House Energy Management System, for balancing Electricity Costs and Residential Comfort, based on Deep Reinforcement Learning

Aleksandra Kaplar[1], Milan Vidaković[1], Aleksandar Kaplar[1], Jovana Vidaković[2], Jelena Slivka[1]

[1] Faculty of Technical Sciences, University of Novi Sad, Trg Dositeja Obradovica 6, 21000 Novi Sad, Serbia, aleksandra.a@uns.ac.rs, minja@uns.ac.rs, aleksandar.kaplar@uns.ac.rs, slivkaje@uns.ac.rs

[2] Faculty of Sciences, University of Novi Sad, Trg Dositeja Obradovica 4, 21000 Novi Sad, Serbia, jovana@dmi.uns.ac.rs

Abstract: Smart homes are becoming increasingly popular for their potential to reduce electricity costs, through device optimization. Balancing residential comfort with electricity cost reduction presents significant challenges. To tackle this problem, we developed a House Energy Management System (HEMS) using Deep Reinforcement Learning (DRL) to reduce electricity costs, by orchestrating device usage, without compromising residential comfort. The HEMS was trained on a smart home simulation powered by the Typhoon HIL application and supplemented with real-world data, from the Mainflux IoT Platform. The simulation included HEMS-controllable and uncontrollable devices, a solar panel and the electricity grid. We modelled a reward function that balances electricity cost with the residents' comfort and used it to train two DRL models: Double Deep Q Network (DDQN) and Proximal Policy Optimization (PPO). Our findings show that PPO maintains thermal comfort and reduces electricity costs more effectively than does DDQN, particularly in the colder season. As the PPO models' behavior is season-dependent, it can reduce residential effort by automatically adjusting device schedules in response to changing weather conditions.

Keywords: Deep Reinforcement Learning; Double Deep Q Network; Proximal Policy Optimization; House Energy Management System; Smart Home

1 Introduction

The electricity consumed by a typical smart home residence reaches up to 40% of the total energy consumption worldwide [1]. Thus, reducing energy consumption in residences, could reduce residents' electricity costs and positively impact our ecosystem. Cost reduction can be achieved by integrating renewable energy sources into smart homes, which are becoming increasingly common. Solar power, the fastest-growing renewable energy source, is frequently integrated into smart homes to reduce their energy consumption from the electric grid [2].

Unfortunately, integrating a home equipped with solar panels, into the electrical grid, is challenging. The primary issue is that solar panel electricity production highly depends on external weather conditions [3] and is highly variable. Such homes still need to be connected to the electric grid. The grid benefits from uniform electricity consumption throughout the day. However, typical daily consumption patterns exhibit peaks when people return from work [4], whereas solar energy production peaks in the warmest (highest solar irradiance) part of the day [5]. The grid typically mitigates this mismatch with initiatives where electricity prices vary throughout the day [6].

Balancing the smart home's daily electricity consumption from the grid can be managed by orchestrating the appropriate times for the house to rely on a grid or a solar panel as an energy source. We can achieve this by strategically scheduling shiftable devices – devices that do not need to operate immediately upon demand but whose operation can be deferred without compromising residents' comfort.

The House Energy Management System (HEMS) is crucial for managing shiftable devices, as their manual scheduling is time-consuming and unintuitive [7]. HEMS should enable a self-sufficient environment for managing electricity costs and smart home devices according to residents' demands [8]. By incorporating artificial intelligence, we can alleviate these burdens for residents and potentially discover innovative energy-saving strategies. This paper proposes the development of the HEMS that automates the orchestration of smart home devices, balancing electricity costs with residents' comfort. We used Deep Reinforcement Learning (DRL), a combination of Reinforcement Learning (RL) and deep neural networks, to optimize HEMS decision-making. We experimented with two DRL models: Double Deep Q Network (DDQN) and Proximal Policy Optimization (PPO).

The primary challenge in using DRL to train HEMS is the necessity of an appropriate training environment. Training a DRL model from scratch in an actual smart home is impractical due to the significant and unnecessary electricity costs incurred during the initial training stages. Consequently, many researchers opt to use simulated training environments [1] [9-11]. This paper proposes employing Typhoon HIL [12] and MainFlux [13] to simulate the smart home environment. Our simulation includes shiftable devices controlled by the HEMS (air conditioning, dishwasher, solar panel, and washing machine) and uncontrollable devices regulated by the residents (e.g., TV and lighting). The electricity prices are set to vary throughout the day to replicate real-world grid incentives.

