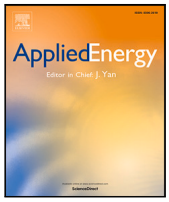
Applied Energy 324 (2022) 119688

Contents lists available at ScienceDirect

Applied Energy

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



Resilience enhancement of multi-agent reinforcement learning-based demand response against adversarial attacks

Lanting Zeng [a], Dawei Qiu [b], Mingyang Sun [a,*]

[a] Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China
[b] Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK

ARTICLE INFO

Keywords:
Demand response
Cyber security
Resilience
Multi-agent reinforcement learning
Adversarial attacks

ABSTRACT

Demand response improves grid security by adjusting the flexibility of consumers meanwhile maintaining their demand–supply balance in real-time. With the large-scale deployment of distributed digital communication technologies and advanced metering infrastructures, data-driven approaches such as multi-agent reinforcement learning (MARL) are being widely employed to solve demand response problems. Nevertheless, the massive interaction of data inside and outside the demand response management system may lead to severe threats from the perspective of cyber-attacks. The cyber security requirements of MARL-based demand response problems are less discussed in the existing studies. To this end, this paper proposes a robust adversarial multi-agent reinforcement learning framework for demand response (RAMARL-DR) with an enhanced resilience against adversarial attacks. In particular, the proposed RAMARL-DR first constructs an adversary agent that aims to cause the worst-case performance via formulating an adversarial attack; and then adopts periodic alternating robust adversarial training scenarios with the optimal adversary aiming to diminish the severe impacts induced by adversarial attacks. Case studies are conducted based on an OpenAI Gym environment CityLearn, which provides a standard evaluation platform of MARL algorithms for demand response problems. Empirical results indicate that the MARL-based demand response management system is vulnerable when the adversary agent occurs, and its performance can be significantly improved after periodic alternating robust adversarial training. It can be found that the adversary agent can result in a 41.43% higher metric value of *Ramping* than the no adversary case, whereas the proposed RAMARL-DR can significantly enhance the system resilience with an approximately 38.85% reduction in the ramping of net demand.

1. Introduction

The power system is undergoing a fundamental revolution from fossil fuels to clean energy for both planning and operation by increasing the penetration of renewable energy resources (RES) [1]. This transition, however, brings out a significant challenge to power system reliability due to the limited-controllable variability and partial predictability of these intermittent RES [2]. To this end, a large scale of distributed energy resources (DER) is being deployed to hedge the uncertain RES and improve the reliability of energy supply [3]. Currently, decentralization and digitalization are supporting the integration of DER into the grid, which may raise another significant challenge of cyber security, creating a negative impact on the normal operation of power systems. For example, a massive cyber-attack occurring on Ukraine's power system in 2015 caused a serious power outage for more than ten thousand households, and facilities [4]. Several works demonstrate the devastating effects of cyber-attacks on power systems. The attacker can manipulate meter readings and launch false data injection attacks against state estimation in electric power grids, further influencing power system control and operation [5,6]. The smart grid data collection infrastructures have been demonstrated to be vulnerable to cyber-attacks [7]. An attack adversary is constructed to impact power grid operation, which builds or rents a botnet of computers and modulates their power consumption [8]. The simulation results estimate that between 2.5 and 9.8 million infections are sufficient to attack the European synchronous grid. Furthermore, Denial-of-Service (DoS) attacks are used to delay, block, or corrupt Smart Grid communication, further impairing the operation of electrical equipment [9–11].

By definition, demand response (DR) technologies are deployed to enable the involvement of end-consumers in the system operation and balance supply and demand at the distribution level by reducing or shifting the energy consumption. Compared with the energy sectors in generation and transmission levels, DR besides end-consumers are

* Corresponding author.
E-mail address: mingyangsun@zju.edu.cn (M. Sun).

https://doi.org/10.1016/j.apenergy.2022.119688
Received 31 March 2022; Received in revised form 7 June 2022; Accepted 12 July 2022
Available online 4 August 2022

more likely to be attacked since DR is envisioned to fully integrate high-speed and two-way communication technologies into millions of power equipment for energy management capabilities [12]. More specifically, in the electricity retail market, such communication can be achieved via Advanced Metering Infrastructures (AMI), e.g., smart meters [13]. However, the large-scale deployment of AMI encourages the use of marginally cheaper hardware which results in limited computational resources to support advanced security functions when an attack exists. For example, NESCOR lists the cyber failure scenarios that might occur in the advanced metering infrastructure (AMI), distributed energy resources (DERs), wide-area monitoring, protection, and control (WAMPAC) systems [14]. Therefore, enhancing the resilience against cyber-attacks under the DR concept becomes increasingly important to maintain a reliable and secure energy system.

Regarding the DR modeling approach, in the literature, the model-based optimization methods are being assumed as the conventional problem-solving approaches and have been successfully applied to solve various DR problems [15]. In [16], mixed-integer linear programming (MILP) is proposed to develop an optimal production scheduling framework for an industrial DR management system with the objective of saving operation costs and reducing demand peaks. In [17], stochastic programming (SP) is proposed for an energy hub scheduling problem, accounting for the DR program and the uncertainties of wind and PV generation. In [18], a model predictive controllers (MPC) approach is proposed for a building heating system under the DR program. Nevertheless, the conventional optimization methods require formulating the energy model accurately and acquiring all the technical parameters of the system components, which are normally impractical considering the privacy concern and the system aging. Furthermore, a series of issues associated with sensor noise, RES variation, equipment faults, and consumed time prevent such methods from deploying real-time control solutions since the perfect information or accurate forecasts of uncertainty parameters are assumed to be acquired.

On the other hand, as a model-free and real-time automatic control method, reinforcement learning (RL) [19] has recently brought increased attention to the DR problems [20]. More specifically, RL is proposed to study the sequential and dynamic decision-making problem of an agent that can gradually learn the optimal control decisions by utilizing experiences acquired from its repeated interactions with the environment without prior knowledge. As such, the exact DR management system and technical parameters are unnecessary. The system's stochastic and dynamic characteristics can also be learned through the vast interactions with the environment in the big-data era. In general, RL can be classified into two categories: single-agent reinforcement learning (SARL) and multi-agent reinforcement learning (MARL) methods [21]. The first category SARL method employs the decision-making process of a single entity, e.g., household, micro-grid, and energy hub. In [22], the authors propose a novel incentive-based DR algorithm. Based on the predicted data, it adopts Q-learning (QL) to derive the optimal incentive rates for different customers considering the benefits of both service providers and customers. In [23], a dueling deep Q network (DDQN) method is proposed to develop the DR management system of interruptible. Under the premise of ensuring power supply for customers, this method reduces the total daily cost of distribution system operators by 16.9% compared to the program without DR. In [24], a deep deterministic policy gradient (DDPG) method is proposed to generate the optimal control strategy for a multi-zone residential HVAC system with the goal of minimizing energy consumption costs while maintaining the users' comfort. To further avoid the physical constraint violation, the authors in [25] propose a safe-DDPG method to optimize the DR problem of an energy hub that can ensure the demand–supply balance without requiring any model knowledge. The second category, the MARL method, is also widely studied to be implemented into DR programs for multiple agents. In [26], a multi-agent QL method is proposed to solve the DR problem for a home energy management system, where each energy device equips with a Q-table to optimize its energy schedules. In [27], a multi-agent deep deterministic policy gradient (MADDPG) is proposed to optimize the optimal schedule for different machine agents in a discrete manufacturing systems energy management. In [28], an improved multi-agent QL method based on a deep neural network (DNN) is proposed to optimize the thermostatically controlled loads for 50 houses, resulting in energy savings of almost 200 kWh per household. In [21], a parameter-sharing (PS) framework based on the MADDPG method is proposed for 300 residential households to solve the DR problem and P2P energy trading problem, which can improve the scalability and persevere the privacy of households.

Recently, researchers have noticed that RL models are vulnerable to adversarial attacks [29], such as Atari games, autonomous driving, robot control tasks, and power systems control. The attacker influences the RL decisions by injecting elaborate perturbations into observations, further destroying the RL-based system operation. In [30], the authors propose a criticality-based adversarial perturbation into the agents' observation and implement such a perturbation into a DQN method. Case studies based on the IEEE 14-bus system and the IEEE 118-bus system show the reward degradation of about 32.1% and 8.9% under adversarial attacks. An adversarial agent is used for cooperative distributed MARL methods to alter other agents and prevent algorithm convergence [31]. Similarly, in [32], the authors develop an adversarial RL agent for cooperative MARL methods to estimate adversarial actions and use gradient-based methods to generate perturbations by adversarial actions. It results in a sharp reduction of the team's winning rate in StarCraft games. External adversarial attacks may cause severe damage to RL-based control systems, especially in safety-critical tasks, such as energy systems. To deploy MARL algorithms to the DR programs, there is an urgent need to enhance the resilience against adversarial attacks.

The RL model lacks decision experience about adversarial examples during regular training, which leads to the control policy being sensitive to the perturbation or fooling by adversarial examples. Thus, several works focus on improving the robustness of the RL models by robust adversarial training. Kos and Song [33] first used adversarial training for robustifying SARL algorithms, which generated perturbation by the Fast Gradient Sign Method (FGSM) [34] and random noise. Furthermore, several studies [35–37] show that the SARL models achieve robust performance after adversarial training with adversarial attacks. Many researchers attempt to design stronger attacks for adversarial training and propose methods to learn an adversary online with the SARL agent. Specifically, the adversarial attack is identified as an adversary agent and participates in the training process of the agent. Pinto et al. [38] construct an adversary to generate perturbations in the environment and propose a robust adversarial reinforcement learning (RARL) framework, which formulates the policy learning as a zero-sum, minimax objective function. It takes a jointly training way to improve the performance of both the adversary and the agent. Zhang et al. [39] propose the alternating training with learned adversaries (ATLA) framework, training an adversary online with the agent. ATLA generates stronger attacks than other methods and significantly improves the agent's performance against tested attacks. Robust MARL methods have also used minimax optimization and equilibrium while considering the opponent and the agent. The robust Markov Game is designed in [40] to handle uncertainties from the reward or transition probability model. To cope with the misbehavior of attacked agents in the MARL system, Li et al. [41] propose robust multi-agent Q-learning (RoM-Q) utilizing minimax optimization in policy updates. However, the security of MARL algorithms under cyber-attacks and robustness improvement by adversarial training have not been fully explored, especially in the DR management system.

To this end, this paper proposes a novel robust adversarial training framework for MARL-based DR models, RAMARL-DR, aiming to improve the resilience of the MARL-based DR management system against adversarial attacks. The RAMARL-DR framework models the external adversarial attacks as a single adversarial RL agent, who

learns an optimal policy to generate observation perturbations, causing the worst-case performance of the controller directly. Then, it utilizes robust adversarial training to improve the performance of the MARL-based DR model against this adversary agent. In particular, this training method is formulated as a robust Markov Game with minimax optimization, which is solved by alternating robust adversarial training inspired by the RARL [38] and ATLA [39] frameworks. The simulation results based on an energy system containing a nine-buildings group demonstrate that the optimal adversary agent helps the MARL-based DR management system learn the experience of tackling adversarial attacks and significantly improve its resilience. In contrast, the adversarial training with a random adversary reduces the controller performance. In addition, we found that the periodic robust adversarial training scenario addresses the problem of robust MARL model convergence difficulty in the episode-by-episode training scenario. In summary, The main contributions of this paper are as follows:

(1) To the best of the authors' knowledge, this is the first work that proposes a resilience enhanced MARL-based DR methodological framework, RAMARL-DR, against the potential adversarial attacks in the cyber space. In particular, this develops a novel robust adversarial training framework, RAMARL, which can mathematically formulate the adversarial Markov Game and improves the MARL models' performance by robust adversarial training. More specifically, it models the adversarial attacks as an optimal adversary agent learned by the SARL algorithm considering the perturbation bound.
(2) In contrast to regular adversarial training, this paper proposes the periodic robust adversarial training scenario, which provides a stable convergence policy rather than instability in the episode-by-episode training method.
(3) Furthermore, case studies are conducted based on an open-source platform, CityLearn, with real-world energy data to demonstrate the superior performance of the proposed method under various adversarial attacks. In particular, it can be found that the adversary agent can result in a 41.43% higher metric value of *Ramping* than the no adversary case, whereas the proposed RAMARL-DR can significantly enhance the system resilience with an approximately 38.85% reduction in the ramping of net demand.
(4) In addition, we investigate the impacts of pre-training stage with the following finding: In the fixed adversary training scenario, the pre-training helps the control policy take initial exploration policy, avoiding stuck in a lousy policy space. However, it has little impact on the alternating adversary training scenario.

