written as:

$$0 \leq \sum_{k=1}^{j}(\eta\delta_k^{in} + \frac{1}{\eta}\delta_k^{out}) \leq B. \tag{27}$$

In matrix form, where $\mathbf{L}$ is the lower triangular matrix of ones with size n:

$$0 \leq \begin{bmatrix}\eta\mathbf{L} & \frac{1}{\eta}\mathbf{L}\end{bmatrix}\begin{bmatrix}\delta^{in}\\ \delta^{out}\end{bmatrix} \leq B. \tag{28}$$

Constraint from Eq. (5) can be written as:

$$-\delta_{max} \leq \eta\delta_j^{in} + \frac{1}{\eta}\delta_j^{out} \leq \delta_{max}. \tag{29}$$

In matrix form:

$$-\delta_{max} \leq \begin{bmatrix}\eta\mathbf{I} & \frac{1}{\eta}\mathbf{I}\end{bmatrix}\begin{bmatrix}\delta^{in}\\ \delta^{out}\end{bmatrix} \leq \delta_{max}. \tag{30}$$

Finally, matrices A, l_1, l_2 of linear constraints take the following block matrix form:

$$A = \begin{bmatrix}\mathbf{I} & \mathbf{I} & \mathbf{I} & \mathbf{I}\\ 0 & 0 & \eta\mathbf{L} & \frac{1}{\eta}\mathbf{L}\\ 0 & 0 & \eta\mathbf{I} & \frac{1}{\eta}\mathbf{I}\end{bmatrix}; \tag{31}$$

$$l_1 = \begin{bmatrix}g-c\\ 0\\ -\delta_{max}\end{bmatrix}; \tag{32}$$

$$u_1 = \begin{bmatrix}g-c\\ B\\ \delta_{max}\end{bmatrix}. \tag{33}$$

2) LP FORMULATION FOR ANY NUMBER OF HOUSEHOLDS

In the previous subsection we described the translation of our problem into an LP when considering only one household. In this subsection we will expand the aforementioned translation and provide a generalized approach for a community of $m \in \mathbb{N}$ households, whereby in the case of a single household ($m = 1$), all equations in this section will simplify to equations given in section V-1.

To generalize from the single household to m households, it is essential that we adjust the energy flow constraint:

$$\sum_{i=1}^{m} g_{ij} - c_{ij} - \delta_{ij}^{in} - \delta_{ij}^{out} = x_j^{in} + x_j^{out}. \tag{34}$$

Energy demand / surplus of individual households will be implicitly calculated as $x_{ij} = g_{ij} - c_{ij} - \delta_{ij}^{in} - \delta_{ij}^{out}$.

Let the new LP variables be:

$$x = \begin{bmatrix}x^{in} & x^{out} & \delta_1^{in} & \delta_1^{out} & \cdots & \delta_m^{in} & \delta_m^{out}\end{bmatrix}^T \in \mathbb{R}^{2(m+1)n}. \tag{35}$$

Then the LP coefficients and bounds of the entire community become:

$$P = \begin{bmatrix}\mathbf{I} & \mathbf{I}\end{bmatrix} \in \mathbb{R}^{n\times 2n}$$

$$Q_i = \begin{bmatrix}\eta_i\mathbf{L} & \frac{1}{\eta_i}\mathbf{L}\\ \eta_i\mathbf{I} & \frac{1}{\eta_i}\mathbf{I}\end{bmatrix} \in \mathbb{R}^{2n\times 2n}$$

$$l_{1i} = \begin{bmatrix}0 & -\delta_{max}\end{bmatrix}^T$$

$$u_{1i} = \begin{bmatrix}B_i & \delta_{max}\end{bmatrix}^T; \tag{36}$$

$$c = \begin{bmatrix}p^e + p^d & p^e & 0 & \cdots & 0\end{bmatrix}; \tag{37}$$

$$A = \begin{bmatrix}P & P & P & \cdots & P & P\\ 0 & Q_1 & 0 & \cdots & 0 & 0\\ 0 & 0 & Q_2 & \cdots & 0 & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots & Q_{m-1} & 0\\ 0 & 0 & 0 & \cdots & 0 & Q_m\end{bmatrix}; \tag{38}$$

$$l_1 = \begin{bmatrix}\sum_{i=1}^{m} g_i - c_i & l_{11} & l_{12} & \cdots & l_{1i} & \cdots & l_{1m}\end{bmatrix}^T;$$

$$u_1 = \left[\sum_{i=1}^{m} g_i - c_i \quad u_{11} \quad u_{12} \quad \cdots \quad u_{1i} \quad \cdots \quad u_{1m}\right]^T;$$
$$l_2 = \left[l_{21} \quad l_{22} \quad \cdots \quad l_{2i} \quad \cdots \quad l_{2m}\right]^T;$$
$$u_2 = \left[u_{21} \quad u_{22} \quad \cdots \quad u_{2i} \quad \cdots \quad u_{2m}\right]^T. \tag{39}$$

The length of vector x is $2(m+1)n$ and the size of matrix A is $(2m+1)n \times 2(m+1)n$. In other words, there are $2(m+1)n$ variables and $(2m+1)n$ constraints for a community of m households and n time steps.

We denote the optimal solution for this problem as x^*, and we can easily compute the optimal charging / discharging policy as:

$$\delta'^{*}_{ij} = \frac{1}{\eta_i}\delta^{out*}_{ij} + \eta_i \delta^{in*}_{ij}. \tag{40}$$

In the simulations, we utilised the linear programming solver from the `scipy` [44] Python library, supported by the dual revised simplex method introduced in [45], which is part of the HiGHS software package.

VI. PROPOSED ONLINE SOLUTION

In the previous section we introduced the problem as an offline optimization problem. However, in the case of home energy management, we are constrained when making a decision, i.e., selecting δ_{ij} at step j, as the future information regarding the consumption, energy generation and price are unknown. To that end we propose Mathematical Optimization and Deep Reinforcement learning for Energy Cost minimization (MODREC) solution.

A. TRAINING PROPOSED SOLUTION - EXPERT TRAINING

To overcome aforementioned practical issue of unavailable future information, we propose an online learner to solve the problem based on MADRL as it allows designing a decentralized, general and scalable solution:

1) **Decentralisation:** All active households in the community should have equivalent roles, making the solution fair for all participants and preventing single point of failure, i.e. the failure of the Home Energy, Management System (HEMS) of one household does not affect CEMS.
2) **Generality:** The proposed framework can be applied to any number of households as every agent is responsible for managing only its respective household. This means that MODREC can also be applied in the single household scenario.
3) **Scalability:** The proposed framework is scalable such that data intensity of the system model grows linearly with the number of households m. The effect of scaling on the LP part of the framework is discussed in section V-2 and on the DRL part is discussed in this section.

In this section we describe the interpretation of the problem as a Markov Decision Process (MDP) and define states, actions and reward functions, which are standardly required when solving problems using RL. We describe DQN, the underlying algorithm of our solution, as well as how multiple agents work together to achieve a better result.

An important part of the solution to the online problem is the definition of states, actions, and rewards.