The architecture of our smart home simulation is divided into three components (Figure 1). The DRL model observes the smart home's current state, including the status of its devices and residents' comfort demands, and determines the action for the next 15-minute time step, specifically which devices should be operated.

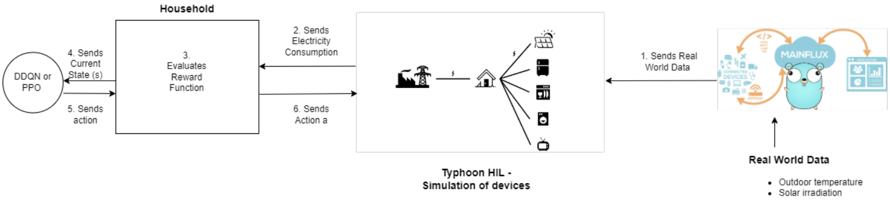


Figure 1

A simplified overview of our solution. Right to the left: The Mainflux IoT platform infuses our simulation with real-world outdoor data; the Typhoon HIL accurately simulates the electricity consumption of each device, with adjustments in solar panel electricity production and indoor temperature based on the infused outdoor data; The smart home module calculates rewards based on electricity costs and residential comfort, which includes considerations for desired device services and thermal comfort; HEMS actions are optimized through the training a DRL model. Two DRL models are evaluated: Proximal Policy Optimization (PPO) and Double Deep Q-Network (DDQN).

Defining a reward function is essential for training a DRL model. In the context of our HEMS, the reward function must balance residential comfort with electricity costs. Following the approach in [14], we constructed a reward function that incorporates both these factors and extended it with a penalty factor. Our observations indicate that this reward enabled the DRL models to increase residential comfort while decreasing electricity costs. The contributions of our paper are as follows:

(1) We propose using Typhoon HIL, a fast and highly accurate third-party software, to simulate a smart home, providing a training environment for DRL models. Typhoon HIL is a flexible and robust testing environment with a user-friendly interface, allowing the users to define and interact with environments and monitor the simulated systems [15-17]. Additionally, we integrate real-world external data into the Typhoon HIL simulation using MainFlux.

(2) We propose a reward function for training DRL models that balances electricity costs with residential comfort. The residents can parametrize it according to their personalized requirements. This paper presents a case study – a hypothetical scenario of residents' comfort demands – demonstrating that DRL models can be effectively trained using the proposed environment and reward.

(3) We found that the PPO model trains faster than DDQN and outperforms DDQN regarding residential thermal comfort.

(4) To the best of our knowledge, previous works used training sets that included several months of data to avoid possible oscillations in calculating the average reward during training. Solutions that used the entire year’s data experienced oscillations in the average reward calculations. We proposed dividing the training and testing data into seasons, allowing for more consistent average reward calculation during training and improving results.

2 Related Work

Over the years, researchers have applied RL in smart homes to schedule the operation of devices and reduce electricity costs. Unlike traditional models that are trained on labelled data sets, RL models require an environment where they can test and evaluate possible actions [1].

2.1 Reinforcement Learning Model's Training Environments

Training an RL model in a real smart home from scratch is impractical. In the early stages of training, RL needs to explore a wide range of actions, many of which may not be beneficial, leading to significant and unnecessary electricity costs. However, real-world data is essential to train a realistic RL-based HEMS. Authors in [18] and [19] address this issue by collecting real-time data via sensors placed in an indoor environment. They use this data to develop models for environment simulations for RL model training. Many other authors also rely on real-world historical data, incorporating it into simulated environments they modelled. [9] [20-22]

Another research direction involves building models that predict relevant factors such as indoor temperature, electricity price, and solar generation rather than using real-time data [9] [10] [23]. Once these prediction models are trained, their hour-ahead predictions are used to train the RL model.

While using historical and real-world data provides the most realistic scenario, it also has practical limitations for training RL models. It is essential to expose the RL model to various plausible scenarios. These scenarios might include extended absences of residents, changes in residential preferences or behavior, power shortages, weather changes, and more. Capturing these scenarios in historical data is not feasible, as it is impractical to disrupt residents' daily lives by enforcing unfavorable events. Moreover, historical data is often available for a limited number of smart home configurations. This lack of diverse data can become problematic as residential environments are often dynamic. New devices may be introduced to the smart home, and HEMS must be able to adapt.

2.2 Deep Reinforcement Learning Models for HEMS

In addition to the traditional RL Q-Learning model, other papers train Deep RL models for developing HEMS. The authors of [19] propose a HEMS solution to reduce electricity costs associated with indoor heating and domestic hot water temperatures while utilizing electricity production from solar panels. They define residential comfort through time constraints for device operation but do not account for variations in electricity prices throughout the day.