The remainder of this paper is organized as follows. Section 2 describes a detailed demand response management system model and reformulates it as a MARL-based control problem mathematically. Section 3 offers the RAMARL-DR framework with three training algorithms. Section 4 provides performance evaluation metrics. In Section 5, we present the open-source demand response simulation environment based on OpenAI Gym, algorithm settings, experiment settings and discuss evaluation performances. Finally, conclusions and future work are given in Section 6.

2. Problem formulation

2.1. Mathematical models of demand response

This work studies a DR management system based on a microgrid containing N buildings. In particular, each building consists of (1) two kinds of inflexible demand: electric demand, thermal (heating & cooling) demand; (2) one renewable-based generator: solar photovoltaic (PV); (3) two kinds of storage systems: electric energy storage and thermal energy storage; (4) two kinds of energy converters: heat pump and electric heater. Of which its model structure is illustrated in Fig. 1. More specifically, the mathematical equations of the controllable components (i.e., two energy converters and two storage systems) within the building are presented as follows:

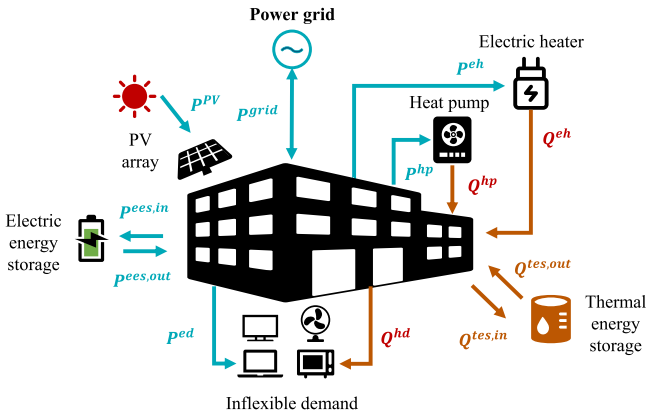


Fig. 1. Energy models of buildings.

- Heat pump: it takes electric power from the power grid P^{hp}, and supplies thermal demand Q^{hp}, to the building heating or cooling and energy storage devices, following the equation,

$$P_t^{hp} = \frac{Q_t^{hp}}{\eta_t^{hp}}, \forall t \in T \tag{1}$$

$$0 \le Q_t^{hp} \le \overline{Q}^{hp}, \forall t \in T \tag{2}$$

where η_t^{hp} in Eq. (1) is the energy conversion coefficient related to indoor target temperatures, outdoor air temperatures, and the technical efficiency coefficient η_{tech}^{hp}; $\overline{Q}_t^{hp}$ in constraint (2) represents the output power capacity of heat pump.

- Electric heater: it takes electric power from the power grid P^{eh} to provide domestic hot water (DHW) energy Q^{eh} for the building or storage devices, following

$$P_t^{eh} = \frac{Q_t^{eh}}{\eta^{eh}}, \forall t \in T \tag{3}$$

$$0 \le Q_t^{eh} \le \overline{Q}^{eh}, \forall t \in T \tag{4}$$

where η^{eh} in Eq. (3) is the heater efficiency of energy conversion from electricity to heat power; $\overline{Q}_t^{eh}$ in constraint (4) represents the output power capacity of electric heater.

- Electric energy storage: the charging and discharging power in the electric energy storage can be denoted as $P^{ees,in}$ and $P^{ees,out}$, respectively. Of which, constraints (5) and (6) restrict the maximum charging and discharging power, where the binary variable $u^{ees} \in \{0,1\}$ is corresponding to the charging ($u^{ees} = 1$) and discharging ($u^{ees} = 0$) status of electric energy storage, since the battery cannot behave charging and discharging simultaneously.

$$0 \le P_t^{ees,in} \le u_t^{ees} \overline{P}^{ees}, \forall t \in T \tag{5}$$

$$(u_t^{ees} - 1)\overline{P}^{ees} \le P_t^{ees,out} \le 0, \forall t \in T \tag{6}$$

where $\overline{P}^{ees}$ indicates the maximum charging/discharging limit of battery. Given the charging power $P_t^{ees,in}$ and discharge power $P_t^{ees,out}$, the SoC of the battery is associated with the energy charging/discharging efficiency η^{ees} can be calculated as:

$$SoC_{t+1}^{ees} = SoC_t^{ees} + (P_t^{ees,in}\eta^{ees}\Delta t)/\overline{S}^{ees} + (P_t^{ees,out}\Delta t)/(\eta^{ees}\overline{S}^{ees}), \forall t \in T \tag{7}$$

Finally, the battery SoC is also limited by its lower and upper limits, which can be expressed as:

$$\underline{S}^{ees} \le S_t^{ees} \le \overline{S}^{ees}, \forall t \in T \tag{8}$$

- Thermal energy storage: it allows to store energy that can release into the building at the appropriate time. Devices include chilled water and DHW tanks, which receive cooling, heating, and DHW energy from heat pumps and electric heaters. Similar to the electric storage system in (5)–(8), the operation model of thermal energy storage can be formulated as:

$$0 \le Q_t^{tes,in} \le u_t^{tes}\overline{Q}^{tes}, \forall t \in T \tag{9}$$

$$(u_t^{tes} - 1)\overline{Q}^{tes} \le Q_t^{tes,out} \le 0, \forall t \in T \tag{10}$$

$$SoC_{t+1}^{tes} = SoC_t^{tes} + (Q_t^{tes,in}\eta^{tes}\Delta t)/\overline{S}^{tes} + (Q_t^{tes,out}\Delta t)/(\eta^{tes}\overline{S}^{tes}), \forall t \in T \tag{11}$$

$$\underline{S}^{tes} \le S_t^{tes} \le \overline{S}^{tes}, \forall t \in T \tag{12}$$

where $Q_t^{tes,in}$ and $Q_t^{tes,out}$ represent the battery charging and discharging power, respectively; $\overline{Q}^{tes}$ indicates the battery power capacity; η^{tes} is the energy efficiency caused by charging and discharging behaviors. Finally, S_t^{tes} represents the battery SoC in the thermal energy system.

Given the demand of electric P_t^{ed} and thermal Q_t^{hd}, production, storage, and conversion, the demand–supply balances of electricity and heat sectors of the building should always be satisfied at each time step, which can be respectively expressed as below:

$$P_t^{ed} - P_t^{pv} + P_t^{ees,in} + P_t^{ees,out} + P_t^{hp} + P_t^{eh} = P_t^{grid}, \forall t \in T \tag{13}$$

$$Q_t^{hd} - Q_t^{hp} - Q_t^{eh} + Q_t^{tes,in} + Q_t^{tes,out} = 0, \forall t \in T \tag{14}$$

where P_t^{grid} represents the net demand (positive) or generation (negative) bought or sold in the power grid. The objective of each building is maintaining the demand–supply balances of both electricity and heat sectors in real-time, leading to a reliable building demand response management system, which aims to flat the demand curve and reduce the peak demand of the micro-grid. However, solving the above demand–supply balance problem in a building system faces several challenges:

(1) It might be impractical to acquire the mathematical models and the technical parameters of all building components explicitly, and the optimization model thus cannot be constructed.

(2) Solving a comprehensive optimization problem including both power and heat sectors in a time-coupling fashion is time-consuming, especially when accounting for the highly dynamic and stochastic characteristics of adversarial attacks.

(3) Conventional mathematical methods do not generalize to the system dynamics since the optimal decisions of DR need to be re-optimized for any new condition.

2.2. Reformulation as a partially observable stochastic game

In order to address the above three challenges raised in Section 2.1, the building DR problem can be reformulated as a partially observable stochastic game (POSG) [42], as evident from Fig. 2, where each building DR controller is defined as the agent and its utilized DR management system is assumed the environment. In this case, the agent does not require any knowledge from the DR management system but learns the optimal control policy together with the attack characteristics by repeatably interacting with the environment. In addition, once the control policy is well trained, it can be directly deployed to the practical scenario for test in milliseconds.

A POSG system can be formulated as a tuple, $\langle N, S, \{A^i\}_{i\in N}, P, \{R^i\}_{i\in N}, \gamma, \{O^i\}_{i\in N}, Z\rangle$: where N is the number of agents; S is the environmental state space; A^i is the action space of agent i and $\{A^i\}_{i\in N}$ is the joint action space, denoting as $\boldsymbol{A} := A^1 \times \cdots \times A^N$; $P : S \times \boldsymbol{A} \times S \to \Delta(S)$ is the state transition probability from state s_t to next state s_{t+1} for given joint actions $\boldsymbol{a_t} = \{a_t^i\}_{i\in N} \in \boldsymbol{A}$; $R^i : S \times \boldsymbol{A} \times S \to \mathbb{R}$ is the immediate reward function of agent i for a transition from $(s_t, \boldsymbol{a_t})$ to s_{t+1}; γ is the discounting rate; O^i is the observation space for agent i and the joint observation space is $\{O^i\}_{i\in N}$, denoting as $\boldsymbol{O} = O^1 \times \cdots \times O^N$; $Z : S \times \boldsymbol{A} \to \Delta(\boldsymbol{O})$ is the probability of observing $\boldsymbol{o_t} \in \boldsymbol{O}$ at any given actions $\boldsymbol{a_t}$ and the new states s_t. At time step t, each agent i chooses an action a_t^i according to the policy $\pi(a_t^i|o_t^i)$ based on its local observation o_t^i. The environment then moves into the next state according to the state transition function P. Each agent i obtains a reward r_t^i and a new local observation o_{t+1}^i. Such process continues and then emits a trajectory of observations, actions, and rewards for each agent i: $\tau_i = o_1^i, a_1^i, r_1^i, o_2^i, \ldots, r_t^i$ over $S \times A_i \times S \to \mathbb{R}$. The goal of each agent i is to find a policy π^i that maximizes its own cumulative discounted reward as indicated in Eq. (15), where $-i$ is the indices of all agents in N except agent i.

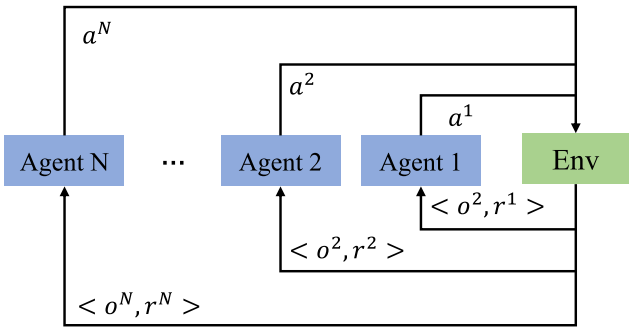


Fig. 2. The schematic diagram for a POSG.

$$J = \sum_{i=1}^{N} \mathbb{E}_{s_{t+1}\sim P(\cdot|s_t,\boldsymbol{a_t}),a^{-i}\sim\pi^{-i}(\cdot|o_t^{-i})} \times \left[\sum_{t\ge 0} \gamma^t R^i\left(s_t, \boldsymbol{a}_t, s_{t+1}\right) \mid a_t^i \sim \pi^i\left(\cdot \mid o_t^i\right), s_0\right] \tag{15}$$

In this multi-agent coordination building DR management system, each agent controls a building and decides how much energy is stored or released in the separate building at any given time. Furthermore, the definitions of the observation space, action space, and rewards function for each agent i are followed.

- observation: o^i is a 19-dimensional vector that includes hour, day, month, outdoor temperature, predicted outdoor temperature, outdoor relative humidity, predicted outdoor relative humidity, indoor temperature, indoor relative humidity, non-shiftable load, cooling storage, heating storage, DHW storage, electric energy storage, direct solar radiation, predicted direct solar radiation, diffuse solar radiation, predicted diffuse solar radiation, and solar generation. The agent's input is the combination of its building observation and other agents' decision information.
- action: a^i is a 2-dimensional vector that indicates the (charging and discharging) power rates of electrical energy storage and thermal energy storage, respectively.
- reward function: r^i evaluates the performance of agents by the demand curve of the district within a simulation period. The reward function of each agent can be the same or different. We define the reward function of each agent, presented as:

$$r^i = -\operatorname{sign}\left(P_i^{grid}\right) \cdot \left(P_i^{grid}\right)^2 \cdot \max\left\{0, \sum_{i=1}^{N} P_i^{grid}\right\} \tag{16}$$

where rewards value depends on the net electricity consumption of building P_i^{grid} and the total net electricity consumption of the entire building group $\sum_{i=1}^{N} P_i^{grid}$.