States s: States are shared amongst all agents and we define the state as such:

$$\mathbf{s}_j = \begin{bmatrix} \mathbf{s}_{1,j-kw} & \mathbf{s}_{1,j-(k-1)w} & \cdots & \mathbf{s}_{1,j-w} \\ \mathbf{s}_{2,j-kw} & \mathbf{s}_{2,j-(k-1)w} & \cdots & \mathbf{s}_{2,j-w} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_{i,j-kw} & \mathbf{s}_{i,j-(k-1)w} & \cdots & \mathbf{s}_{i,j-w} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{s}_{m,j-kw} & \mathbf{s}_{m,j-(k-1)w} & \cdots & \mathbf{s}_{m,j-w} \\ p_{e,j-kw} & p_{e,j-(k-1)w} & \cdots & p_{e,j-w} \\ p_{d,j-kw} & p_{d,j-(k-1)w} & \cdots & p_{d,j-w} \\ t_{d,j-kw} & t_{d,j-(k-1)w} & \cdots & t_{d,j-w} \\ t_{w,j-kw} & t_{w,j-(k-1)w} & \cdots & t_{w,j-w} \\ t_{y,j-kw} & t_{y,j-(k-1)w} & \cdots & t_{y,j-w} \end{bmatrix}, \tag{41}$$

$$\mathbf{s}_{i,j-(k-k')w} = \begin{bmatrix} g_{i,j-(k-k')w} \\ c_{i,j-(k-k')w} \\ SoC_{i,j-(k-k')w} \end{bmatrix}. \tag{42}$$

The state is therefore defined as a sliding window with memory of k steps. A step width w is used to capture longer trends while reducing the state size as to decrease the number of neural network parameters. It includes data about generation, consumption and SoC for all households as well as p_e and p_d. There is also data about time, as energy price is often dependent on the time in a day t_d (peak and off-peak hours), the day in a week t_w (working and free days) and also on the season of the year through the day in a year t_y (to capture seasonal effects). In practice, this state information is exchanged among households at 15-minute intervals rather than continuously, thereby reducing communication overhead. Furthermore, all shared variables are anonymized so that household identities cannot be inferred, although participation within the energy community framework is assumed to be voluntary. Conclusively, a state is represented by a matrix $\in \mathbb{R}^{3m+5\times k}$ for m households and the chosen sliding window size k.

Actions a: Actions define how much energy should be charged to / discharged from the BESS. There are five available actions that influence δ and they can take values $a_{ij} \in 0, 1, 2, 3, 4$. In fact, δ_{ij} is a function of a_{ij}:

$$\delta_{ij}(a_{ij}) = \begin{cases} f_i(a_{ij}); & SoC_{i,j-1} + f_i(a_{ij}) \in [0, B_i] \\ -SoC_{i,j-1}; & SoC_{i,j-1} + f_i(a_{ij}) < 0 \\ B_i - SoC_{i,j-1}; & B_i < SoC_{i,j-1} + f_i(a_{ij}), \end{cases} \tag{43}$$

$$f_i(a_{ij}) = \begin{cases} \delta_{max_i}; & a_{ij} = 0 \\ \frac{1}{2}\delta_{max_i}; & a_{ij} = 1 \\ 0; & a_{ij} = 2 \\ -\frac{1}{2}\delta_{max_i}; & a_{ij} = 3 \\ -\delta_{max_i}; & a_{ij} = 4. \end{cases} \tag{44}$$

Reward function R: The reward function for agent i at step j is defined as:

$$R^*_{ij} = 1 - \frac{|SoC^*_{ij} - SoC_{ij}|}{B_i}, \tag{45}$$

where $SoC^*_{ij} = \sum_{k=1}^{j} \delta'^*_{ik}$ is the SoC of the optimal solution provided via LP and SoC_{ij} is the SoC of the CEMS guided by DRL. However, this reward function is applicable only in the context of an offline solution. For online learning, we implement an alternative reward function as follows:

$$R'_{ij} = -y_j, \tag{46}$$

where y_j is the cumulative energy cost paid by the entire community at step j.

1) MULTI-AGENT DRL

We outline the training of the proposed solution MODREC in Algorithm 1, based on the DQN algorithm [46]. DQN combines Q-learning with deep neural networks to facilitate efficient learning of optimal decision-making strategies. At its core, DQN leverages a neural network to approximate Q-values, which are estimates of the expected cumulative reward associated with each state-action pair. Through interaction with the environment, the agent collects experiences, formally represented as state–action–reward–next state tuples (s_t, a_t, r_t, s_{t+1}), which are stored in a memory buffer for subsequent expert training. Periodically, the algorithm samples batches of experiences (line 16) to compute target Q-values based on the Bellman equation, balancing between immediate and expected future rewards. By minimizing the discrepancy between predicted and target Q-values through minimizing the cost function (line 20), the neural network gradually refines its estimations. To ensure stability, DQN employs a separate target network, periodically updated with update rate λ (line 22) to align with the primary network. With an ϵ-greedy exploration, the agent navigates between exploiting its current knowledge and exploring the environment. ϵ-greedy exploration works by selecting random actions with some probability, thus addressing uncertainty in the environment. After the action a_{ij} is selected (line 12), the system determines the amount of energy it has to buy from or sell to the grid using the formula described in Eq. (43).

In the multi-agent approach, every household i is managed by its own corresponding agent (its own policy network Q_i, target network Q'_i, replay memory D_i and rewards R_i). However, they share among themselves the same states $\mathbf{s}$, meaning that every agent holds information about all other households, which helps them making more informed decisions towards a joint global objective. The training of the proposed solution is divided into 2 parts:

1) **Imitating:** The imitating part is conducted for N_{eps} episodes and it is intended to train the agent to take actions that most closely follow the ones that were determined by the offline solution. It uses the reward function described in Eq. (45). By using a reward that considers the performance of the entire community, we encourage cooperation between agents.
2) **Fine-tuning:** For one additional episode of the training process (without initialization of the networks), the reward function described with Eq. (46) is used to replace the reward function of Algorithm 1 in line 14. This part of training is intended for the agents to adapt to the actual measure they are maximizing.

Algorithm 1 Expert Training of Proposed MODREC Solution

1: **for** $i = 1, \ldots, m$ **do**
2: Randomly initialise policy network $Q_i(\mathbf{s}, a|\theta)$
3: Initialise target network Q'_i with weights $\theta'_i \leftarrow \theta_i$
4: Initialise replay memory $\mathcal{D}_i$ to capacity D
5: Fill replay memory $\mathcal{D}_i$ with experience extracted from the LP solution
6: Observe initial state $\mathbf{s}_j$ at time-step $j = 0$
7: **end for**
8: **for** $k = 1, \ldots, N_{eps}$ **do**
9: **for** $j = 1, \ldots, n$ **do**
10: **for** $i = 1, \ldots, m$ **do**
11: With probability $\epsilon_k = p_\epsilon^{k-1}$ select a random action
12: Otherwise select $a_{ij} = \arg\max_a Q_i(\mathbf{s}_j, a|\theta)$
13: Calculate x_{ij}, and y_{ij}
14: Observe $\mathbf{s}_{j+1}$ and determine R_{ij} with Eq. (45)
15: Store experience $\mathbf{s}_{ij}, \mathbf{s}_{j+1}, R_{ij}, a_{ij}$ in $\mathcal{D}$
16: Sample random batch of J experiences from $\mathcal{D}$
17: **for** every $\{\mathbf{s}_j, \mathbf{s}_{j+1}, R_{ij}, a_{ij}\}$ in batch **do**
18: $y'_{ij} = R_{ij} + \gamma max_{a_{j+1}} Q'_i(\mathbf{s}_{j+1}, a_{i(j+1)})$
19: **end for**
20: Calculate loss: $\mathcal{Z} = \frac{1}{J}\sum_{j=0}^{J-1}(Q_i(\mathbf{s}_j, a_{ij}) - y'_{ij})^2$
21: Update $Q_i(\mathbf{s}, a|\theta_i)$ by minimizing the loss $\mathcal{Z}$
22: Softly update the target network: $\theta'_i \leftarrow \lambda\theta_i + (1-\lambda)\theta'_i$
23: **end for**
24: **end for**
25: **end for**

B. DEPLOYING PROPOSED SOLUTION

Once training is completed, the agents are deployed in an online setting using the reward function from Eq.(46) with a small exploration rate, set to $\epsilon = 0.01$. The deployment procedure is summarized in Algorithm 2. Unlike the training phase, this stage does not require network initialization, as it operates directly with the pre-trained agents and their corresponding networks. Furthermore, the reward function is adapted: the function used during training (Algorithm 1, line 14) is replaced with the deployment reward function specified in Algorithm 2 (line 9).

It is important to note that this phase represents the solution operating in a real environment. In our work, however, deployment is conducted within a simulation framework, which allows us to evaluate and validate the proposed solution under controlled conditions. The results of this validation are presented in the following sections.