Another DRL model that achieved promising results is Double Deep Q-Network (DDQN). Liu et al. [24] proposed HEMS based on the DDQN model, which was trained to optimize electricity costs for smart home devices while considering constraints on operating times and varying electricity prices (i.e., tariff). Our paper also considers the tariffs and preferred operating times for individual devices. However, our residential comfort definition additionally includes thermal comfort.

Forootani et al. [25] implemented a satisfaction-based HEMS that comprises multiple Deep Q-Network (DQN) models. Each DQN model corresponds to a specific device and is trained separately to minimize its electricity cost while considering the residential preferences for using that device. Additionally, their smart home setup includes an electric vehicle. In contrast, our HEMS is based on a single DRL model with a reward function that balances residential comfort with electricity costs through managing multiple smart home devices.

In addition to the previously mentioned papers, which focused on DRL strategies based on value functions (estimations of expected cumulative rewards), another research direction involves DRL based on actor-critic algorithms. Actor-critic algorithms use two neural networks: the policy network that determines the optimal action, and the value network evaluates the action by estimating the value function.

Huang et al. [26] implemented a combination of two models, Deep Q-Network (DQN) and Deep Deterministic Policy Gradient (DDPG) to orchestrate smart home devices, considering Heating, Ventilation, Air Conditioning (HVAC), renewables, and storage. They define the residential comfort in terms of thermal comfort and preferred operating times for individual devices.

Li et al. [27] utilized PPO to model a smart home with three device types: critical, shiftable, and controllable. Critical devices have predetermined operating periods. Shiftable devices can adjust their operation time based on tariffs. Controllable devices can be flexible or regulated. The action set in our solution is similar to that of Li et al. [27]. Like them, we define binary control variables for shiftable devices (turning the device on or off). However, while Li et al. [27] focused exclusively on optimizing electricity costs, we also optimized residential comfort.

Mbuwir et al. [28] introduced a hybrid approach that combined PPO with a rule-based control system for electric vehicle charging. The reward function relied on the outcomes of both the PPO actions and rule-based real-time controllers.

Sun et al. [29] defined a multi-agent[1] DRL based on PPO. They minimized electricity costs on a large-scale HEMS. Their smart home environments included devices, energy storage systems, and renewable energy sources. They addressed the microgrid market problem, where actions taken by each smart home HEMS involved purchasing power from or selling power to the grid.

[1] Agents are autonomous entities that perceive the environment, take actions, and learn from their interactions.

In their study, Azuatalam et al. [30] utilized PPO to optimize HVAC control in a building. Their objective was to balance thermal comfort and electricity costs. The RL model was trained using historical data, including weather conditions and solar radiation. A penalty factor was incorporated to ensure thermal comfort, penalizing deviations from the thermal comfort range of 20℃ to 25℃. In addition to thermal comfort, our solution also optimizes the operation of other smart home devices.

Zhang et al. [31] defined a multi-agent DRL approach employing PPO to optimize the airflow in an HVAC system, based on the observed state. Sensor data provided a more comprehensive insight into the observed state. This approach aimed to balance thermal comfort and electricity costs.

Our work combines the approaches of Huang et al. [26], Liu et al. [24], and Forootani et al. [25]. We considered two DRL models for our HEMS: a DDQN model, as Liu et al. [24], and a PPO model, as used by Li et al. [27], Azuatalam et al. [30], and Zhang et al. [31]. Our smart home setup is similar to Huang et al. [26], featuring air conditioning, a solar panel, shiftable devices with preferred operation times managed by HEMS, and uncontrollable devices that run on demand. We developed a scale for residential comfort based on ideas from [10] and [23], aiming to balance objectives of residential comfort – which encompassed preferred operation times and thermal comfort – and electricity costs. Our solution uniquely considers both electricity consumption and residential comfort across various desired working ranges, such as the working hours for air conditioners. Unlike previous studies, we employed real-world historical data in our simulations with the Typhoon HIL simulation platform due to its generalized observation space. Typhoon HIL allows us to simulate various smart home configurations rather than being limited to a specific case scenario. This flexibility is crucial for generalizing our findings across different household types.

3 Methodology and Implementation

The Markov Decision Process (MDP) is a mathematical representation of RL. MDP defines a goal-oriented learning approach [32], making it an ideal solution for our optimization problem of scheduling device operations to balance electricity costs and residential comfort. The MDP framework encompasses three major decision-making elements: environment, agent, and reward.