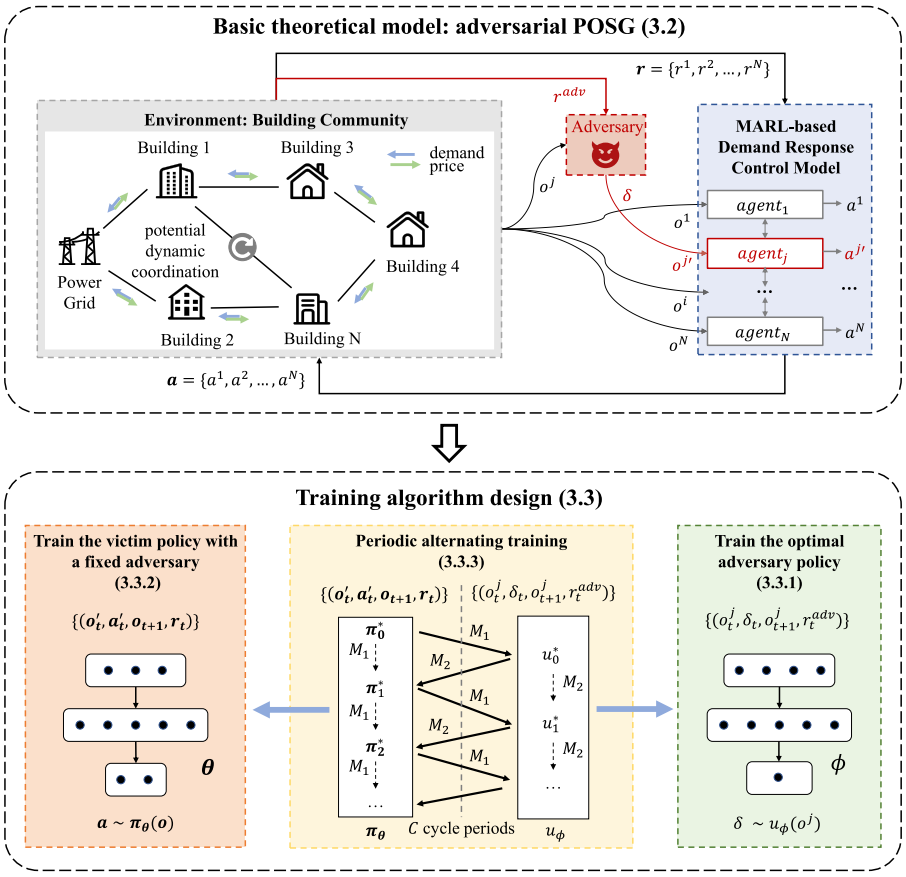


Fig. 3. The overall framework of proposed RAMARL-DR.

Let denote $\boldsymbol{o_t} = \{o_t^1, o_t^2, \ldots, o_t^N\}$, $\boldsymbol{a_t} = \{a_t{}^1, a_t{}^2, \ldots, a_t^N\}$ and $\boldsymbol{r_t} = \{r_t^1, r_t^2, \ldots, r_t^N\}$. Multiple agents observe observation of each building $o_t^i, i \in N$ and operate decisions simultaneously at current state s_t. After making decisions by the controller, it performs joint actions a_t and the environment transits to next state s_{t+1}. Then, each agent receives its immediate reward r_t^i. The control policy of agent i is $\pi^i = \pi_{\theta^i}, i \in N$, where θ^i represents the parameters of the agent policy. In cooperative settings, the goal of the MARL-based building DR management system is to maximize the cumulative total team reward J by optimizing parameters of agents policies $\boldsymbol{\theta} = \{\theta^1, \theta^2, \ldots, \theta^N\}$, where J is denoted by Eq. (15).

$$\max_{\theta} \boldsymbol{J}(\boldsymbol{\theta}) \tag{17}$$

3. Proposed methodology

3.1. The overall framework of RAMARL-DR

For the considered environment, the DR management system in a micro-grid carries out coordination tasks, where each building takes joint actions not only considering its own energy consumption but also the district consumption. In turn, the executed joint action will affect the DR management system as well as other buildings. As discussed before, the RL algorithm is found that it is vulnerable to attacks, especially MARL models. For deploying MARL into DR systems in practicality, the robust adversarial multi-agent reinforcement learning-based DR (RAMARL-DR) is proposed to explore its vulnerabilities and enhance its resilience against attacks. The overall methodological framework of is RAMARL-DR shown in Fig. 3, which is composed of two main parts: (1) the basic theoretical model: adversarial POSG; and (2) the training algorithm design. The first part formulates the adversarial Markov Game model and describes the decision process of the MARL-based DR system with attacks. Based on the proposed adversarial POSG model, three training algorithms for two kinds of RL agents are designed to enhance the robustness of the MARL-based DR system under attack.

First, we propose the basic theoretical model, adversarial POSG, for the DR management system. Attack is formulated as an adversary agent, who generates adversarial attacks to perturb the observations of a single building agent with the target of affecting all agents and thus, exploring the vulnerabilities of the MARL-based DR management system. In this work, the adversary is modeled by a single-agent RL algorithm, participating in training as a part of the micro-grid. The optimal adversary induces the DR controller to make incorrect decisions (e.g., increasing electricity consumption at peak times), which attempts to result in the worst-case performance of the DR management system.

On this basis, the objective of the next part is to design the training algorithms to enhance the resilience of the MARL-based DR management system under the optimal adversary by robust adversarial training. The robust adversarial training helps the DR management system deal with adversarial attacks considering the history experience [33]. For convenience, the MARL-based building DR management system is denoted as the victim system, and its policy is victim policy. This POSG with adversary participation is defined as adversarial POSG, where the adversary generates perturbation δ to alter observations of agent j. To solve this problem, three algorithms are developed in the RAMARL-DR framework, including (1) training the optimal adversary policy; (2) training the victim policy with a fixed adversary; and (3) periodic alternating training, which will be illustrated in the following subsections, respectively.

3.2. Basic theoretical model: Adversarial POSG

Given the victim policy $\boldsymbol{\pi_\theta} = \{\pi_{\theta^1}, \pi_{\theta^2}, \ldots, \pi_{\theta^N}\}$, a single adversarial policy u_ϕ is designed to model adversarial attacks and threaten one

of agents, where ϕ represents the parameters of the adversarial agent policy. Specifically, the adversary crafts the bounded perturbation to observations of the victim agent, $\delta_t \sim u_\phi(o_t^j) \in B(o^j)$, where δ_t is the corresponding adversary action and $B(o^j)$ is the bound limit of the perturbation. Then, the adversarial inputs of the agent j is $o_t^{j\prime} = o_t^j + \delta_t$. The victim policy makes decision based on disturbed observations, $\boldsymbol{a}_{t'} \sim \boldsymbol{\pi}_{\boldsymbol{\theta}}\left(o_t^{j\prime}, \{o_t^i\}_{i\in N, i\neq j}\right)$. If the adversarial perturbation is within physical constraints from physical characteristics and magnitude bounds, such as stably increasing inflexible energy demands and energy storage in capacity, it cannot be detected by a defense mechanism. Hence, the adversarial perturbation can be bounded to $B(o^j)$ to find the vulnerabilities of the MARL-based DR management system.

In this decision-making scenario, the environment states transitions result from joint actions by the victim and adversary policy. Consequently, the adversarial POSG system can be mathematically represented as:

$$\langle N, S, A^{adv}, \{A^i\}_{i\in N}, P, \{R^i\}_{i\in N}, R^{adv}, \gamma, \{O^i\}_{i\in N}, Z\rangle \tag{18}$$

where N is the number of victim agents; S is the environmental state space; A^{adv} and R^{adv} are the action space and reward function of the adversary, respectively; A^i is the action space of the victim agent i and $\{A^i\}_{i\in N}$ is the joint action space, denoting as $\boldsymbol{A} := A^1 \times \cdots \times A^N$; $P : S \times A^{adv} \times \boldsymbol{A} \times S \to \Delta(S)$ is the state transition probability from state s_t to next state s_{t+1} for each time step t, given an adversarial attacks A^{adv} and joint actions $\boldsymbol{a_t} = \{a_t^i\}_{i\in N} \in \boldsymbol{A}$; $R^i : S \times A^{adv} \times \boldsymbol{A} \times S \to \mathbb{R}$ is the immediate reward function of agent i for a transition from $(s_t, \boldsymbol{a_t})$ to s_{t+1}; γ is the discounting rate; O^i is the observation space for agent i and the joint observation space is $\{O^i\}_{i\in N}$, denoting as $\boldsymbol{O} = O^1 \times \cdots \times O^N$; $Z : S \times \boldsymbol{A} \to \Delta(\boldsymbol{O})$ is the probability of observing $\boldsymbol{o_t} \in \boldsymbol{O}$ at any given actions $\boldsymbol{a_t}$ and the new states s_t. Notice that N, S, $\{A^i\}_{i\in N}$, γ, $\{O^i\}_{i\in N}$ and Z are same with definition in POSG model, while P and R^i are altered with the impacts of A^{adv}.

In our proposed RAMARL-DR framework, an adversary policy against the observations of agent j adopts the single-agent SAC algorithm. It takes deterministic policy, which mapping the observation of agent j to the disturbed observation, $\delta = u_\phi(o^j)$, $o^{j\prime} = o^j + \delta$. To satisfy the bound constraint $B(o^j)$, the output of the neural network at the last layer in the soft actor–critic (SAC) model is a continuous value between 0 and 1, $f'(o^j) \in (0, 1)$. Then, the output is mapped to the bound constraint to get the final perturbation output, $u_\phi(o^j) = f'(o^j) \times B(o^j)$. The adversarial reward function is the opposite value of the team reward:

$$r_{adv} = -\sum_i^N r^i = \sum_i^N \left\{ \operatorname{sign}\left(P_i^{grid}\right) \cdot \left(P_i^{grid}\right)^2 \cdot \max\left\{0, \sum_{i=1}^N P_i^{grid}\right\}\right\} \tag{19}$$

For the adversary, it trains to learn an optimal policy aiming to destroy the district demand balance by minimizing the total team reward J.

$$\min_\phi J(\boldsymbol{\theta}, \phi) \tag{20}$$

However, the robust victim policy improve its resilience against adversary by maximizing J under the optimal adversary.

$$\max_{\boldsymbol{\theta}} \min_\phi J(\boldsymbol{\theta}, \phi) \tag{21}$$

As illustrated in Eq. (21), the objective function of this adversarial POSG game is a minimax function about the parameters of the victim policy and the adversary policy. This is a minimax optimization problem, aiming to find the Nash equilibria policies θ^* and ϕ^* that realize:

$$J^* = \max_{\boldsymbol{\theta}} \min_\phi J(\boldsymbol{\theta}, \phi) = \min_\phi \max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}, \phi) \tag{22}$$

3.3. Training algorithms design

To realize the resilience enhancement based on the above defined adversarial POSG, the proposed RAMARL-DR method optimizes both the adversary and the victim via the following three main algorithms: (1) train an optimal deterministic adversarial policy to generate bounded disturbance; (2) improve the robustness of the victim policy under the optimal adversary by adversarial training; (3) develop the periodic alternating training strategy to enhance the resilience of MARL-based DR management system.

3.3.1. Train the optimal adversary policy

Given a fixed victim policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$, we train an optimal adversary policy u_ϕ according to Eq. (20). Setting agent j as the perturbation target agent, the detailed training process is presented in Algorithm 1. In each of the M episodes, we reset the micro-grid environment with the same initial state. At every time step of episodes, the deterministic adversarial policy u_ϕ generates perturbation δ directly, $\delta_t = u_\phi(o_t^j), o_t^{j\prime} = \delta_t + o_t^j$. Then, the victim policy adjust energy storage responding to perturbed observations $\boldsymbol{o}_{t'}$, $\boldsymbol{a}_{t'} = \boldsymbol{\pi}_{\boldsymbol{\theta}}(\boldsymbol{o}'_t)$. After performing actions, the environment returns a reward r_t^{adv} to the adversarial agent. During training, the replay buffer D_u stores trajectories $\left\{(o_t^j, \delta_t, o_{t+1}^j, r_t^{adv})\right\}$. The parameters of the adversary policy are optimized using a policy optimizer.