Algorithm 2 Proposed MODREC Online Solution

for $i = 1, \ldots, m$ **do**
  Observe initial state $\mathbf{s}_j$ at time-step $j = 0$
end for
for $j = 1, \ldots, n$ **do**
  for $i = 1, \ldots, m$ **do**
    With probability ϵ select a random action
    Otherwise select $a_{ij} = \arg\max_a Q_i(\mathbf{s}_j, a|\theta)$
    Calculate x_{ij}, and y_{ij}
    Observe $\mathbf{s}_{j+1}$ and determine R_{ij} with Eq. (46)
    Store experience $\mathbf{s}_{ij}, \mathbf{s}_{j+1}, R_{ij}, a_{ij}$ in $\mathcal{D}$
    Sample random batch of J experiences from $\mathcal{D}$
    for every $\{\mathbf{s}_j, \mathbf{s}_{j+1}, R_{ij}, a_{ij}\}$ in batch **do**
      $y'_{ij} = R_{ij} + \gamma max_{a_{j+1}} Q'_i(\mathbf{s}_{j+1}, a_{i(j+1)})$
    end for
    Calculate loss: $\mathcal{Z} = \frac{1}{J}\sum_{j=0}^{J-1}(Q_i(\mathbf{s}_j, a_{ij}) - y'_{ij})^2$
    Update $Q_i(\mathbf{s}, a|\theta_i)$ by minimizing the loss $\mathcal{Z}$
    Softly update the target network: $\theta'_i \leftarrow \lambda\theta_i + (1-\lambda)\theta'_i$
  end for
end for

TABLE 3. Neural network and DQN parameters.

Parameter	Value
Learning rate	0.01
Batch size (J)	256
Memory capacity (D)	8000
Number of dense layers (l)	2
Dense layer size (d)	16
Activation function	ReLU
Target network update frequency	500
λ	0.9
N_{eps}	12
p_ϵ	0.8
k	12
w	8

C. PARAMETERS

Our solution employs a neural network with parameters summarized in Table 3. The N_{eps} parameter is determined using a validation set, whose impact on performance is illustrated in Fig.7, as training time significantly influences the results. Due to computational constraints, exhaustive search of all hyper-parameters was impossible. Still, other hyper-parameters were chosen either with a grid search or by using standard values used in such algorithms, verified in previous research [46]. The neural network architecture was decided as the optimal ratio between performance and speed.

VII. EVALUATION METHODOLOGY

We evaluated the proposed solution by performing several numerical simulation case studies. In this section we present the evaluation methodology. We first explain the structure and sources of the dataset used in simulations and then discuss computational complexity of MODREC. Then we introduce three baseline policies and the main metric with which MODREC will be evaluated; we call it Normalized Score. In the next section we discuss the performance of MODREC on said metric and also delve into benefits of MODREC regarding load shifting.

A. DATA

In this section, we describe the acquisition of data on energy consumption, generation, and prices. The energy generation and price data correspond to an approximate geographical area centered around Ljubljana, Slovenia.

1) **Consumption:** Household consumption was modelled using an open source tool for generating load profiles, `pyloadprofilegenerator` [7].
2) **Generation:** Information about solar energy generation was gathered using the web application PVGIS [8].
3) **Price:** We use real-world time-varying Slovenian market prices at 15-minute resolution, which naturally exhibit substantial variability across days and seasons. Transmission charge p^d in this study is set to 0.25 p^e.

The final dataset consists of data points that are 15 minutes apart and span from 1^{st} January 2022 at 0:00 until 31^{st} December 2023 at 23:45. Consequently, in our evaluation, MODREC makes decisions at 15-minute intervals, since using a finer time step would require interpolation beyond the available measurements. By combining synthetically generated consumption profiles with real-world solar generation and electricity price data, the resulting dataset offers a flexible, scenario-driven representation aligned with European household behavior and current market conditions. Additionally, the dynamic pricing data require the solution to continuously adapt to time-varying price realizations; therefore, our evaluation reflects adaptation to diverse, time-varying price conditions within the considered community. Although several established energy community datasets exist, their limitations in terms of scale, appliance-level completeness, temporal coverage, or geographical relevance led us to adopt synthetic consumption data. This approach enabled the construction of a scalable and context-aware dataset, tailored to the objectives and regional focus of our study, and well-suited for demonstrating and evaluating the proposed MODREC solution.

B. COMPUTATIONAL COMPLEXITY

We evaluated the computational complexity and execution time of the deep learning component of the proposed MODREC approach in a 5-household community setting, i.e., $m = 5$, using a full year of data recorded at 15-minute intervals. The evaluation is shown in Table 4. The experiments were conducted on an AMD EPYC 75F3 32-Core Processor. We compare the performance of the LP optimization with the MADRL training process. Interestingly, in both approaches, the number of households impacts the complexity; however, the primary contributing factors differ. For the LP solution, the number of time steps is most significant, while for the DQN-based method, the complexity is primarily influenced by the neural network

TABLE 4. Comparison of parameter complexity and execution time.

Method	Computational Complexity	Execution Time (m=5, n=98, l=2, d=16)
LP	$\mathcal{O}(2(m+1)n)$	341 s
DQN	$\mathcal{O}(mld)$	484 s

size, more specifically, the number of hidden layers and neurons per layer. We also benchmarked the execution time for each method. For example, the LP method was decomposed into 98 subproblems and required a total of 5 minutes and 41 seconds to solve. In contrast, training agents using DQN averaged 8 minutes and 4 seconds per episode. Consequently, we conclude that although different factors influence the computational complexity of each approach, the time required to update them is comparable. It is important to note that the reported results do not consider response time, i.e., the time required to make a decision. In the case of the proposed MODREC framework, decision-making involves a simple inference step using a pre-trained neural network. As such, the response time is on the order of seconds, making the approach suitable for (near) real-time applications.

C. REFERENCE POLICIES

The performance of the proposed MODREC solution is compared to 3 baseline policies used as a reference:

- *Naive* policy is a rule-based policy, which sells any potential generated energy surplus immediately, not relying on energy storage. It is considered as a lower bound for our solution (it results in higher cumulative cost, but is worse in terms of performance). The cumulative cost on the test set will be denoted as y_l.
- *Clairvoyant* policy is devised by using LP on the test set, assuming all variables, such as consumption, generation and price, are known for the entire interval. In other words, this is a baseline assuming a perfect forecast of unknown variables. It is impossible to achieve a better result than this baseline and is considered as an upper bound for our solution (again, we interpret the resulting lower cumulative cost as a better result). The cumulative cost on the test set will be denoted as y_u.
- *LP+LogReg* policy is a policy similar to our proposed method, with logistic regression replacing the DRL agent. Essentially, each household with a BESS is assigned a logistic regression model, where the LP results on past data are used as the training data for the models. A similar approach has shown promising results as a stock trading strategy [47].

For robust evaluation of the performance of the MODREC solution we define a benchmark score function, comparing our method to both baselines. Similar scores have been used to evaluate RL models [48].

$$\text{Normalized Score} = \frac{y_l - y}{y_l - y_u} \tag{47}$$

Here, y is the cumulative cost of energy for a community managed by MODREC. This scoring function maps the cumulative cost of a model to an interval [0, 1]. It takes the value of 0 if the cumulative cost of the proposed solution is the same as the Naive baseline's cumulative cost and the value of 1 if the proposed solution's cumulative cost is the same as the cumulative cost of the Clairvoyant baseline. We interpret the model's performance as better when its Normalized Score is higher, and worse when the Normalized Score is lower. It could occur that the score might be lower than 0; that would happen if the cumulative cost produced by the investigated model was higher than that of the Naive baseline.

VIII. EVALUATION

In this section we present results of how MODREC[1] performed in a simulated multi-household environment. We look at several aspects: firstly we discuss the reduction in energy costs for a 5-household community in detail, and additionally present reductions in energy costs for communities of ten and twenty households. Then we analyze how closely MODREC imitates the optimal behavior and we also look at how resistant it is to loss of information. Lastly, we argue that MODREC helps with balancing the load of the external energy grid by offsetting BESS charging to off-peak times of the day.