3.1 Environment

The environment used to train the DRL-based HEMS is a simulated smart home built using Typhoon HIL [12]. The environment is described in our previous works [33] [34]. The simulated smart home contains two device types: shiftable devices controlled by HEMS and uncontrollable devices $L_{uncontrollable} = \{d_1\}$ that must run on demand, regardless of external factors such as price tariffs. Shiftable devices include air conditioning, dishwasher, washing machine, and solar panel, $L_{controllable} = \{d_2, d_3, d_4, d_5\}$. HEMS can postpone or advance the operation of these devices to take advantage of the lower price tariffs or solar panel production. The residents define the preferred operation time for these devices, as detailed in Section 4.

The energy consumption of a device is calculated as:

$$E_s(t) = P_s * AS_s * t \tag{1}$$

where s represents the device, P_s denotes the amount of power a device consumes when it is operating, and the action set AS_s= [0, 1] is a binary value indicating whether the device is OFF or ON during timestep t.

Each device has predefined inputs (actions HEMS may take at the start of the time step) and outputs (observation HEMS makes at the end of the timestep). When working, the devices differ in electrical consumption patterns, corresponding to the available states and actions of the devices (Sections 3.2.2 and 3.2.3). Similar device-specific constraints as in [26] are applied.

3.2 HEMS

Our goal is to schedule the operation of devices to balance electricity costs with residential comfort. To achieve this, in each time step, HEMS can choose to turn devices on or off. Running devices during low electricity price periods reduces costs but may compromise residential comfort if the desired service is not provided on time. HEMS may also choose to run devices even during high electricity price periods if the solar panel produces enough energy. When the HEMS turns on a device, Typhoon HIL simulates the device's electricity consumption over the subsequent time steps (as shown in the example presented in Figure 2) until the device completes its task HEMS chooses to turn it off.

3.2.1 Reward

The reward function is dependent on the environment. In video games such as [35], the reward function can take discrete values of 0 (the taken action did not change the state), 1 (the taken action led to a better state), and -1 (the taken action led to a worse state). The long-term goal is to maximize the cumulative reward [32].

We used the reward function from [33], which was previously adopted from [14]:

$$reward = \left[\left(\sum_{s=1}^{S} \mathrm{benefit}_s(t)\cdot \mathrm{E}_s(t)\right) - C_{\mathrm{electricity}}(t)\cdot E_{\mathrm{grid}}(t)\right] \qquad (2)$$

where:

$\boldsymbol{S = L_{uncontrollable} + L_{controllable}}$ is the number of smart home devices.

$\boldsymbol{benefit_s}$ is the monetary benefit for having the device's service during the time interval; the monetary benefits are calculated by the *Typhoon framework* according to residential demands listed in Table 1.

$\boldsymbol{E_s}$ from equation (1), is the energy spent [kWh] for running the device during the time step. To ensure a more stable learning process, all electricity consumption values E_s were normalized to the range [0, 1]

$\boldsymbol{C_{electricity}}$ is the cost of electricity during the time step

$\boldsymbol{E_{grid}}$ is the energy provided by the grid the time interval [kWh]

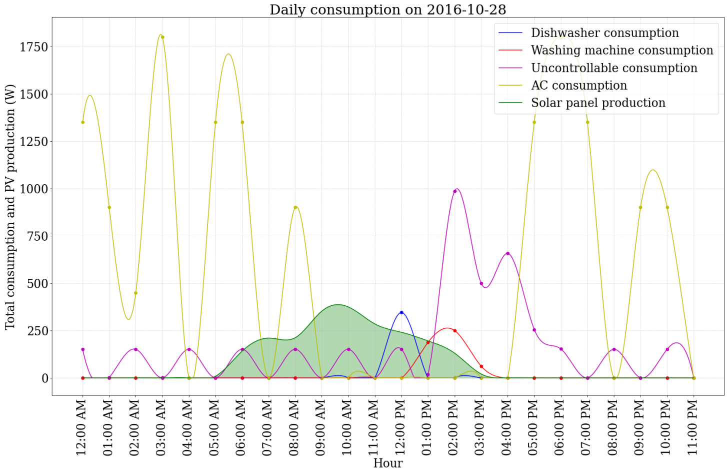


Figure 2

An example of energy consumption of smart home devices. Uncontrollable devices' consumption varies across different simulation days. Solar power production varies depending on the real-world outdoor weather conditions recorded for each day. Other devices' consumption (dishwasher, washing machine, and AC) also varies daily, based on the decisions made by HEMS.