Algorithm 1 Training the Optimal Adversary Policy

Input: Environment $\mathcal{E}$, number of episodes M, number of time steps of each episode K and parameters $\boldsymbol{\theta}$ of the victim policy $\boldsymbol{\pi}$.
Output: Parameters ϕ of the adversary policy u.
Initialization: Parameters ϕ of the adversary policy u.

for episode = 1 to M **do**
  Initialize micro-grid environment $\mathcal{E}$ and get observations of each building o_0^i, as the input of each agent i
  while step < K **do**
    Run u_ϕ to generate the adversarial example, $\delta_t = u_\phi(o_t^j)$, $o_t^{j\prime} = \delta_t + o_t^j$
    Agents receive observations $\boldsymbol{o}'_t = \{o_t^1, \ldots, o_t^{j\prime}, \ldots, o_t^N\}$
    Run $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ to get control actions $\boldsymbol{a}'_t = \{\pi_{\theta^1}(o_t^1), \ldots, \pi_{\theta^j}(o_t^{j\prime}), \ldots, \pi_{\theta^N}(o_t^N)\}$
    Perform actions $\boldsymbol{a}'_t$ and get next state s_{t+1}
    Receive adversarial rewards r_t^{adv}
    Store trajectories $\{(o_t^j, \delta_t, o_{t+1}^j, r_t^{adv})\}$ in replay buffer D_u for the adversary policy u_ϕ
    Update the parameters of the adversary policy, $\phi \leftarrow$ Policy Optimizer (D_u, ϕ)
  end while
end for

3.3.2. Train the victim policy with a fixed optimal adversary

Given the victim policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is pre-trained and fixed, the adversary policy u_ϕ^* can be learned to obtain a worst-case performance under bounded perturbations. In this context, enhance the worst-case performances of the victim policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ under the optimal adversary policy u_ϕ^* can also improve the robustness of the MARL-based DR management system. To this end, we optimize the victim policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ by robust adversarial training with the fixed adversary policy u_ϕ^*. The robust adversarial training algorithm is outlined in Algorithm 2. The parameters updating of $\boldsymbol{\theta}$ is the respective optimization procedure of $\{\theta^1, \theta^2, \ldots, \theta^N\}$ with their own replay buffer $D_{\pi^i} = \left\{(o_t^i, a_t^i, o_{t+1}^i, r_t^i)\right\}$.

Note that $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ has been preliminarily trained without an adversary, and the robust adversarial training further optimizes its parameters with the optimal adversary u_ϕ^* trained in Section 3.3.1. The pre-training for $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ without adversarial attacks provides a good initial exploration strategy. The optimal adversary is trained for the specific victim policy may have no attack effect on the randomly initialized victim policy. Meanwhile, if the victim policy takes adversarial training first, it will

focus on defending against adversarial attacks while not on the performance under normal states. Thus, the pre-training is necessary in the adversarial training taking fixed adversary, which has been verified in the experiments.

Algorithm 2 Training the Victim Policy with the Optimal Adversary Policy

Input: Environment $\mathcal{E}$, number of episodes M, number of time steps of each episode K, parameters ϕ^* of the adversary policy u^* and preliminarily trained parameters θ of the victim policy $\boldsymbol{\pi}$.
Output: Parameters θ of the victim policy $\boldsymbol{\pi}$.

for episode = 1 to M **do**
  Initialize micro-grid environment $\mathcal{E}$ and get observations of each building o_0^i, as the input of each agent i
  while step < K **do**
    Run u_ϕ^* to generate the adversarial example, $\delta_t = u_\phi^*(o_t^j)$, $o_t^{j\prime} = \delta_t + o_t^j$
    Agents receive observations $\boldsymbol{o}_t' = \{o_t^1, \ldots, o_t^{j\prime}, \ldots, o_t^N\}$
    Run $\boldsymbol{\pi}_\theta$ to get control actions $\boldsymbol{a}_t' = \{\pi_{\theta^1}(o_t^1), \ldots, \pi_{\theta^j}(o_t^{j\prime}), \ldots, \pi_{\theta^N}(o_t^N)\}$
    Perform actions $\boldsymbol{a}_t'$ and get next state s_{t+1}
    Receive each agents rewards $\boldsymbol{r}_t = \{r^i(s_t, \boldsymbol{a}_t', s_{t+1})\}_{i\in N}$
    Store trajectories $\{(\boldsymbol{o}_t', \boldsymbol{a}_t', \boldsymbol{o}_{t+1}, \boldsymbol{r}_t)\}$ in replay buffer $\boldsymbol{D}_\pi$ for $\boldsymbol{\pi}_\theta$
    Update the parameters of the victim policy, $\theta \leftarrow$ Policy Optimizer $(\boldsymbol{D}_\pi, \boldsymbol{\theta})$
  end while
end for

3.3.3. Periodic alternating robust adversarial training

Although the above algorithms can improve the robustness of MARL-based DR to some extent, it is imperative to note that the above-fixed training approaches cannot make a guarantee to achieve the Nash equilibria polices θ^* and ϕ^* to realize J^*. Inspired by the RARL [38] and ATLA frameworks [39], we solve the minimax optimization (22) by periodic alternating robust adversarial training.

When the victim policy $\boldsymbol{\pi}_\theta$ is fixed, an optimal adversary u_ϕ^* against it is obtained by solving the optimization problem:

$$\min_\phi J(\theta, \phi) \tag{23}$$

Then, the optimal adversary policy u_ϕ^* is utilized to improve the robustness of victim policy $\boldsymbol{\pi}_\theta$ by adversarial training. With the fixed adversary policy u_ϕ^*, the objective of the victim policy is identified by:

$$\max_\theta J(\theta, \phi^*) \tag{24}$$

The previous victim policy, the after-training victim policy, and the optimal adversary policy are denoted as $\boldsymbol{\pi_0}^*$, $\boldsymbol{\pi_1}^*$, u_0^* respectively. Aiming to seek a more optimal adversary policy u_1^* against $\boldsymbol{\pi_1}^*$, it can be trained by solving the problem (23). Next, a more robust victim policy can be learned with a stronger adversary u_1^*. The alternating training process is detailed in Eqs. (25). It is intuitively that alternating training between the adversary policy u and the victim policy $\boldsymbol{\pi}$ will converge to the equilibrium point J^* where their policies are optimal.

$$\begin{aligned} \phi_0^* &= \min_\phi J(\boldsymbol{\theta_0}, \phi) \\ \boldsymbol{\theta_1}^* &= \max_\theta J(\theta, \phi_0^*) \\ \phi_1^* &= \min_\phi J(\boldsymbol{\theta_1}^*, \phi) \\ \boldsymbol{\theta_2}^* &= \max_\theta J(\theta, \phi_1^*) \\ &\cdots \end{aligned} \tag{25}$$

Specifically, this periodic alternating training algorithm takes alternating training way in sequence cycle. First, the parameters of $\boldsymbol{\pi}$ are optimized to maximize J while the parameters of u are held within M_1 training episodes. Then, the parameters of $\boldsymbol{\pi}$ are held while the parameters of u are trained to minimize J within M_2 training episodes. This sequence is repeated until convergence to J^*. Our alternating scenario is distinguished from previous research [39]. The episode-by-episode alternating training scenario easily leads to a slow converge or instability for MARL models [43]. The cycle period is introduced to speed up the convergence of both the adversary and the victim. Enough training episodes in one training cycle keep the stability and performance for both sides. The whole learning process is given in Algorithm 3.

Algorithm 3 Periodic Alternating Training

Input: Environment $\mathcal{E}$, number of cycle periods C, number of episodes M_1 for π in one cycle, number of episodes M_2 for u in one cycle, number of time steps of each episode K.
Output: Parameters θ of $\boldsymbol{\pi}$ and parameters ϕ of u.
Initialization: Parameters θ of $\boldsymbol{\pi}$ and parameters ϕ of u.

for cycle = 1 to C **do**
  Get the optimal adversary policy $u_\phi^* \leftarrow u_\phi$
  for episode = 1 to M_1 **do**
    Initialize micro-grid environment $\mathcal{E}$ and get observations of each building o_0^i, as the input of each agent i
    while step < K **do**
      Run u_ϕ^* to generate the adversarial example, $\delta_t = u_\phi^*(o_t^j)$, $o_t^{j\prime} = \delta_t + o_t^j$
      Agents receive observations $\boldsymbol{o}_t' = \{o_t^1, \ldots, o_t^{j\prime}, \ldots, o_t^N\}$
      Run $\boldsymbol{\pi}_\theta$ to get control actions $\boldsymbol{a}_t' = \{\pi_{\theta^1}(o_t^1), \ldots, \pi_{\theta^j}(o_t^{j\prime}), \ldots, \pi_{\theta^N}(o_t^N)\}$
      Perform actions $\boldsymbol{a}_t'$ and get next state s_{t+1}
      Receive each agents rewards $\boldsymbol{r}_t = \{r^i(s_t, \boldsymbol{a}_t', s_{t+1})\}_{i\in N}$
      Store trajectories $\{(\boldsymbol{o}_t', \boldsymbol{a}_t', \boldsymbol{o}_{t+1}, \boldsymbol{r}_t)\}$ in replay buffer $\boldsymbol{D}_\pi$ for $\boldsymbol{\pi}_\theta$
      Update the parameters of the victim policy, $\theta \leftarrow$ Policy Optimizer $(\boldsymbol{D}_\pi, \boldsymbol{\theta})$
    end while
  end for
  Get the optimal victim policy $\boldsymbol{\pi}_\theta^* \leftarrow \boldsymbol{\pi}_\theta$
  for episode = 1 to M_2 **do**
    Initialize micro-grid environment $\mathcal{E}$ and get observations of each building o_0^i, as the input of each agent i
    while step < K **do**
      Run u_ϕ to generate the adversarial example, $\delta_t = u_\phi(o_t^j)$, $o_t^{j\prime} = \delta_t + o_t^j$
      Agents receive observations $\boldsymbol{o}_t' = \{o_t^1, \ldots, o_t^{j\prime}, \ldots, o_t^N\}$
      Run $\boldsymbol{\pi}_\theta^*$ to get control actions $\boldsymbol{a}_t' = \{\pi_{\theta^{1*}}(o_t^1), \ldots, \pi_{\theta^{j*}}(o_t^{j\prime}), \ldots, \pi_{\theta^{N*}}(o_t^N)\}$
      Perform actions $\boldsymbol{a}_t'$ and get next state s_{t+1}
      Receive adversarial rewards r_t^{adv}
      Store trajectories $\{(o_t^j, \delta_t, o_{t+1}^j, r_t^{adv})\}$ in replay buffer D_u for the adversary policy u_ϕ
      Update the parameters of the adversary policy, $\phi \leftarrow$ Policy Optimizer (D_u, ϕ)
    end while
  end for
end for

4. Performance evaluation metrics

To quantify the performance of the proposed approach, seven metrics are defined and employed based on CityLearn Challenge 2021 [44], as follows:

- *Ramping*: It represents the accumulated ramping of electricity consumption, indicating the flat level of the demand curves, as followed by:

$$Ramping = \sum_t^T \left| P_{all,t}^{grid} - P_{all,t-1}^{grid} \right|, t \in T \tag{26}$$

where $P_{all,t}^{grid} = \sum_{i=1}^N P_{i,t}^{grid}$ is the total net electricity consumption of the district buildings at each time step.

- *1-Load factor*: It evaluates the energy usage efficiency:

$$1 - Load\ factor = 1 - \frac{\text{average } P_{all,t}^{grid}}{\max P_{all,t}^{grid}}, t \in T \tag{27}$$

where the average net electricity load divided by the maximum electricity load during one year is the load factor. The higher the load factor, the smaller the generation cost for the same maximum load.

- *Avg. daily peak*: It is the average daily net peak demand during one year:

$$Avg.\ daily\ peak = \text{average}\ P^{grid}_{all,day_d}, d \in T/24 \tag{28}$$

where $P^{grid}_{all,day_d} = \max P^{grid}_{all,t}, t \in [24d+1, 24(d+1)]$ denotes the daily net peak demand and d represents the day index.