A. DESCRIPTION OF CASE STUDIES

We evaluate MODREC through three case studies consisting of communities of five, ten, and twenty households. The main parameters of these case studies are summarized in Tables 5, 6, and 7. For each household, the tables report the maximum BESS capacity (B), average annual consumption, average photovoltaic (PV) generation, charging/discharging power (δ_{max}), and efficiency (η). Households without a BESS are denoted with "/". Case study 2 differs from the others by employing a higher δ_{max}, i.e., increased charging/discharging power, which leads to higher potential savings due to faster energy transfer. In all case studies, 40% of households are assumed to lack a BESS and therefore do not directly influence the simulation outcomes. All BESSs have a capacity of 5kWh, respecting realistic BESS capacities.

B. ENERGY COST SAVINGS

Annual energy costs per household for case study 1, in absolute values, are depicted in Fig. 4 for the baseline policies and for MODREC. Although still higher than the Clairvoyant baseline, the MODREC policy results in annual savings of €819 (i.e. 28% less than the Naive baseline). Additionally, while costs for households without PV and a BESS remain relatively unchanged, those with a BESS see a reduction of nearly €300 in annual costs.

Additionally, the results demonstrate that the proposed MODREC solution outperforms both the Naive and LP+LogReg baseline strategies. This is most evident when evaluating the cumulative energy cost across the entire

[1]The implementation of MODREC is available on GitHub: https://github.com/sensorlab/smart-community-energy-management-with-LP-DRL

TABLE 5. Household parameters for 5 households.

	B (kWh)	c (MWh/year)	g (MWh/year)	$\delta_{\max}$ (kWh/15min)	η (%)
H1	5	6.51	7.95	1	98
H2	5	8.15	7.95	1	98
H3	/	5.26	0	/	/
H4	5	4.29	7.95	1	98
H5	/	6.32	0	/	/

TABLE 6. Household parameters for 10 households.

	B (kWh)	c (MWh/year)	g (MWh/year)	$\delta_{\max}$ (kWh/15min)	η (%)
H1	5	6.42	7.95	1	98
H2	5	8.67	7.95	1	98
H3	5	5.55	0	1	98
H4	5	4.56	7.95	1	98
H5	5	6.71	0	1	98
H6	5	7.23	7.95	1	98
H7	/	9.88	7.95	/	/
H8	/	5.04	0	/	/
H9	/	4.30	7.95	/	/
H10	/	6.45	0	/	/

five-household community. Specifically, the Naive approach yields a total cost of €2,894, while the LP+LogReg policy results in a modest improvement, reducing the cost to €2,636. In contrast, the MODREC solution achieves a significantly lower cumulative cost of €2,075. Such a result, represents savings of over €800 compared to the Naive baseline. It is important to note, however, that the Clairvoyant policy, which represents an idealized upper bound with perfect foresight, still achieves a lower total cost. This indicates that while MODREC performs substantially better than comparable practical methods, there remains some potential for improvement within the system.

In Fig. 5, mean household energy costs for communities of 5, 10 and 20 households are presented side by side, with comparisons between all investigated methods. Evidently, MODREC outperforms the Naive and LP+LogReg baselines,

TABLE 7. Household parameters for 20 households.

	B (kWh)	c (MWh/year)	g (MWh/year)	$\delta_{\max}$ (kWh/15min)	η (%)
H1	5	6.42	7.95	1	98
H2	5	8.67	7.95	1	98
H3	5	5.55	0	1	98
H4	5	4.56	7.95	1	98
H5	5	6.71	0	1	98
H6	5	7.23	7.95	1	98
H7	5	9.88	7.95	1	98
H8	5	5.04	0	1	98
H9	5	4.30	7.95	1	98
H10	5	6.45	0	1	98
H11	5	9.39	7.95	1	98
H12	5	2.88	7.95	1	98
H13	/	6.58	0	/	/
H14	/	6.31	7.95	/	/
H15	/	5.95	0	/	/
H16	/	8.69	7.95	/	/
H17	/	8.02	7.95	/	/
H18	/	6.86	0	/	/
H19	/	4.88	7.95	/	/
H20	/	6.12	0	/	/

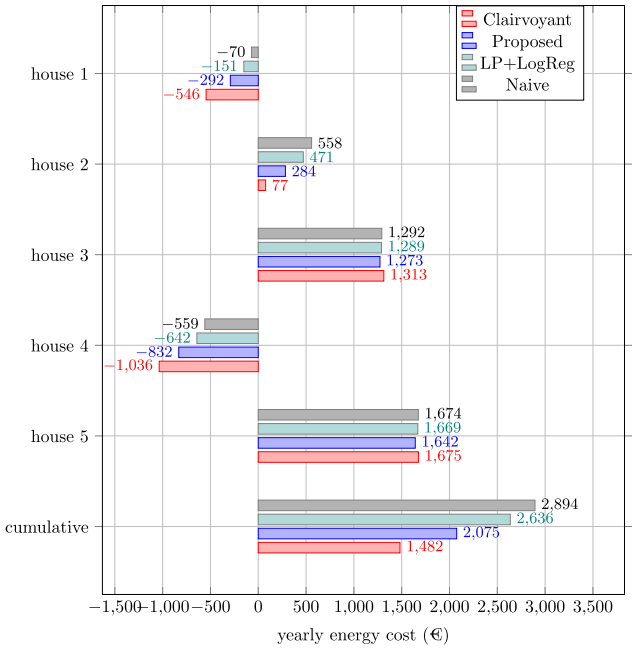


FIGURE 4. Annual energy costs for the 5-household community. It is evident that costs for households with a BESS are substantially reduced, but not to the detriment of other households.

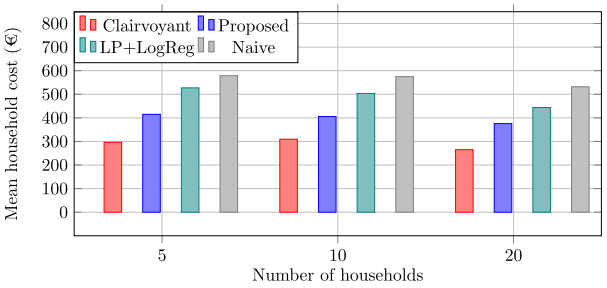


FIGURE 5. Comparison between annual energy costs for all investigated methods across different community sizes. The y-axis shows mean household energy cost.

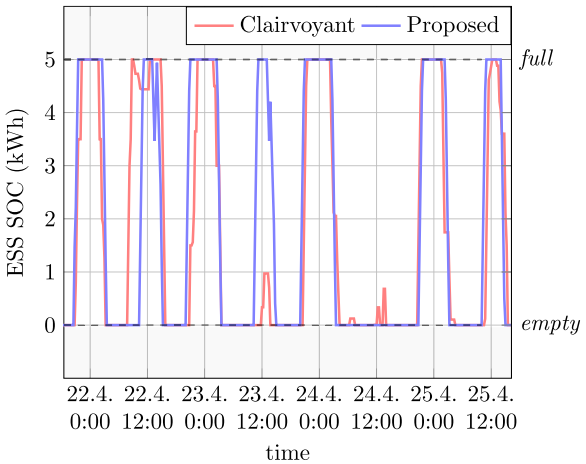
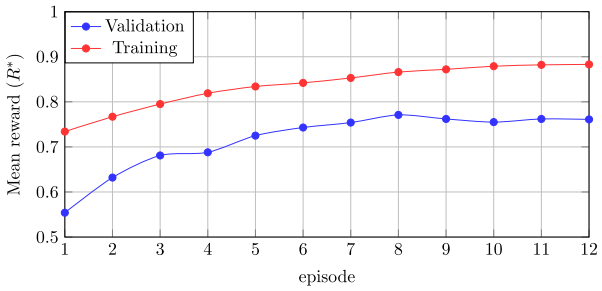


FIGURE 6. SoC comparison for the Clairvoyant and MODREC policies.