The reward function depends on residential comfort, defined through the perceived monetary benefit of running the device during a certain time period. We used three levels of monetary benefit:

High: Residents would choose to run the device even if it is the peak electricity price period.

Medium: Residents allow that operating the device may be curtailed during this period.

Don't care: Residents do not care if the device operates during this period

Monetary values corresponding to these benefit levels are adjustable simulation parameters listed in Section 4.

When training DRL models using the reward function defined in Equation (2), we observed that it took many episodes for the reward to stabilize. The unstable reward during training led to inconsistencies in the optimal actions for the same state during the testing phase, indicating that different actions were deemed optimal for the same states. Inspired by [35], we combated this issue by adding a penalty factor at the end of an episode (day). The penalty was applied when the model failed to activate devices by the end of the day. Considering the penalty factor, the total reward is:

$$\hat{R} = \begin{cases} R - Penalty & if \text{ the } device\ was\ not\ activated \\ R & otherwise \end{cases}$$

where the $\hat{R}$ denotes the modified reward, R the original reward, and $Penalty$ a negative penalty constant. The penalty encourages the model to consider turning on the device during the day, mitigating the issue of inconsistent optimal actions.

3.2.2 State

State representation is tightly coupled with the observed environment. For reference, in video games, a state may be defined as a single game frame [35]. In the HEMS context, the environment's state is described through device outputs, the current electricity price, and the resident-defined monetary benefit of using the device in that time step. Thus, we model the state as a tuple of five discrete values:

$$s = \left(B_{AC}, B_{dishwasher}, B_{washing_{machine}}, W_{solar_{panel}}, T \right)$$

In the above equation, B_{device} denotes the resident-assigned monetary benefit of using the device in the observed time step and takes values "don't care," "medium," and "high". W_{device} denotes whether the device is operating, taking values 0 (turned off) and 1 (turned on). T denotes the electricity price in the observed timestep.

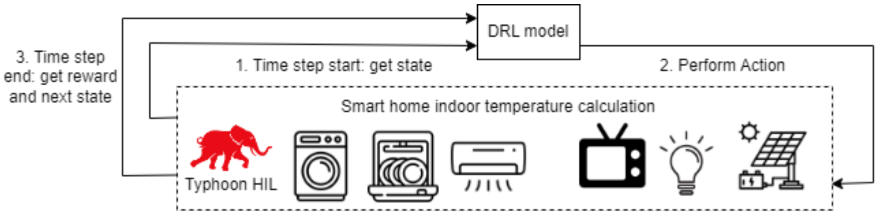
3.2.3 Actions

In each state, the DRL model chooses an action that may change the state of the environment. The DRL model aims to learn the optimal actions to take in the observed states. Optimal actions are those that maximize the cumulative reward.

In the HEMS context, each device has associated actions. Our action set comprises six discrete values $a_{set} = [0,\ 5]$. These values represent the following actions: 0 – turn on the air conditioner, 1 – turn off the air conditioner, 2 - turn on the dishwasher, 3 - turn off the dishwasher, 4 – turn on the washing machine, and 5 – turn off the washing machine. Other smart home devices, such as uncontrollable devices and the solar panel, are not controlled by the HEMS. Solar panel electricity production depends on the outdoor weather data, and the usage of uncontrollable devices is randomized throughout the day to simulate resident behavior.

3.2.4 Training the HEMS

The architecture of our solution is presented in Figure 1. The smart home environment is thoroughly described in [34]. This paper focuses on training and evaluating DDQN and PPO models in this environment. Figure 3 illustrates the training process of DDQN and PPO models. The model takes actions to "learn" the optimal action for the observed state. In return, the model receives a reward presented in Equation (2).


DRL model
3. Time step end: get reward and next state
1. Time step start: get state
2. Perform Action
Smart home indoor temperature calculation
Typhoon HIL

The architecture of the environment. The DRL model is either PPO or DDQN.

DQN leverages deep neural networks to learn optimal policies in environments where the number of states and actions is too large for traditional RL methods. The network predicts reward outcomes from specific actions within given states, helping the algorithm maximize cumulative rewards over time. Extending DQN, DDQN employs two neural networks to reduce action values overestimations - one selects actions while the other evaluates their predicted rewards. This dual-network approach enhances the stability and accuracy of training. Further details on DDQN can be found in papers [36] [37].