- *Peak demand*: It represents maximal daily net peak demand during one year:

$$Peak\ demand = \max P^{grid}_{all,day_d}, d \in T/24 \tag{29}$$

- *Net elec. consumption*: It represents the total district net electricity consumption during one year.

$$Net\ elec.\ consumption = \sum_{t}^{T} P^{grid}_{all,t}, t \in T \tag{30}$$

- *Carbon emissions*: It calculates the total amount of district carbon emissions.

$$Carbon\ emissions = \sum_{t}^{T} \max\left\{0, P^{grid}_{all,t}\right\} \times CI_t, t \in T \tag{31}$$

where CI_t is the carbon intensity at time step t, which measures how much CO2 is being produced per unit of electrical energy generated.

- *score*: It is the average value of the above metrics.

$$\begin{aligned} Score =& (Ramping + 1 - Load\ factor \\ &+ Avg.\ daily\ peak + Peak\ demand + \\ &Net\ elec.\ consumption + Carbon\ emissions)/6 \end{aligned} \tag{32}$$

Meanwhile, the CityLearn platform provides a rule-based controller (RBC) as a baseline to measure the performance of RL algorithms. The RBC aims at minimizing the electricity cost by storing energy at night and releasing it during the day. Note that all metrics values are normalized by the metrics value of RBC. Any metric > 1 is worse than that of the RBC, and < 1 means that the controller is better than the RBC. Finally, the *score* is presented to evaluate the average value of all metrics comprehensively, and the lower values of metrics indicate the better resilience of the DR management systems.

5. Case study

5.1. Data description

The proposed RAMARL-DR framework is constructed based on the open-source OpenAI Gym environment, CityLearn [45], which provides a standard research platform for the application of MARL in urban building energy management and DR. In particular, the platform includes four energy demand datasets, which have been pre-simulated using EnergyPlus in four different climate zones (Z1–Z4) of the USA, respectively. This work is carried out based on hot-humid climate Z1, which contains one-year energy data of a micro-grid with nine buildings on an hourly time slot. As provided in Table 1, there are one medium office (id=1), one fast food restaurant (id = 2), one standalone retail (id = 3), one strip mall retail (id = 4), and five medium residential buildings (id=5-9) in the building group. Each building has its individual different energy storage capacity, PV generation capacity, and different energy consumption profiles. The $\overline{Q}^{hp}$, $\overline{Q}^{eh}$ and $\overline{Q}^{tes}$ are automatically sized by SoC^{tes}. The thermal energy of the building contains cooling energy, heating energy, and DHW energy stored by storage devices, such as chilled water and DHW tanks. Moreover, their storage capacities are represented as the multiple hours the storage device can satisfy the maximum annual hourly cooling or heating demand if fully charged.

Table 1
Technical parameters of energy components in different buildings.

Building ID	η^{hp}_{tech} (%)	η^{eh} (%)	η^{ees} (%)	$\overline{P}^{ees}$ (kW)	$\underline{S}^{ees}$ (kWh)	$\overline{S}^{ees}$ (kWh)
1	20	90	90	100	0	140
2	21	92	90	40	0	80
3	23	87	90	20	0	50
4	22	90	90	30	0	75
5	24	90	90	25	0	50
6	20	85	90	10	0	30
7	22	90	90	15	0	40
8	24	93	90	10	0	30
9	22	90	90	20	0	35

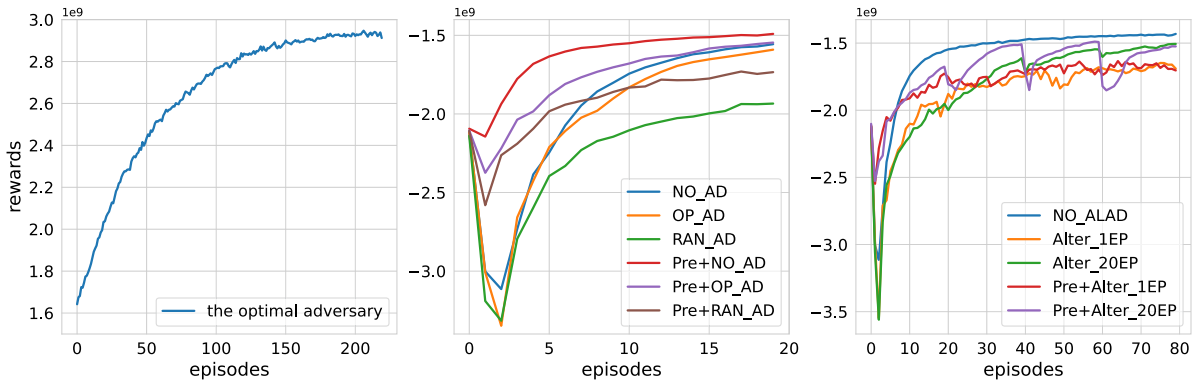
5.2. Algorithm setting

The MARL-based DR management system employs the MARLISA controller from paper [46], which provides more effective load shaping in model-free, decentralized, and scalable with the cooperating setting. The controller extends the SAC algorithm into a multi-agent cooperation fashion through reward sharing and mutual information sharing. Each agent controls one building and executes an iterative sequential action selection algorithm for coordinated DR. Specifically, the first agent picks an action and predicts how much electricity the building will consume if that action is taken. The electricity consumption of the building on the next time step is predicted with normalized observation o^i and action a^i by a well-trained gradient boosting decision tree (GBDT). Then, the current agent's information about the action and predicted consumption is shared with the next agent, who acts in the same operation, sharing the details received previously. Although the action selection of the agents in the MARLISA controller is sequential, their action execution and observation are synchronous. Table 2 presents the hyperparameters used for MARLISA in simulation experiments.

5.3. Experiment design

To validate the superior performance of the proposed RAMARL-DR framework in enhancing the resilience of MARL-based DR management system against adversarial attacks and study the effects of training scenarios on the model performance, we implement three groups of experiments: (1) *Train the optimal adversary with a fixed victim policy*: In this case, we first train a victim policy $\boldsymbol{\pi}$ of the MARL-based DR management system. Then, the victim policy $\boldsymbol{\pi}$ is fixed, and the optimal adversary u is learned by Algorithm 1 to explore the vulnerability. For the test scenario, experiment 1 compares the performance of the MARL-based DR management system with control policy $\boldsymbol{\pi}$ under no attack (without attack), random attack (the random adversary generates attacks), and optimal attack (the optimal adversary u generates attacks).

(2) *Train the victim policy with a fixed optimal adversary*: This group experiments demonstrate the effects of adversarial training with an optimal adversary on the robustness improvement of the MARL-based DR management system by executing six different training scenarios for the MARLISA controller, denoted by NO_AD, OP_AD, RAN_AD, Pre+NO_AD, Pre+OP_AD, Pre+RAN_AD, to compare their performance, as illustrated in Table 3. It can be seen that the former three models take adversarial training 20 episodes with no adversary, optimal, and random adversaries. In comparison, the latter three models take normal pre-training and then robust adversarial training for 20 episodes with no adversary, optimal, and random adversaries. The adversarial training follows Algorithm 2 with different adversaries, where the optimal adversary has been trained in experiment 1, and the random adversary generates random perturbation within bounds. For the test scenario, in experiment 2, the DR management systems with the above trained different controllers are evaluated under no attack, random attack, and optimal attack.



(a) experiment1 (b) experiment2 (c) experiment3

Fig. 4. The evolution of episodic cumulative rewards for three group experiments.

Table 2
Hyperparameter of the MARLISA controller.

	Learning rate lr	Decay rate τ	Discounting rate γ	Buffer size	Batch size	Hidden layers
MARLISA [46]	$3e-4$	$5e-3$	0.99	1e5	256	[256, 256]

Table 3
Six different training scenarios of the DR management system with the MARLISA controller in experiment 2.

Model	Normal pre-training	Robust adversarial training	
	Episodes	Episodes	Adversary
NO_AD	/	20	No adversary
OP_AD	/	20	The optimal adversary
RAN_AD	/	20	The random adversary
Pre+NO_AD	20	20	No adversary
Pre+OP_AD	20	20	The optimal adversary
Pre+RAN_AD	20	20	The random adversary

(3) *Periodic alternating robust adversarial training*: In this group experiments, we execute periodic alternating robust adversarial training as indicated in the Algorithm 3, where the victim policy π and the adversarial policy are training alternatively with different training episodes. To investigate the effects of whether taking pre-training and episodes number M_1, M_2 on model performance, we take five training scenarios, NO_ALAD, Alter_1EP, Alter_20EP, Pre+Alter_1EP, Pre+Alter_20EP, as shown in Table 4. The controller model NO_ALAD takes normal training 80 episodes without attacks. The group models {Alter_1EP, Pre+Alter_1EP} are compared with {Alter_20EP, Pre+Alter_20EP} to evaluate the effects of episodes number on the model performance. Besides, the group models {Alter_1EP, Alter_20EP} are compared with {Pre+Alter_20EP, Pre+Alter_20EP} to study the effects of whether taking pre-training. In this case, there are four test scenarios to verify the resilience of five controllers, no attack, random attack, fixed optimal attack generated by the fixed optimal adversary, and alternative optimal attack generated by optimal alternative adversaries of respective controller models.

All the building DR management systems with the MARLISA controller are trained and tested in the same simulation environment.

5.4. Training performance

This section aims to evaluate the training performance of the proposed RAMARL-DR model for three experiments as designed in Section 5.3, of which their learning curves are illustrated in Fig. 4(a)–(c), respectively.

The first experiment is designed to train an optimal adversary attack while keeping the model policy being fixed of the MARLISA controller inside the DR management system. It can be observed from Fig. 4(a) that the optimal adversary (attack) policy trends to reach convergence after 200 episodes.

In experiment 2, six training models for the MARLISA controller are trained and compared while keeping the adversary policy being fixed (collected) from experiment 1. The first observation from Fig. 4(b) is that the cumulative rewards of six MARLISA controllers all reach convergence within 20 episodes; the converged reward levels, however, are different. More specifically, the rewards of models RAN_AD (green) and Pre+RAN_AD (brown) are much lower than the others; this is mainly due to the random adversary policy without taking the optimal attacking strategies into consideration. On the other hand, the models OP_AD (orange) and Pre+OP_AD (purple) converge nearly to the models without adversarial training, i.e., NO_AD (blue) and Pre+NO_AD (red). Such good performance shows that the optimal adversary contributes to the control policy learning experiences of tackling adversarial attacks, while the random adversary is not effective in RAN_AD and Pre+RAN_AD. Furthermore, it can be observed from Fig. 4(b) that the models with pre-training (Pre+NO_AD, Pre+OP_AD, Pre+RAN_AD) perform well in the initial training stage and get better-converged rewards than those models without pre-training (NO_AD, OP_AD, RAN_AD). Thus, pre-training is evaluated to improve the model performance by introducing an excellent initial exploration policy.

In experiment 3, the baseline controller model NO_ALAD (blue) in Fig. 4(c) reaches convergence around 40 episodes and is expected to obtain the highest cumulative reward without any oscillation. However, once the MARLISA controller policy and adversary policy are updated each other alternately, the learning curves start exhibiting oscillations that vary for the updating frequency and pre-training condition. It can be seen from Fig. 4(c) that Alter_1EP (orange) and Pre+Alter_1EP (red) both converge to the similar reward level. However, Pre+Alter_1EP can speed up as well as obtain a higher starting point than Alter_1EP. This is because Pre+Alter_1EP benefits from a pre-trained policy while Alter_1EP starts from zero knowledge. This phenomenon can also be observed from the comparison between Alter_20EP (green) and Pre+Alter_20EP (purple). However, the difference is that the oscillation is very significant for Pre+Alter_20EP, although it can recover and get back within 20 episodes. As a result, it could be concluded that pre-training does not help improve the training performance but just accelerate the training speed during the initial phase. Finally, it can be found that the rewards of Alter_20EP and Pre+Alter_20EP are much higher than Alter_1EP and Pre+Alter_1EP. As a result, it is suggested that the suitable training episodes (20EP) is capable of exhibiting a better performance in terms of both policy quality and stability.

Table 4
Five different training scenarios of the DR management system with the MARLISA controller in experiment 3.