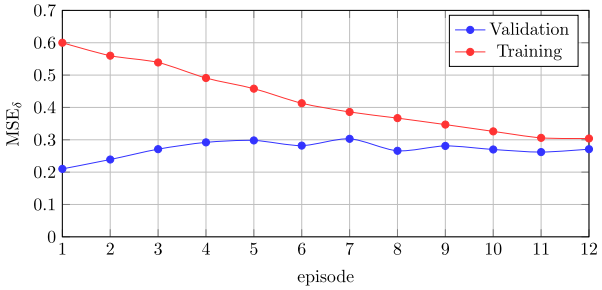
reinforcing our claim that MODREC is applicable to diverse communities. The performance of MODREC relative to

TABLE 8. Results on the case studies. Normalized Score reflects how well the agents imitate the Clairvoyant baseline.

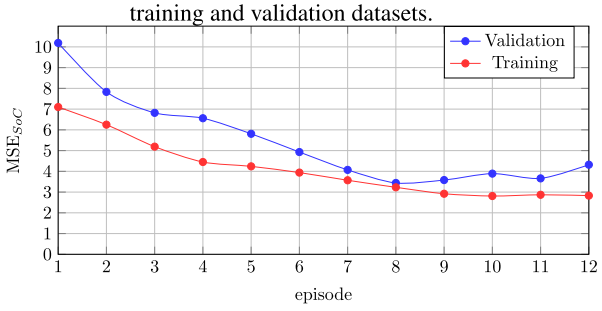
# households	Normalized Score	annual savings (vs. Naive)	Mean R^*	MSE_δ	MSE_{SoC}
5	0.58	28%	0.7847	0.2670	2.9178
10	0.63	29%	0.7912	0.2836	3.2547
20	0.58	29%	0.7674	0.2800	3.8120



(a) Mean reward (R^*, refer to Eq. 45) across episodes for the training and validation datasets.



(b) Mean Squared Error (MSE) w.r.t. δ across episodes for the training and validation datasets.



(c) MSE w.r.t. SoC across episodes for the training and validation datasets.

FIGURE 7. Change of the mean reward and MSE w.r.t. δ and SoC on the training and validation datasets across training for the 5-household case.

the baselines is the same irrespective of the number of households.

Table 8 summarizes the scores for the case studies across 5, 10 and 20 households. The Normalized Scores fall between the baselines, with the 10-household case having a bit higher Normalized Score. Additionally, annual cost savings in terms of percentage of the Naive baseline cost, Mean R^* and MSE_δ stay about the same. MSE_{SoC} increases with the number of households.

C. IMITATION OF OPTIMAL SOLUTION

As described in Section VI-A, MODREC is trained to imitate the optimal policy. This behaviour is evident in Fig. 6, where

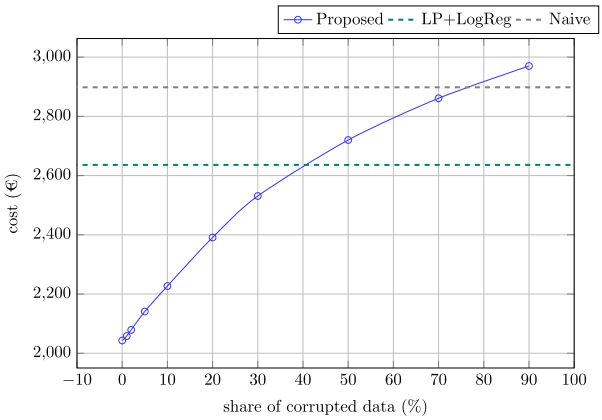


FIGURE 8. Performance of MODREC subjected to data corruption. MODREC performs better with up to 40% of data corrupted than the LP+LogReg baseline with no data corruption.

the SoC values of the house 1 BESS are compared for MODREC and the Clairvoyant baseline for a particular time interval of 4 days. MODREC closely imitates the Clairvoyant baseline policy, but due to uncertainty about future conditions some decisions are different. With this figure we demonstrate that the DRL agents actually learn to imitate the optimal policy. This is important because optimal behavior is only possible in offline (deterministic) settings, but MODREC is able to transfer this knowledge into an online environment.

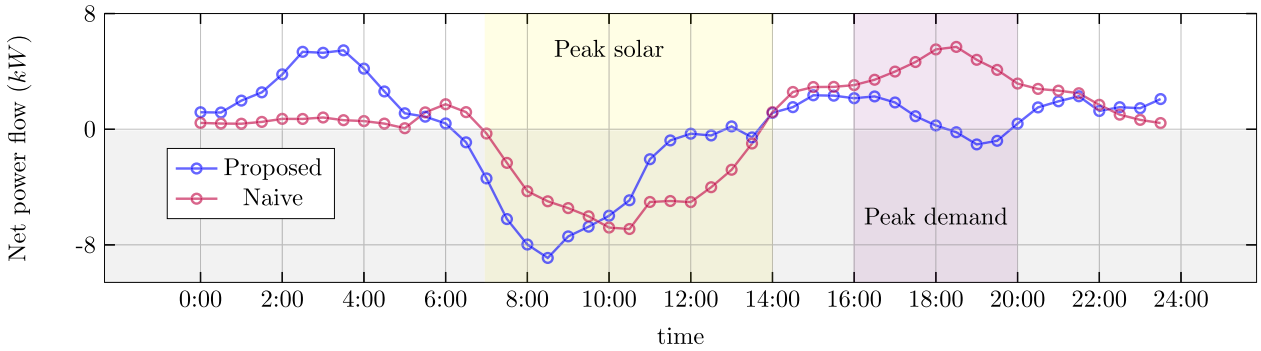
Furthermore, we quantify the accuracy of prediction by computing the Mean Squared Error (MSE) separately between action values (δ) and SoC values of MODREC and the optimal policy. In this case, the MSE with respect to δ is given in Eq. 48 and with respect to SoC in Eq. 49:

$$\text{MSE}_\delta = \frac{1}{m_{BESS} n} \sum_{B_i \neq 0} \sum_{j=1}^{N} (\delta_{ij} - \delta^*_{ij})^2 \tag{48}$$

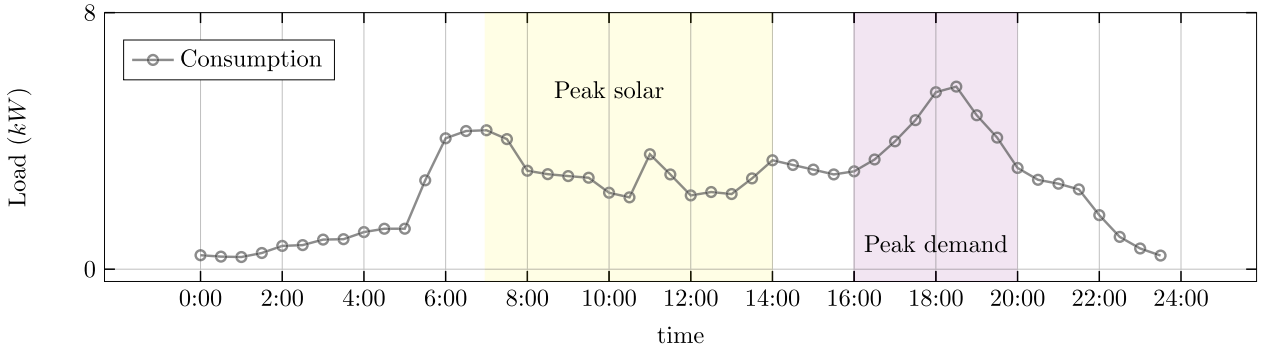
$$\text{MSE}_{SoC} = \frac{1}{m_{BESS} n} \sum_{B_i \neq 0} \sum_{j=1}^{N} (SoC_{ij} - SoC^*_{ij})^2 \tag{49}$$

Note that above equations only considers households that own a BESS, where m_{BESS} is the number of such households.