PPO is an advanced policy gradient algorithm distinct from value-based methods like DDQN. Unlike DDQN, which derive policy indirectly through value functions, PPO optimizes the policy directly. This direct method combined with the use of multiple data epochs per update, contribute to PPO's robustness and efficiency, making it ideal for environments with high-dimensional action spaces. More comprehensive information on PPO is available in papers [38-41].

4 Simulation

To evaluate our solution, we defined a case study with the following smart home simulation parameters: geographical location, specific time period, electricity price tariffs, and resident-defined monetary benefits, determining their preferences for device operation times. These parameters can be adjusted to simulate a smart home with differently defined residential comfort in a different geographical location.

We collected solar irradiation and temperature data for external weather conditions from the 1st to the 22nd of January, June, and October 2016. The data was collected for Berlin from Solcast [42] and divided into the training and test portions.

The electricity price tariffs ($C_{electricity}$) are representative of a typical German tariff scheme for small residential customers: the peak price is 0.4 euro/kWh lasting from 5 AM to 1 PM, the shoulder price is 0.3 euro/kWh lasting from 1 PM to 12 PM, and the off-peak electrical price is 0.2 euro/kWh lasting from 12 AM to 5 AM.

Resident-defined monetary benefit values ($benefit_s$) are based on the electricity price tariffs, as recommended in [14], and presented in Table 1. The high monetary benefit is set to be higher than the peak electricity price, implying that the device should be operated even during the peak electricity price period. The medium monetary benefit is selected to be between shoulder and peak electricity prices, implying that running the device may be curtailed during the peak pricing period.

Table 1

User-defined benefits [34]

Device	Hours	Importance	Energy consumption (Wh)
Air conditioning	5 PM-8 AM (next day) 8 AM-5 PM	High Do not care	0 – 1800
Dishwasher	6 AM-9AM 9 AM-11 AM Rest of the day	Medium High Do not care	0 – 690
Solar Panel	All day	High	Varies depending on real-world data
Uncontrollable devices	All day	High	0 – 2034
Washing Machine	1 PM-7 PM Rest of the day	High Do not care	0 – 250

The DRL models are implemented using Python 3.11 and TensorFlow 2.0 framework on a desktop with 16GB RAM and an Intel i5 processor. DDQN model's hyperparameters were set to: discount factor 0.99, episode length 96, epochs 600, exploration decay rate 5e-4, and learning rate 25e-4. PPO model's hyperparameters were: clip parameter 0.2, discount factor 0.99, episode length 96, epochs 600, learning rate (policy network) 2e-5, and learning rate (value network) 1e-4.

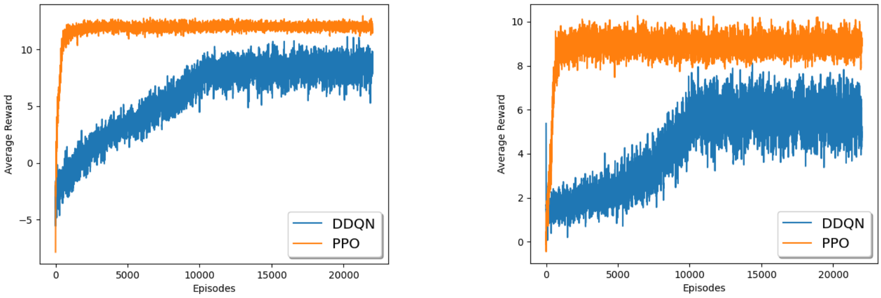
4.1 Performance of the DDQN and PPO Models

Figure 4 shows the learning curve of the DDQN and PPO training process. In each training scenario, the reward increased over the training episodes for both models. The PPO model stabilized after less than 400 episodes, while the DDQN model took more than 10000 episodes to stabilize. Figures 4 (a) and (b) show that the average

DDQN reward is lower and oscillates more after convergence than the average PPO reward. Based on the graphs, PPO is expected to perform better on the test data.

4.2 Testing Phase - Comparing the Models' Performance

To compare the trained models, we compared the reward they achieved on the test data (from the 22nd to the 29th of January, June, and October 2016). During the testing phase, DDQN and PPO models apply their version of the optimal action for the observed state. We analyzed total reward, device comfort, and thermal comfort.



a) Winter (1st to 22nd January 2016)

4 Average rewards for Winter and Summer training periods

Table 2 shows that PPO outperformed DDQN in terms of total reward in January and June and achieved the same total reward in October. In January, PPO saved 7 euros per week in electricity costs at the expense of thermal comfort. In June, despite spending 2 euros more weekly than DDQN, PPO provided better thermal and device comfort. In October, DDQN outperformed PPO in thermal comfort at the cost of using 13 euros per week more in electricity cost.