Model	Normal pre-training	Periodic alternating robust adversarial training				
	Episodes	Adversary	Total episodes	M1	M2	C
NO_ALAD	80	/	/	/	/	/
Alter_1EP	/	The optimal adversary	80	1	1	80
Alter_20EP	/	The optimal adversary	80	20	20	4
Pre+Alter_1EP	20	The optimal adversary	80	1	1	80
Pre+Alter_20EP	20	The optimal adversary	80	20	20	4

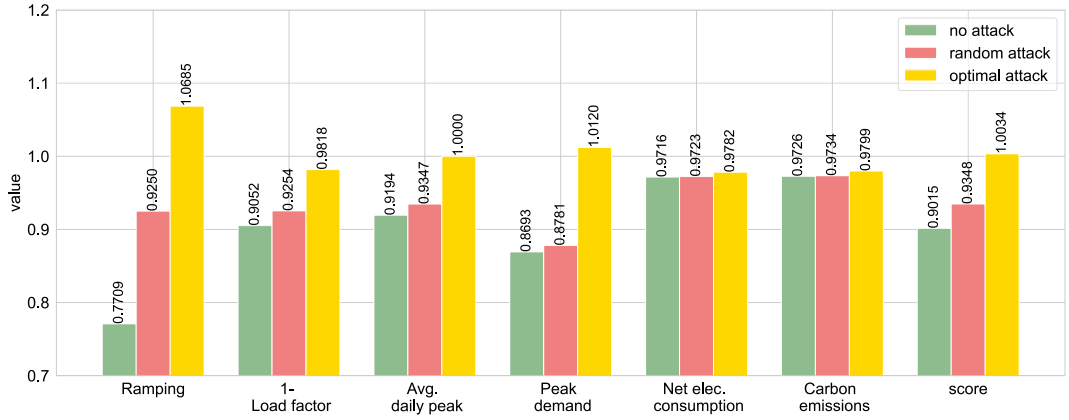


Fig. 5. The evaluation metrics values of the MARLISA controller under no attack, random attack, optimal attack in experiment 1.

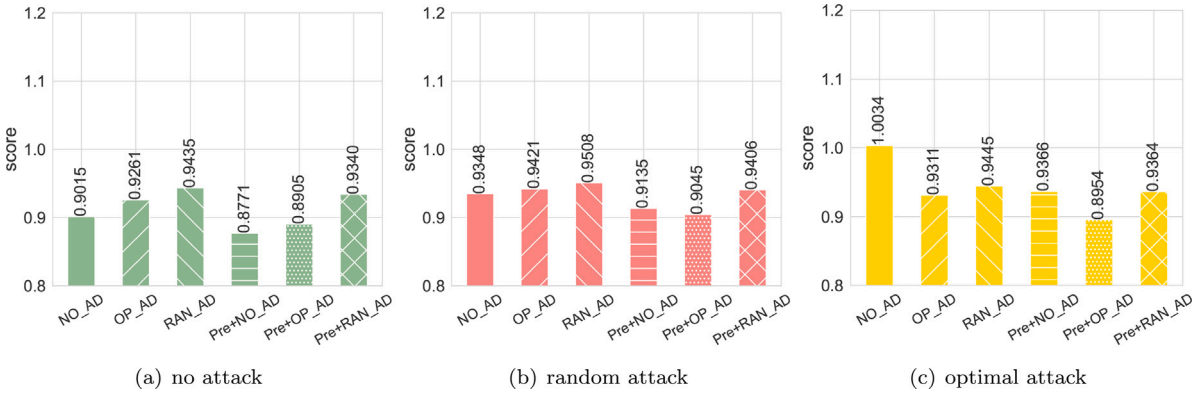


Fig. 6. The evaluation metric *score* of six MARLISA controllers under no attack, random attack and optimal attack in experiment 2.

5.5. Evaluation metrics analysis

After comparing the training performance of different control models for three experiments in Section 5.4, this section aims to quantitatively evaluate and analyze the metric and *score* performance expressed in Section 4 for three experiments, respectively.

5.5.1. Train the optimal adversary policy with a fixed victim policy

Fig. 5 compares the performance of six metrics together with their averaged *score* for different adversarial attacks, i.e., no attack, random attack, and optimal attack. The first observation from Fig. 5 is that the differences among three adversarial attacks are significant for the first four metrics, i.e., *Ramping*, *1-Load factor*, *Avg. daily peak*, and *Peak demand*. In other words, the adversarial attacks have a higher effect on the model resilience in terms of the above four metrics. In detail, the values of *Ramping* under the optimal and random attacks respectively increase by 38.60% and 19.99% compared to no attack. Such increases can be obtained for *1-Load factor* as well in 8.46% and 2.23%. As a result, it can be concluded that the demand curve is much uneven with adversarial attacks. In addition, another two metrics *Avg. daily peak* and *Peak demand* are slightly increased by 1.66% and 1.01% under random attack, while significantly increased by 8.77% and 16.42% under optimal attack. Nevertheless, the total net electricity demand *Net elec. consumption* and total carbon emissions *Carbon emissions* are almost the same for different attack models. Finally, the averaged *score* under optimal attack is 11.30% and 7.34% higher than no attack and random attack, respectively. In conclusion, adversarial attacks do not impact the total amount of energy storage and release but influence the energy storage or release at each time step, which causes the demand curve to oscillate.

5.5.2. Train the victim policy with a fixed optimal adversary

In this section, six MARLISA controllers taking robust adversarial training are tested individually for three kinds of fixed attacks, i.e., no, random, and optimal. More specifically, their *score* performance and the associated metric values are presented in Fig. 6 and Table 5, respectively.

It can be observed from Fig. 6 that the performance of *score* with pre-training (Pre+NO_AD, Pre+OP_AD, Pre+RAN_AD) is always better than those without pre-training (NO_AD, OP_AD, RAN_AD) for all three

Table 5
Metrics performance of six MARLISA controllers under no attack, random attack and optimal attack in experiment 2.

Case	Model	Ramping	*1-Load factor*	*Avg. daily peak*	*Peak demand*	*Net elec. consumption*	*Carbon emissions*
No attack	NO_AD	0.7709	0.9052	0.9194	0.8693	0.9716	0.9726
	OP_AD	0.8367	0.9309	0.9248	0.9189	0.9721	0.9730
	RAN_AD	0.9199	0.9356	0.9502	0.9065	0.9737	0.9750
	Pre+NO_AD	**0.6895**	**0.8876**	**0.8887**	**0.8536**	0.9712	0.9721
	Pre+OP_AD	0.7353	0.9104	0.8990	0.8588	**0.9694**	**0.9700**
	Pre+RAN_AD	0.9024	0.9375	0.9402	0.8747	0.9741	0.9749
Random attack	NO_AD	0.9250	0.9254	0.9347	0.8781	0.9723	0.9734
	OP_AD	0.9140	0.9434	0.9332	0.9159	0.9726	0.9736
	RAN_AD	0.9617	0.9351	0.9519	0.9073	0.9738	0.9750
	Pre+NO_AD	0.8479	**0.9092**	0.9056	0.8732	0.9721	0.9732
	Pre+OP_AD	**0.7969**	0.9188	**0.9026**	**0.8688**	**0.9697**	**0.9704**
	Pre+RAN_AD	0.9334	0.9399	0.9438	0.8773	0.9742	0.9750
Optimal attack	NO_AD	1.0685	0.9818	1.0000	1.0120	0.9782	0.9799
	OP_AD	0.8587	0.9366	0.9300	0.9162	0.9723	0.9732
	RAN_AD	0.9233	0.9378	0.9489	0.9085	0.9736	0.9748
	Pre+NO_AD	0.8643	0.9309	0.9429	0.9254	0.9773	0.9788
	Pre+OP_AD	**0.7499**	**0.9231**	**0.8978**	**0.8609**	**0.9699**	**0.9705**
	Pre+RAN_AD	0.9083	0.9381	0.9417	0.8808	0.9742	0.9750

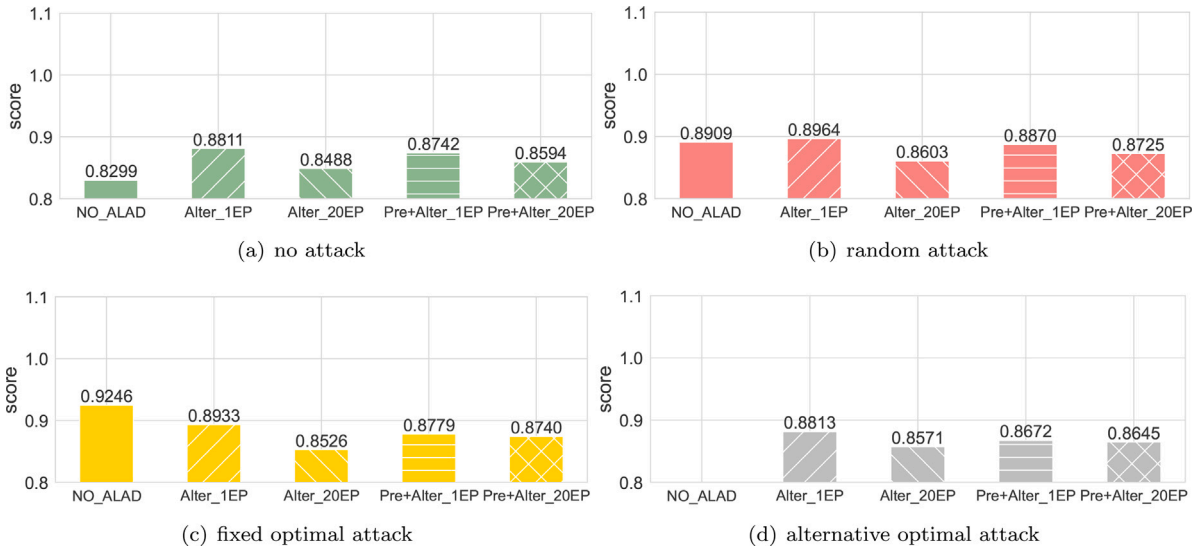


Fig. 7. The evaluation metric *score* of five MARLISA controllers under no attack, random attack, fixed optimal attack and alternative optimal attack in experiment 3.

test scenarios. As a result, such comparisons can demonstrate the benefits of pre-training in improving *score* performance in case of any test adversary strategy and adversarial training policy. Now, further analysis should be focused on the three training models with pre-training, i.e., Pre+NO_AD, Pre+OP_AD, Pre+RAN_AD. More specifically, in the first test scenario of no attack, Pre+NO_AD achieves the best *score* performance (i.e., lowest *score* value), which means the random and optimal adversarial training does not help improve the performance if there is no attack in the DR program. However, in the latter two test scenarios, when random and optimal adversary strategies are used for attacking, the optimal adversarial training Pre+OP_AD achieves the best performance. This is because robust adversarial training is capable of efficiently mitigating the influence caused by the attacks.

As defined in Section 4, the *score* is averaged by six metrics. Table 5 thus provides and compares the specific value of each metric for different adversarial attack strategies and training models. It can be found that there are not many differences between the last two metrics *Net elec. consumption* and *Carbon emissions* among different training models for each attack strategy. That means the total energy consumption is unaffected so that the energy usage security inside the building is always guaranteed. To this end, it can be found from Table 5 that the major differences are coming from the first four metrics, i.e., *Ramping*, *1-Load factor*, *Avg. daily peak* and *Peak demand*. Notably, all these four metrics are related to the demand profiles of daily usage. Similar to the *score* performance illustrated in Fig. 6, (1) Pre+NO_AD achieves the best metric performance in the first test scenario of no attack; and (2) Pre+OP_AD achieves the best metric performance in the latter two test scenarios of random and optimal attacks. As a result, it can be concluded that the demand profile exhibits a flatter pattern (reflected by lower *Ramping*, *1-Load factor*) meanwhile the demand peak is significantly reduced (reflected by lower *Avg. daily peak*, *Peak demand*) after considering a robust adversarial training model Pre+OP_AD.

5.5.3. Periodic alternating robust adversarial training

In experiment 3, a more complex adversarial training structure that alternatively trains controller policy and adversary policy is proposed. In addition, two periodic updating settings per episode and 20 episodes are evaluated in this experiment to investigate the impact of periodic training on policy performance. Finally, the pre-training method is also applied to the two periodic updating models for comparison, respectively.