In Fig. 7, we investigate the ability of MODREC to imitate the optimal policy throughout the training process by plotting R^* from Eq. 45, which measures how closely MODREC imitates SoC of the optimal policy, MSE_δ and MSE_{SoC}. A stark difference is evident for the validation dataset as during training MODREC's R^* is increased and MSE_{SoC} is decreased, while MSE_δ remains about the same. An argument is to be made that R^* is better measure to consider as this way



(a) Averaged community net power flow comparison between the Naive baseline and the proposed method MODREC for January 2023.



(b) Averaged community consumption for January 2023.

FIGURE 9. Above is shown the comparison between the Naive baseline and MODREC in the temporal dimension. Our solution manages to decrease load during morning and evening peaks by increasing it during low demand periods such as night time.

MODREC aims to preserve the optimal state of CEMS, while MSE errors might cause CEMS state to drift away from the optimum with time.

As shown in Fig.7, training is performed with gradually decreasing randomness (see Algorithm 1), whereas validation is conducted with near-zero randomness (see Algorithm 2). The results indicate that the mean reward increases for both training and validation sets. Interestingly, MSE_δ decreases only for the training set but slightly increases for the validation set, while MSE_{SoC} decreases in both cases. This behavior can be attributed to the objective of MODREC during training, which is to model the BESS SoC of the optimal policy (captured by R^* and MSE_{SoC}), rather than to directly model the optimal policy's actions (captured by MSE_δ). Although these objectives are highly correlated, approximating the actions of the optimal policy may, over many time steps, accumulate errors that result in deviations in system state.

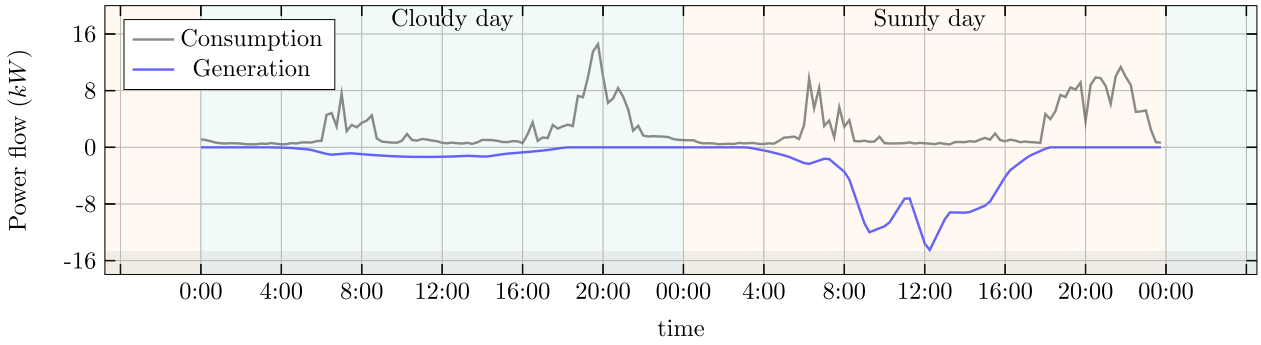
D. FAULT TOLERANCE

An additional aspect examined in this study is the fault tolerance of the proposed MODREC solution. In particular, we evaluate its ability to maintain performance when portions of the input data, such as energy consumption, generation, pricing, or BESS SoC, are missing or corrupted. To simulate this, controlled data corruption is introduced into the test dataset by randomly setting each data point to zero with a specified probability. The results, shown in Fig. 8, illustrate the impact of increasing data corruption probabilities on system performance. As expected, the cumulative energy cost grows with higher levels of data corruption. Nevertheless, even under moderate data corruption level (0–40%), MODREC consistently outperforms the LP+LogReg baseline evaluated on uncorrupted data.

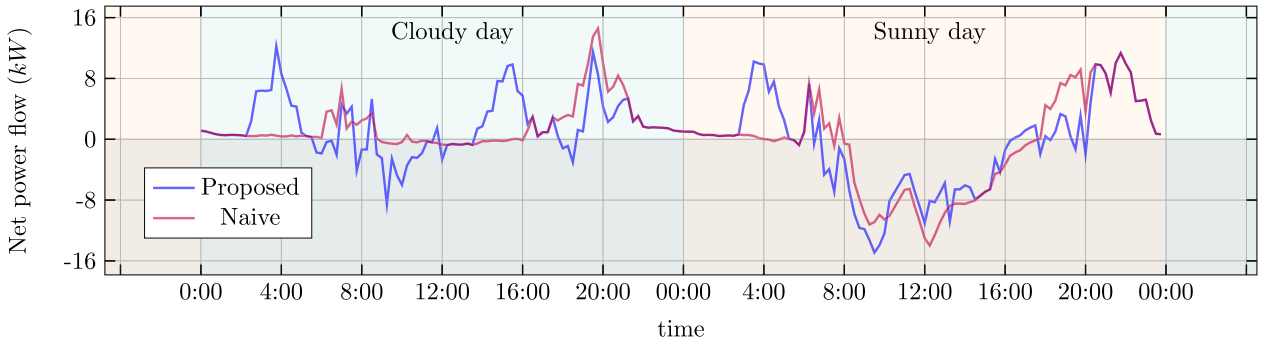
Furthermore, performance decline remains gradual and it does not critically impair the overall system functionality. This robustness highlights the suitability of MODREC for real-world deployment, where energy management systems must frequently contend with imperfect data resulting from network outages, latency, or sensor malfunctions. The ability to sustain acceptable performance under such conditions reinforces both the practical applicability and reliability of the proposed solution in operational environments.

E. CHANGES TO LOAD PROFILE

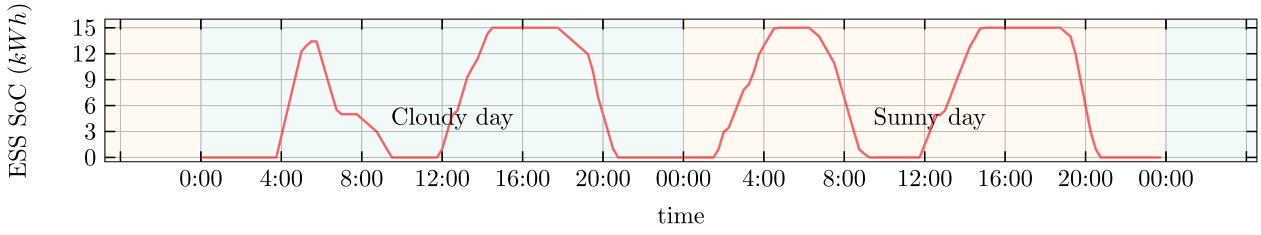
An additional benefit of MODREC is that, due to its price-responsive nature, it consequently performs load shifting, in turn helping to balance the energy grid. In Fig.9 a comparison of average daily load profiles for January 2023 in case study 1 is presented. In high demand periods, such as in the morning between 6:00 and 9:00 and in the evening between 16:00 and 21:00, MODREC discharges



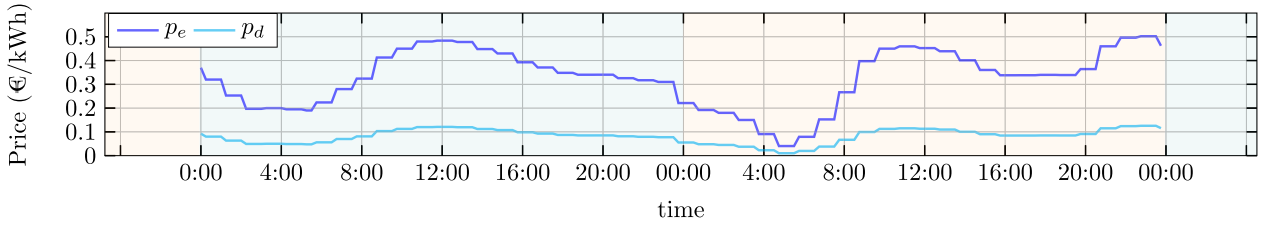
(a) Consumption and generation power during 2 days. The blue curve on the cloudy day is almost flat, signifying low production of solar energy.