Table 2

Total weekly cost, comfort, and reward for January, June, and October. Thermal comfort refers to the air conditioning device, while device comfort refers to the washing machine and dishwasher.

	Algorithm	**Total energy cost (euro)**	**Total thermal comfort (euro)**	**Total device comfort (euro)**	**Total reward (euro)**
22nd – 29th January	DDQN	159	**195**	55	91
	PPO	**152**	189	55	**92**
22nd – 29th June	DDQN	**42**	50	53	61
	PPO	44	**55**	**55**	**66**
22nd – 29th October	DDQN	142	**175**	57	90
	PPO	**129**	162	57	90

5 Discussion

It is hard to directly compare our results to those presented in related work due to significant differences in training and testing settings, which can be summarized as follows:

(1) Different periods are used for training and testing the models
(2) Various types of smart home simulation setups are employed
(3) Training objectives are defined differently
(4) There are variations in the training setup

5.1 Training and Testing Periods

In [19], the testing extended from June to December, with the average reward beginning to decline by October. The reward was inversely proportional to electricity cost, which suggests that the decline can be attributed to outdoor weather conditions. During the winter months, frequent heating was necessary to maintain the resident comfort, leading to increased electricity costs and negatively impacting the reward [19]. Similarly, during the training phase, an issue was identified in June when analyzing temperature data; a drastic temperature variation was detected on a randomly selected day. Higher temperatures required more frequent cooling to maintain comfort [19], significantly affecting the model's convergence. For comparison, when analyzing outdoor temperatures of any day in January, the temperatures were consistently low, and the daily variation was not as extreme as in June. We addressed these seasonal impacts by dividing the training and testing data by seasons, which resulted in a more stable training curve (Figure 4).

The training set in studies [24] [26] [27] included data spanning one year. In [26], testing was conducted for one month immediately following the training period, while [24] tested the model over 50 subsequent days. The study [27] used a whole year of data for both training and testing. In contrast, the authors of [30] used only winter months. In [19], the training set included data from May to December, while testing was conducted on data from June to December of the following year. Authors in [1] trained and tested their approach using data from a single month, while the study [10] focused on a single day. In [30] and [31], the authors acknowledged the potential seasonal effect during training and testing and limited their dataset to January. In our study, we employed data from the first part of each month for training and the remainder for testing. This procedure was repeated using three distinct months to cover winter, summer, and autumn.

Like the study [1], we avoided potential dependencies between consecutive days by randomly selecting days from the first 22 days of the month for 22000 training episodes. Authors of [26] also chose random training days with single-day episodes.

5.2 Smart Home Simulation Setups

From the RL perspective, an effective simulation environment must offer both precision and speed. Typhoon HIL excels in both areas. It precisely models internal house temperatures based on external weather conditions and fine-grained variations in device electricity consumption patterns. Its ability to process data swiftly supports the rapid simulation of multiple training episodes, which is essential for efficient iterative learning and model optimization.

In contrast, other solutions used MATLAB/Simulink [1] [11] [23] and historical data [1] [19] [24] [26] to simulate the smart home or an artificial neural network model to predict environment data [10] [23], rather than using real-world data or precise simulation models. Compared to MATLAB, Typhoon HIL facilitates real-time control and monitoring of simulation parameters [43], which will be advantageous when integrating the model into a real smart home in the future.

5.3 Training Objectives

This study defined a reward function that balanced electricity costs with residential comfort, incorporating both thermal comfort and resident-defined preferred device operation times. In contrast, most studies considered only a part of this optimization goal. For example, the study [11] minimized electricity costs and assigned device priorities on a scale of 1 to 6, while the study [19] solely optimized thermal comfort.

Like in our study, the settings in studies [11] and [23] allowed residents to specify the periods during which devices should operate. If the HEMS failed to provide the desired service, a dissatisfaction cost was included in the reward function. Our study provides a more flexible setup where the residents could assign monetary benefits for running the service during certain day periods.

Study [1] focused on optimizing electricity costs and thermal comfort, disregarding other devices. They defined thermal comfort as maintaining a fixed temperature, whereas our study allows residents to specify periods when they are concerned about the house temperature and periods when they are not. Unlike our approach, the study [1], used a fixed electricity cost instead of tariffs. Study [19] considered thermal comfort and electricity costs but did not consider electricity tariffs.