Fig. 7 illustrates the *score* performance of five controllers for four different test scenarios, of which the last scenario alternative optimal attack is the new one for experiment 3, defined in Section 5.3. Similar to the results obtained in experiment 2, no alternative adversarial training achieves the best performance (0.8299) among five controllers

Table 6
Metrics performance of five MARLISA controllers under no attack, random attack, fixed optimal attack and alternative optimal attack in experiment 3.

Case	Model	*Ramping*	*1-Load factor*	*Avg. daily peak*	*Peak demand*	*Net elec. consumption*	*Carbon emissions*
No attack	**NO_ALAD**	**0.5559**	**0.8277**	**0.8570**	**0.8010**	0.9686	0.9693
	Alter_1EP	0.7455	0.8762	0.8977	0.8290	0.9687	0.9695
	Alter_20EP	0.6166	0.8408	0.8689	0.8300	0.9678	0.9685
	Pre+Alter_1EP	0.6821	0.8791	0.8864	0.8644	**0.9666**	**0.9669**
	Pre+Alter_20EP	0.6558	0.8669	0.8750	0.8236	0.9672	0.9678
Random attack	NO_ALAD	0.7831	0.8899	0.8852	0.8475	0.9695	0.9702
	Alter_1EP	0.8096	0.8778	0.9019	0.8503	0.9689	0.9697
	Alter_20EP	**0.6702**	**0.8489**	**0.8724**	0.8337	0.9680	0.9687
	Pre+Alter_1EP	0.7547	0.8767	0.8923	0.8640	**0.9669**	**0.9673**
	Pre+Alter_20EP	0.7164	0.8723	0.8784	**0.8324**	0.9673	0.9679
Fixed optimal attack	NO_ALAD	0.7862	0.9240	0.9233	0.9646	0.9740	0.9753
	Alter_1EP	0.7728	0.8687	0.9009	0.8782	0.9690	0.9698
	Alter_20EP	**0.6536**	**0.8402**	**0.8721**	**0.8119**	0.9685	0.9693
	Pre+Alter_1EP	0.7144	0.8720	0.8892	0.8561	**0.9675**	**0.9681**
	Pre+Alter_20EP	0.7196	0.8697	0.8842	0.8333	0.9681	0.9689
alternative optimal attack	NO_ALAD	/	/	/	/	/	/
	Alter_1EP	0.7285	0.8796	0.8929	0.8491	0.9685	0.9693
	Alter_20EP	**0.6325**	**0.8506**	**0.8725**	0.8509	0.9676	0.9684
	Pre+Alter_1EP	0.6848	0.8589	0.8788	**0.8470**	**0.9666**	**0.9671**
	Pre+Alter_20EP	0.6387	0.8795	0.8772	0.8574	0.9669	0.9674

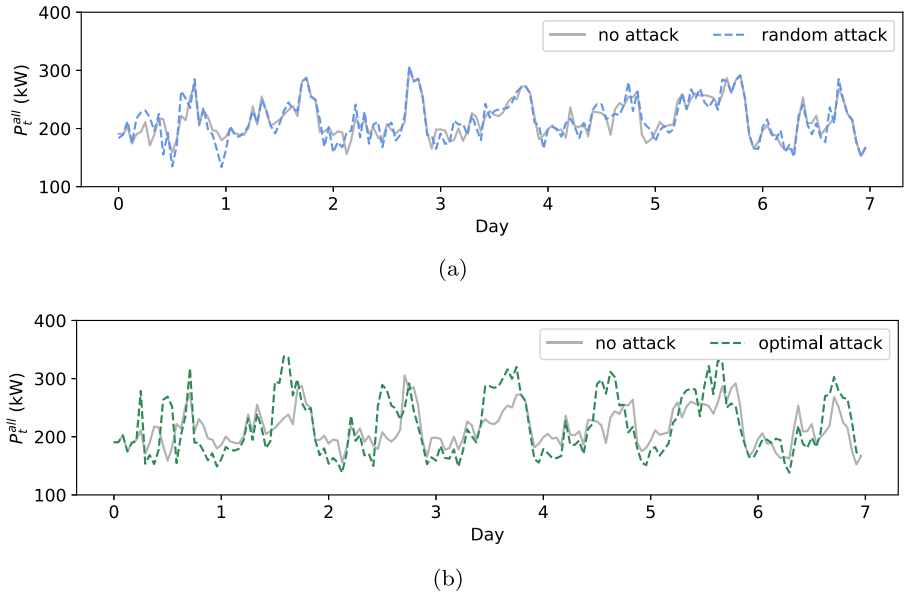


Fig. 8. Weekly district net electricity consumption profiles of the MARLISA controller under no attack, random attack and optimal attack in experiment 1.

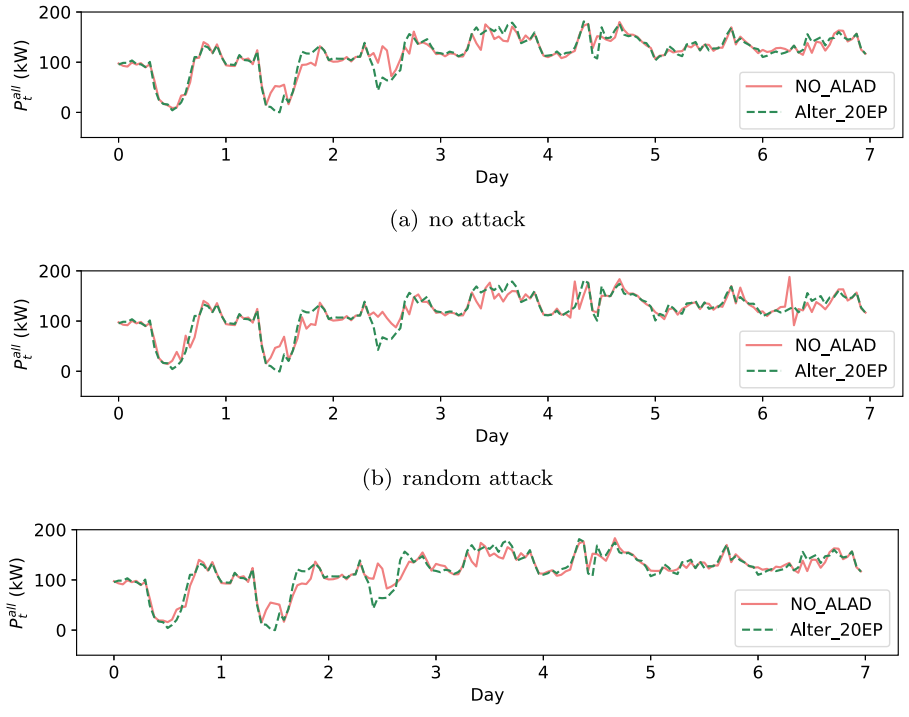
if there is no attack during the test process, as shown in Fig. 7(a). However, once the random and optimal adversarial attacks are considered in the test scenarios, the alternative training controllers do help improve the *score* performance with the relatively lower *score* values. In detail, it can be observed from Fig. 7 that the *score* performance of controllers under 20 episodes' alternative training (i.e., Alter_20EP, Pre+Alter_20EP) performs better than those under 1 episode's alternative training (i.e., Alter_1EP, Pre+Alter_1EP). This trend suggests that effective training episodes (e.g., update per 20 episodes) can improve *score* performance rather than high-frequency updates (e.g., update per episode). The final interesting result found from Fig. 7 is that pre-training controllers (Pre+Alter_1EP, Pre+Alter_20EP) do not help improve the *score* performance and exhibit higher *score* values than those without pre-training (Alter_1EP, Alter_20EP) for all four test scenarios. Such phenomenon seems to contradict the conclusion from experiment 2 as discussed in Section 5.5.2.

The detailed metric performance associated with each *score* is presented in Table 6. First of all, it can be found that both metrics *Net elec. consumption* and *Carbon emissions* show their little differences for all five controllers and four test scenarios. Furthermore, the controller NO_ALAD performs the best under no attack scenario, while Alter_20EP performs the best for the rest of the three attack scenarios. These results follow the same trends as the *score* performance illustrated in Fig. 7.

5.6. Performance analysis

To further elaborate on the generalization performance of the learned DR controller models, this section aims to (1) compare weekly demand profiles evaluated in experiments 1 and 3; (2) analyze the energy profiles and the flexibility of storage inside the building energy management.

Fig. 8 shows the weekly demand profiles for three test scenarios in experiment 1, where random attack (blue line in (a)) and optimal attack (green line in (b)) are compared with the baseline no attack (gray lines in both (a) and (b)). It can be observed that the optimal attack exhibits a more uneven pattern, a higher peak value, and more demand peaks than the random attack, given the baseline of no attack. It is believed that the optimal attack exposes the vulnerability of the MARLISA controller and weakens the resilience of the DR management system. On the other hand, Fig. 9 shows the weekly demand profiles of controllers

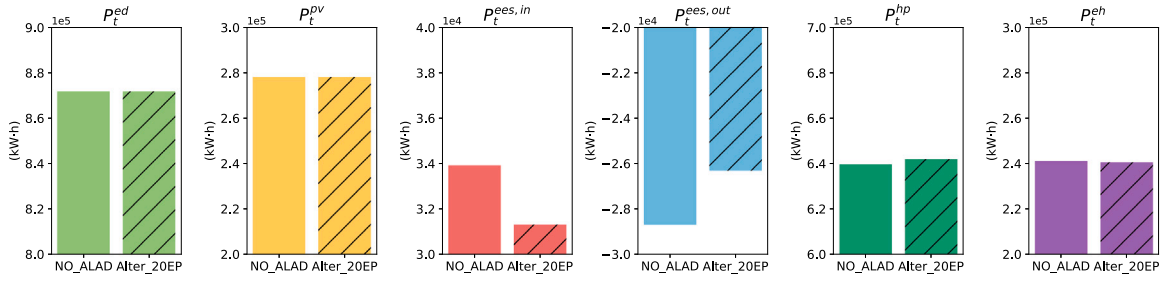


(a) no attack

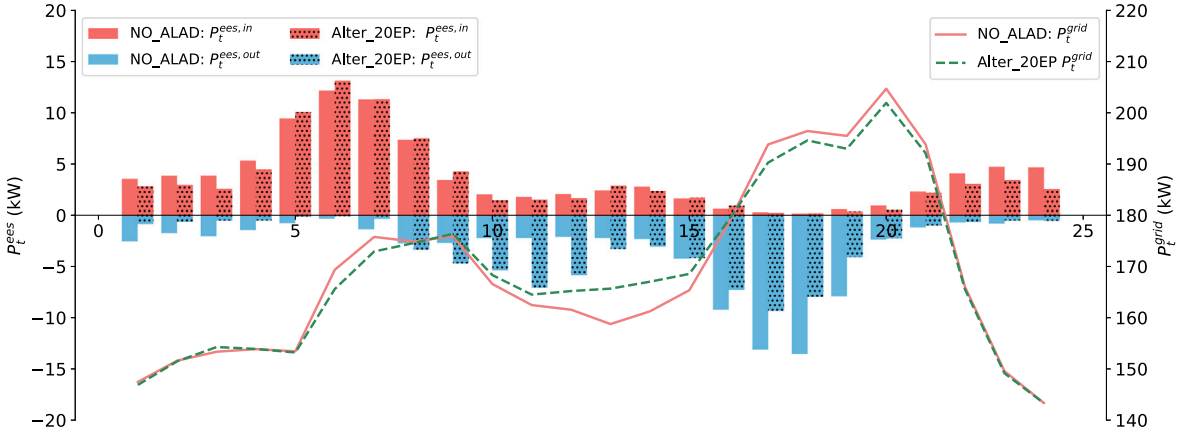
(b) random attack

(c) optimal attack

Fig. 9. Weekly district net electricity consumption profiles of the controller models NO_ALAD and Alter_20EP under no attack, random attack and optimal attack in experiment 3.



(a) six electricity consumption on a year



(b) hourly average net electricity consumption and P_t^{ees}

Fig. 10. Various electricity consumption of the controller models NO_ALAD and Alter_20EP.

NO_ALAD and Alter_20EP for three test scenarios in experiment 3, respectively. It can be found that the controller Alter_20EP exhibits a better performance than NO_ALAD in terms of fewer demand peaks and a flatter pattern. Furthermore, the demand profiles of controller Alter_20EP are robust under all cases, while controller NO_ALAD is sensitive to the test scenarios.