(b) Load profiles of the communities managed by the Naive Baseline and MODREC.



(c) Cumulative BESS SoC of CEMS managed by MODREC.



(d) Energy Prices

FIGURE 10. **Load profile comparison for the Naive baseline and MODREC for two consecutive days, one cloudy and one sunny, along with cumulative CEMS SoC and energy prices for that interval.**

the community BESSs, reducing its own demand and also outputing power to the grid for it to be consumed elsewhere where demand is high. The BESSs get recharged during periods of high solar production around midday and from the grid in night time, when consumption is low and price is lower.

In Fig.10 the load profile of the 5-household community during a cloudy and a sunny day in May 2023 in case study 1 is shown. It showcases the behaviour of MODREC under different energy generation circumstances, in particular two scenarios, where energy generation is high and where it is almost zero throughout the day.

IX. CONCLUSION

In this paper we have proposed a new solution for energy community energy management with a fixed load. Our

method, MODREC, joins strengths of both mathematical deterministic optimization (namely LP) and MADRL, where each member household of the community is managed by a DRL agent, tasked with charging / discharging the home's BESS to reduce the community energy cost. The agents operate in real time, but are pre-trained on historical data, where they are trained to imitate an optimal strategy. This strategy, computed using LP, cannot be applied in real time. We demonstrate the potential of this approach in a data-driven simulation, where we show that savings of up to 29% of the cumulative community energy cost are possible compared to a naive baseline.

In future work, we aim to extend the proposed framework along several directions. First, we will explore alternative reinforcement learning algorithms such as DDPG and PPO, which are well-suited for continuous action spaces and may enhance the adaptability of the control strategy. Second, we plan to integrate short-term forecasting techniques into the decision-making process. In this context, Age of Information (AoI)-based data prioritization [49] offers a promising avenue to improve responsiveness and prediction accuracy by ensuring that the most up-to-date smart meter data informs model updates. In parallel, we intend to extend our evaluation through larger-scale community simulations to assess the scalability and coordination capabilities of the proposed framework under more diverse and realistic operating conditions. Finally, we aim to generalize the proposed approach into a broader methodology for solving online decision-making problems with access to historical data [50]. This involves first solving an offline optimization problem to obtain an expert policy, and then using that policy to train an online learning model capable of making real-time decisions under uncertainty.

REFERENCES

[1] D. Brown, S. Hall, and M. E. Davis, "What is prosumerism for? Exploring the normative dimensions of decentralised energy transitions," *Energy Res. Social Sci.*, vol. 66, Aug. 2020, Art. no. 101475.
[2] I. Walker and A. Hope, "Householders' readiness for demand-side response: A qualitative study of how domestic tasks might be shifted in time," *Energy Buildings*, vol. 215, May 2020, Art. no. 109888.
[3] J. Leitão, P. Gil, B. Ribeiro, and A. Cardoso, "A survey on home energy management," *IEEE Access*, vol. 8, pp. 5699–5722, 2020.
[4] J. Eschmann, *Reward Function Design in Reinforcement Learning*. Cham, Switzerland: Springer, 2021, pp. 25–33, doi: 10.1007/978-3-030-41188-6_3.
[5] European Commission. (Sep. 26, 2025). *Energy Communities*. [Online]. Available: https://energy.ec.europa.eu/topics/markets-and-consumers/energy-consumers-and-prosumers/energy-communitiesen
[6] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–35, Apr. 2017.
[7] N. Pflugradt, P. Stenzel, L. Kotzur, and D. Stolten, "LoadProfileGenerator: An agent-based behavior simulation for generating residential load profiles," *J. Open Source Softw.*, vol. 7, no. 71, p. 3574, Mar. 2022.
[8] T. Huld, R. Müller, and A. Gambardella, "A new solar radiation database for estimating PV performance in Europe and Africa," *Sol. Energy*, vol. 86, no. 6, pp. 1803–1815, Jun. 2012.
[9] X. Xu, Y. Jia, Y. Xu, Z. Xu, S. Chai, and C. S. Lai, "A multi-agent reinforcement learning-based data-driven method for home energy management," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3201–3211, Jul. 2020.
[10] F. Alfaverh, M. Denaï, and Y. Sun, "Demand response strategy based on reinforcement learning and fuzzy reasoning for home energy management," *IEEE Access*, vol. 8, pp. 39310–39321, 2020.
[11] J. Silvente and L. G. Papageorgiou, "An MILP formulation for the optimal management of microgrids with task interruptions," *Appl. Energy*, vol. 206, pp. 1131–1146, Nov. 2017.
[12] T. Yu, D. S. Kim, and S.-Y. Son, "Optimization of scheduling for home appliances in conjunction with renewable and energy storage resources," *Int. J. Smart Home*, vol. 7, no. 4, pp. 261–272, 2013.
[13] K. Ma, S. Hu, J. Yang, X. Xu, and X. Guan, "Appliances scheduling via cooperative multi-swarm PSO under day-ahead prices and photovoltaic generation," *Appl. Soft Comput.*, vol. 62, pp. 504–513, Jan. 2018.
[14] C. Chen, J. Wang, Y. Heo, and S. Kishore, "MPC-based appliance scheduling for residential building energy management controller," *IEEE Trans. Smart Grid*, vol. 4, no. 3, pp. 1401–1410, Sep. 2013.
[15] E. Matallanas, M. Castillo-Cagigal, A. Gutiérrez, F. Monasterio-Huelin, E. Caamaño-Martín, D. Masa, and J. Jiménez-Leube, "Neural network controller for active demand-side management with PV energy in the residential sector," *Appl. Energy*, vol. 91, no. 1, pp. 90–97, Mar. 2012.
[16] M. Pokorn, M. Mohorčič, A. Čampa, and J. Hribar, "Smart home energy cost minimisation using energy trading with deep reinforcement learning," in *Proc. 10th ACM Int. Conf. Syst. Energy-Efficient Buildings, Cities, Transp.*, Nov. 2023, pp. 361–365.
[17] M. Pokorn and J. Hribar, "Smart homes, smarter savings: Energy trading with deep reinforcement learning," in *Proc. IEEE 22nd Medit. Electrotechnical Conf. (MELECON)*, Jun. 2024, pp. 19–24.
[18] B. Mbuwir, F. Ruelens, F. Spiessens, and G. Deconinck, "Battery energy management in a microgrid using batch reinforcement learning," *Energies*, vol. 10, no. 11, p. 1846, Nov. 2017.
[19] M. Hu and F. Xiao, "Price-responsive model-based optimal demand response control of inverter air conditioners using genetic algorithm," *Appl. Energy*, vol. 219, pp. 151–164, Jun. 2018.
[20] Z. Yahia and A. Pradhan, "Optimal load scheduling of household appliances considering consumer preferences: An experimental analysis," *Energy*, vol. 163, pp. 15–26, Nov. 2018.
[21] Z. Yu, L. Jia, M. C. Murphy-Hoye, A. Pratt, and L. Tong, "Modeling and stochastic control for home energy management," *IEEE Trans. Smart Grid*, vol. 4, no. 4, pp. 2244–2255, Dec. 2013.
[22] B.-G. Kim, Y. Zhang, M. van der Schaar, and J.-W. Lee, "Dynamic pricing and energy consumption scheduling with reinforcement learning," *IEEE Trans. Smart Grid*, vol. 7, no. 5, pp. 2187–2198, Sep. 2016.
[23] H. Wang, T. Huang, X. Liao, H. Abu-Rub, and G. Chen, "Reinforcement learning in energy trading game among smart microgrids," *IEEE Trans. Ind. Electron.*, vol. 63, no. 8, pp. 5109–5119, Aug. 2016.
[24] D. Neves and C. A. Silva, "Optimal electricity dispatch on isolated mini-grids using a demand response strategy for thermal storage backup with genetic algorithms," *Energy*, vol. 82, pp. 436–445, Mar. 2015.
[25] X. Luan, J. Wu, S. Ren, and H. Xiang, "Cooperative power consumption in the smart grid based on coalition formation game," in *Proc. 16th Int. Conf. Adv. Commun. Technol.*, Feb. 2014, pp. 640–644.
[26] A. Barbato, A. Capone, G. Carello, M. Delfanti, M. Merlo, and A. Zaminga, "House energy demand optimization in single and multi-user scenarios," in *Proc. IEEE Int. Conf. Smart Grid Commun. (SmartGridComm)*, Oct. 2011, pp. 345–350.
[27] S. Zhou, Z. Hu, W. Gu, M. Jiang, and X.-P. Zhang, "Artificial intelligence based smart energy community management: A reinforcement learning approach," *CSEE J. Power Energy Syst.*, vol. 5, no. 1, pp. 1–10, Mar. 2019.
[28] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Cooperative day-ahead bidding strategies for demand-side expected cost minimization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 5224–5228.
[29] F. Charbonnier, B. Peng, J. Vienne, E. Stai, T. Morstyn, and M. McCulloch, "Centralised rehearsal of decentralised cooperation: Multi-agent reinforcement learning for the scalable coordination of residential energy flexibility," *Appl. Energy*, vol. 377, Jan. 2025, Art. no. 124406.
[30] T. Peirelinck, C. Hermans, F. Spiessens, and G. Deconinck, "Combined peak reduction and self-consumption using proximal policy optimisation," *Energy AI*, vol. 16, May 2024, Art. no. 100323, doi: 10.1016/j.egyai.2023.100323.
[31] H. T. Dinh, K.-H. Lee, and D. Kim, "Supervised-learning-based hour-ahead demand response for a behavior-based home energy management system approximating MILP optimization," *Appl. Energy*, vol. 321, Sep. 2022, Art. no. 119382. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306261922007231

[32] L. Yu, S. Qin, M. Zhang, C. Shen, T. Jiang, and X. Guan, "A review of deep reinforcement learning for smart building energy management," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12046–12063, Aug. 2021.
[33] T. Ahmad, H. Chen, Y. Guo, and J. Wang, "A comprehensive overview on the data driven and large scale based approaches for forecasting of building energy demand: A review," *Energy Buildings*, vol. 165, pp. 301–320, Apr. 2018.
[34] J. K. Strayer, *Linear Programming and Its Applications*. Cham, Switzerland: Springer, 2012.
[35] C. A. Floudas and X. Lin, "Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications," *Ann. Oper. Res.*, vol. 139, no. 1, pp. 131–162, Oct. 2005.
[36] A. Schrijver, *Theory of Linear and Integer Programming*. Hoboken, NJ, USA: Wiley, 1998.
[37] N. Karmarkar, "A new polynomial-time algorithm for linear programming," in *Proc. 16th Annu. ACM Symp. Theory Comput. (STOC)*, 1984, pp. 302–311.
[38] D. Zhang, X. Han, and C. Deng, "Review on the research and practice of deep learning and reinforcement learning in smart grids," *CSEE J. Power Energy Syst.*, vol. 4, no. 3, pp. 362–370, Sep. 2018.
[39] A. Ghadertootoonchi, M. Moeini-Aghtaie, and M. Davoudi, "A hybrid linear programming-reinforcement learning method for optimal energy hub management," *IEEE Trans. Smart Grid*, vol. 14, no. 1, pp. 157–166, Jan. 2023.
[40] C. Deng and K. Wu, "Residential demand response strategy based on deep deterministic policy gradient," *Processes*, vol. 9, no. 4, p. 660, Apr. 2021. [Online]. Available: https://www.mdpi.com/2227-9717/9/4/660
[41] A. Yassine, "Analysis of a cooperative and coalition formation game model among energy consumers in the smart grid," in *Proc. 3rd Int. Conf. Commun. Inf. Technol. (ICCIT)*, Jun. 2013, pp. 152–156.
[42] T. Dengiz and M. Kleinebrahm, "Imitation learning with artificial neural networks for demand response with a heuristic control approach for heat pumps," 2024, *arXiv:2407.11561*.
[43] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*. New York, NY, USA: McGraw-Hill, 2015.
[44] P. Virtanen et al., "SciPy 1.0: Fundamental algorithms for scientific computing in Python," *Nature Methods*, vol. 17, no. 3, pp. 261–272, 2020.
[45] Q. Huangfu and J. A. J. Hall, "Parallelizing the dual revised simplex method," *Math. Program. Comput.*, vol. 10, no. 1, pp. 119–142, Mar. 2018.
[46] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[47] P. Beaudan and S. He, "Applying machine learning to trading strategies: Using logistic regression to build momentum-based trading strategies," *SSRN Electron. J.*, 2019, doi: 10.2139/ssrn.3325656.
[48] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," *J. Artif. Intell. Res.*, vol. 47, pp. 253–279, Jun. 2013.
[49] J. Hribar, C. Fortuna, and M. Mohorčič, "The role of age of information in enhancing short-term energy forecasting," *Energy*, vol. 318, Mar. 2025, Art. no. 134704.
[50] J. Hribar, M. Mohorčič, and A. Čampa, "Improving energy autonomy of positive energy districts using multi-agent deep reinforcement learning," *Sci. Rep.*, vol. 15, no. 1, p. 27798, Jul. 2025.

MATIC POKORN is currently pursuing the master's degree in computer science and mathematics with the University of Ljubljana. Since 2023, he has been with the Jožef Stefan Institute, where he has been working on reinforcement learning research. His recent work focuses on the application of reinforcement learning to home energy management systems. His research interests include mathematical modeling, machine learning, reinforcement learning, and optimization.

ANDREJ ČAMPA received the Ph.D. degree in electrical engineering, specializing in numerical methods and mathematical simulations, from the University of Ljubljana, Ljubljana, Slovenia, in 2009. He has more than ten years of experience working in research and development institutions covering diverse fields from photovoltaics, optoelectronics, and semiconductor materials to numerical modeling and big data analytics, which represents his current expertise domain. His areas of interest at ComSensus and the Jožef Stefan Institute are closely related to understanding and optimization in smart grids.

MIHA SMOLNIKAR received the B.Sc. degree in electrical engineering from the University of Ljubljana, Ljubljana, Slovenia, in 2005. After graduation, he joined the Department of Communication Systems, Jožef Stefan Institute, as a Researcher. In 2011, he co-founded a high-tech company ComSensus, developing digitalization solutions for various industries. He has more than 15 years of experience in embedded systems, wireless communications, and smart grids. He actively participates in EU-funded collaborative projects. He has published several peer-reviewed journal articles and conference papers.

MIHAEL MOHORČIČ (Senior Member, IEEE) is currently the Head of the Department of Communication Systems and a Scientific Counselor at the Jožef Stefan Institute, as well as a Full Professor at the Jožef Stefan International Postgraduate School. His recent work focuses on AI-driven resource management in wireless communications, smart infrastructure connectivity, and intelligent sensing applications. He has contributed to more than 25 international research projects in mobile and satellite communications, UAV communication systems, and wireless sensor networks, along with more than 15 national basic and applied research projects. He has co-authored more than 230 journal articles and conference publications, three books, and one patent. His research spans advanced wireless communication systems, including mobile, satellite, and stratospheric networks; heterogeneous and ad hoc networks; wireless sensor networks; and the Internet of Things. He actively serves on conference organizing committees, including as the General Chair and the TPC Chair.

JERNEJ HRIBAR (Member, IEEE) received the Ph.D. degree in electrical engineering from Trinity College Dublin, Ireland, in 2020. He is currently a Research Fellow with the Jožef Stefan Institute, Ljubljana, Slovenia. He has contributed to several international projects and co-authored multiple journal publications and patents. His research interests include deep reinforcement learning, multi-agent systems, smart infrastructure management, and intelligent networking.