After demonstrating the differences in electric demand profiles between three attack scenarios, the other objective of this section lies in investigating the energy portfolios inside the building management system with and without robust adversarial training under attacks, indicated by models Alter_20EP and NO_ALAD, respectively. To this end, Fig. 10(a) compares the total yearly energies of building's six components for NO_ALAD and Alter_20EP, including inflexible demands P_t^{ed}, PV generation P_t^{pv}, electric energy storage charging $P_t^{ees,in}$ and discharging $P_t^{ees,out}$ behaviors, consumption of heat pump P_t^{hp}, and electric heater P_t^{eh}.

It can be observed from Fig. 10(a) that the major differences between NO_ALAD and Alter_20EP are coming from the storage charging (red) and discharging (blue) behaviors. In detail, buildings under Alter_20EP less deploy their storage flexibility to balance the demand–supply quantity with respect to NO_ALAD. To better demonstrate the effect of storage strategies on buildings' net demand profile, the hourly averaged charging and discharging schedules of storage together with their resulted demand profiles for NO_ALAD and Alter_20EP are plotted in Fig. 10(b). Although NO_ALAD and Alter_20EP differ with regard to the robust adversarial training (under the latter) or not (under the former), they exhibit some common trends. Storage starts to charge the battery during the periods of morning and evening when demand is relatively low, while discharging the battery during the periods of mid-day and night when PV is abundant, and demand is relatively high. Nonetheless, there are also evident differences between these two models. Firstly, storage under Alter_20EP increases the discharging behaviors in the mid-day in order to increase the absorption of PV resources while increasing the demand (negative) levels. Secondly, storage under Alter_20EP reduces the discharging behaviors at night in order to reduce the demand peaks. Such two differences come together to flatten the demand profiles, as depicted in Fig. 10(b).

6. Conclusion

This paper proposes a novel robust MARL-based DR methodological framework, RAMARL-DR, based on the optimal adversary to enhance the resilience of the DR management system against cyber-attacks. In particular, a single-agent SAC algorithm is employed to learn an optimal adversary policy, which can generate the adversarial examples causing the worst-case performance for the multi-agent system. After that, the learned optimal adversary policy can be used to train the victim policy in a periodic alternative way via conducting robust adversarial training. The superior performance of the proposed RAMARL-DR method has been demonstrated based on an OpenAI Gym environment CityLearn. The main conclusions stemming from the results include: (1) the current MARL-based DR approach is indeed vulnerable under adversarial attacks; (2) the robustness of MARL-based DR can be improved after conducting adversarial training with the fixed optimal adversary, while the adversarial training with the random adversary negatively affects its performance; (3) the proposed periodic alternating robust adversarial training to resolve the convergence issue in the MARL-based DR management system caused by episode-by-episode alternating training.

Future work could be conducted to investigate the identification method for the most vulnerable agents in the multi-agent environment and further improve the robustness of the MARL-based DR management system by optimizing the periods of alternating training. In addition, this work only focuses on the effects of demand–supply balances and flatting net demand. Another future work thus aims to investigate the economic perspective of this problem and involve the grid electricity price signals.

CRediT authorship contribution statement

Lanting Zeng: Methodology, Software, Data curation, Validation, Formal analysis, Writing – original draft, Writing – review & editing. **Dawei Qiu:** Methodology, Data curation, Formal analysis, Writing – original draft, Writing – review & editing. **Mingyang Sun:** Methodology, Writing – original draft, Writing – review & editing, Conceptualization, Project administration, Supervision, Funding acquisition.

Declaration of competing interest

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.

Acknowledgments

This work was supported in part by the the National Natural Science Foundation of China under Grants 52161135201, U20A20159, 62103371.

References

[1] Plessmann G, Blechinger P. How to meet EU GHG emission reduction targets? A model based decarbonization pathway for Europe's electricity supply system until 2050. Energy Strategy Rev 2017;15:19–32.
[2] Perez-Arriaga IJ, Batlle C. Impacts of intermittent renewables on electricity generation system operation. Econ Energy Environ Policy 2012;1(2):3–18.
[3] Wang Q, Zhang C, Ding Y, Xydis G, Wang J, Østergaard J. Review of real-time electricity markets for integrating distributed energy resources and demand response. Appl Energy 2015;138:695–706.
[4] Assante M. Confirmation of a coordinated attack on the Ukrainian power grid. SANS industrial control systems security. 2016, URL https://www.sans.org/blog/confirmation-of-a-coordinated-attack-on-the-ukrainian-power-grid/.
[5] Liu Y, Ning P, Reiter MK. False data injection attacks against state estimation in electric power grids. ACM Trans Inf Syst Secur 2011;14(1):1–33.
[6] Kosut O, Jia L, Thomas RJ, Tong L. Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In: 2010 first IEEE international conference on smart grid communications. IEEE; 2010, p. 220–5.
[7] Coppolino L, Romano L, et al. Exposing vulnerabilities in electric power grids: An experimental approach. Int J Crit Infrastruct Prot 2014;7(1):51–60.
[8] Dabrowski A, Ullrich J, Weippl ER. Grid shock: Coordinated load-changing attacks on power grids: The non-smart power grid is vulnerable to cyber attacks as well. In: Proceedings of the 33rd annual computer security applications conference. 2017. p. 303–14.
[9] Lu Z, Wang W, Wang C. Modeling, evaluation and detection of jamming attacks in time-critical wireless applications. IEEE Trans Mob Comput 2014;13(8):1746–59. http://dx.doi.org/10.1109/TMC.2013.146.
[10] Lu Z, Lu X, Wang W, Wang C. Review and evaluation of security threats on the communication networks in the smart grid. In: 2010-milcom 2010 military communications conference. IEEE; 2010, p. 1830–5.
[11] Sgouras KI, Birda AD, Labridis DP. Cyber attack impact on critical smart grid infrastructures. In: ISGT 2014. IEEE; 2014, p. 1–5.
[12] Albadi MH, El-Saadany EF. Demand response in electricity markets: An overview. In: 2007 IEEE power engineering society general meeting. IEEE; 2007, p. 1–5.
[13] Yang J, Zhao J, Luo F, Wen F, Dong ZY. Decision-making for electricity retailers: A brief survey. IEEE Trans Smart Grid 2017;9(5):4140–53.
[14] Lee A. Electric sector failure scenarios and impact analyses. Technical working group, 1, National electric sector cybersecurity organization resource (NESCOR); 2013.
[15] Wang J, Zhong H, Ma Z, Xia Q, Kang C. Review and prospect of integrated demand response in the multi-energy system. Appl Energy 2017;202:772–82.
[16] Kelley MT, Pattison RC, Baldick R, Baldea M. An MILP framework for optimizing demand response operation of air separation units. Appl Energy 2018;222:951–66.
[17] Heidari A, Mortazavi SS, Bansal RC. Stochastic effects of ice storage on improvement of an energy hub optimal operation including demand response and renewable energies. Appl Energy 2020;261:114393.
[18] Bianchini G, Casini M, Vicino A, Zarrilli D. Demand-response in building heating systems: A model predictive control approach. Appl Energy 2016;168:159–70.
[19] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT Press; 2018.
[20] Vázquez-Canteli JR, Nagy Z. Reinforcement learning for demand response: A review of algorithms and modeling techniques. Appl Energy 2019;235:1072–89.
[21] Qiu D, Ye Y, Papadaskalopoulos D, Strbac G. Scalable coordinated management of peer-to-peer energy trading: A multi-cluster deep reinforcement learning approach. Appl Energy 2021;292:116940.

[22] Lu R, Hong SH. Incentive-based demand response for smart grid with reinforcement learning and deep neural network. Appl Energy 2019;236:937–49.
[23] Wang B, Li Y, Ming W, Wang S. Deep reinforcement learning method for demand response management of interruptible load. IEEE Trans Smart Grid 2020;11(4):3146–55.
[24] Du Y, Zandi H, Kotevska O, Kurte K, Munk J, Amasyali K, et al. Intelligent multi-zone residential HVAC control strategy based on deep reinforcement learning. Appl Energy 2021;281:116117.
[25] Qiu D, Dong Z, Zhang X, Wang Y, Strbac G. Safe reinforcement learning for real-time automatic control in a smart energy-hub. Appl Energy 2022;309:118403.
[26] Lu R, Hong SH, Yu M. Demand response for home energy management using reinforcement learning and artificial neural network. IEEE Trans Smart Grid 2019;10(6):6629–39.
[27] Lu R, Li Y-C, Li Y, Jiang J, Ding Y. Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management. Appl Energy 2020;276:115473.
[28] Kazmi H, Suykens J, Balint A, Driesen J. Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads. Appl Energy 2019;238:1022–35.
[29] Huang S, Papernot N, Goodfellow I, Duan Y, Abbeel P. Adversarial attacks on neural network policies. 2017, arXiv preprint arXiv:1702.02284.
[30] Zheng Y, Yan Z, Chen K, Sun J, Xu Y, Liu Y. Vulnerability assessment of deep reinforcement learning models for power system topology optimization. IEEE Trans Smart Grid 2021;12(4):3613–23.
[31] Figura M, Kosaraju KC, Gupta V. Adversarial attacks in consensus-based multi-agent reinforcement learning. In: 2021 American control conference. IEEE; 2021, p. 3050–5.
[32] Lin J, Dzeparoska K, Zhang SQ, Leon-Garcia A, Papernot N. On the robustness of cooperative multi-agent reinforcement learning. In: 2020 IEEE security and privacy workshops. IEEE; 2020, p. 62–8.
[33] Kos J, Song D. Delving into adversarial attacks on deep policies. 2017, arXiv preprint arXiv:1705.06452.
[34] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. 2014, arXiv preprint arXiv:1412.6572.
[35] Pattanaik A, Tang Z, Liu S, Bommannan G, Chowdhary G. Robust Deep Reinforcement Learning with Adversarial Attacks. In: Proceedings of the 17th international conference on autonomous agents and multiagent systems. 2018. p. 2040–2.
[36] Chen T, Niu W, Xiang Y, Bai X, Liu J, Han Z, et al. Gradient band-based adversarial training for generalized attack immunity of A3C path finding. 2018, arXiv preprint arXiv:1807.06752.
[37] Tan KL, Esfandiari Y, Lee XY, Sarkar S, et al. Robustifying reinforcement learning agents via action space adversarial training. In: 2020 American control conference. IEEE; 2020, p. 3959–64.
[38] Pinto L, Davidson J, Sukthankar R, Gupta A. Robust adversarial reinforcement learning. In: International conference on machine learning. PMLR; 2017, p. 2817–26.
[39] Zhang H, Chen H, Boning DS, Hsieh C-J. Robust reinforcement learning on state observations with learned optimal adversary. In: International conference on learning representations. 2021.
[40] Zhang K, Sun T, Tao Y, Genc S, Mallya S, Basar T. Robust multi-agent reinforcement learning with model uncertainty. Adv Neural Inf Process Syst 2020;33:10571–83.
[41] Li S, Wu Y, Cui X, Dong H, Fang F, Russell S. Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In: Proceedings of the AAAI conference on artificial intelligence, vol. 33. (01):2019, p. 4213–20.
[42] Yang Y, Wang J. An overview of multi-agent reinforcement learning from game theoretical perspective. 2020, arXiv preprint arXiv:2011.00583.
[43] Gleave A, Dennis M, Kant N, Wild C, Levine S, Russsell S. Adversarial policies: Attacking deep reinforcement learning. In: International conference on learning representations. 2020.
[44] Nagy Z, Vázquez-Canteli JR, Dey S, Henze G. The learn challenge 2021. In: Proceedings of the 8th ACM international conference on systems for energy-efficient buildings, cities, and transportation. 2021. p. 218–9.
[45] Vázquez-Canteli JR, Kämpf J, Henze G, Nagy Z. CityLearn v1. 0: An OpenAI gym environment for demand response with deep reinforcement learning. In: Proceedings of the 6th ACM international conference on systems for energy-efficient buildings, cities, and transportation. 2019. p. 356–7.
[46] Vazquez-Canteli JR, Henze G, Nagy Z. MARLISA: Multi-agent reinforcement learning with iterative sequential action selection for load shaping of grid-interactive connected buildings. In: Proceedings of the 7th ACM international conference on systems for energy-efficient buildings, cities, and transportation. 2020. p. 170–9.