Like our study, studies [24] and [25] considered electricity costs, thermal and device comfort. However, these studies define reward functions differently. Study [24] optimized thermal and device comfort, transforming the device constraints into rewards by integrating weighting factors into the total reward. Like our simulation, they used a 15-minute time step. In contrast, their setting allowed selling the electricity back to the grid and using a house battery to save the produced electricity. Study [25] imposed dissatisfaction penalties, calculating thermal dissatisfaction as the difference between the indoor and the desired temperature and device dissatisfaction as the discrepancy between selected and actual device working time.

5.4 Performance Overview

In our study, PPO converged faster and achieved a higher average reward than DDQN during training. Both DDQN and PPO were trained in the same smart home environment. Sun et al. [29] found that PPO achieved a higher average reward and converged faster than DQN. However, they used different environments for these models – DQN was trained in a discrete action space, while PPO was trained in a continuous action space, allowing an infinite number of possible actions.

We acknowledge that the resident-defined preferences for device operation times and the electricity tariff greatly influence the results presented in our study. We performed a case study to evaluate whether DDQN or PPO are viable for training HEMS to control this flexible scenario. We acknowledge that manually defining monetary benefits can introduce bias and be laborious. As a part of our future work, we aim to develop a model that learns household-specific habits, thereby alleviating the burden of manual preference settings for residents.

In [19], the indoor temperature was gathered from a homeowner to identify user preferences. The study [24] utilized an extensive real-world data set to simulate smart home devices. In contrast, our study offers the residents the flexibility to set desired temperature ranges and preferred operation times for the devices. Unlike [19], we considered electricity tariffs. Although we did not collect real-world data like [24] and [19], our approach allows homeowners to define the desired temperature for specific periods of the day.

Several potential limitations exist when using DRL in smart home energy management. Firstly, training DRL models demands large amounts of high-quality data, which may not always be available consistently [44]. Real-time decision-making may be challenging, as the computational demands of DRL could hinder its application in scenarios requiring immediate responses.

DRL models can struggle and may require retraining when the environment changes, such as shifts in user behavior, and fluctuations in electricity prices and weather conditions that affect consumption patterns [45]. Moreover, DRL models are highly dependent on initial conditions and training data, with variations in these leading to different outcomes. This sensitivity can affect the model’s reliability and performance. A possible solution for addressing variable electricity prices could be to implement a separate model trained to predict patterns of changes in electricity prices. The model’s predictions could then be incorporated into the RL model’s state.

The scalability and complexity of the model also pose significant challenges as the number of devices and the complexity of tasks increase, potentially making the training process more computationally expensive and time-consuming [46].

Finally, security, privacy, and ethical considerations may present an issue, especially regarding user autonomy and consent in data collection and use for training models [47].

Conclusions

This paper addresses the challenge of orchestrating smart home devices, in order to balance residential comfort with electricity costs. We proposed employing DRL models (DDQN and PPO) to achieve this objective. These models were trained using a smart home simulation, implemented via the Typhoon HIL and Mainflux IoT platforms. Typhoon HIL served as a reliable toolchain for modelling smart homes that closely replicate real-world environments. Mainflux IoT supplied real-world weather data to Typhoon HIL. This setup enabled accurate smart home simulations, thus, allowing the models to explore various device management strategies in a rapid manner.

We compared the performance of trained DDQN and PPO models concerning thermal and device comfort and electricity costs across different seasons. While both algorithms successfully reduced electricity costs, PPO outperformed DDQN in terms of thermal comfort and electricity cost savings during the colder seasons. In the warmer seasons, PPO outperformed DDQN in terms of device comfort.

Our future goal is to develop a self-sustaining energy management system, aiming to achieve a zero-energy consumption residence, that operates independently, like an off-grid system. This system will harness energy from various renewable energy sources, such as solar panels, wind, geothermal and will have the capability to store any surplus power in electric power banks, including batteries and electric vehicles. Additionally, owners could sell any excess power back to the distribution grid.

Acknowledgments

This research has been supported by the Ministry of Science, Technological Development and Innovation (Contract No. 451-03-65/2024-03/200156) and the Faculty of Technical Sciences, University of Novi Sad through project “Scientific and Artistic Research Work of Researchers in Teaching and Associate Positions at the Faculty of Technical Sciences, University of Novi Sad” (No. 01-3394/1), and Typhoon HIL, Inc. (www.typhoon-hil.com), who provided the equipment and infrastructure for the project.

References

[1] A. Gupta, Y. Badr, A. Negahban, and R. G. Qiu, “Energy-efficient heating control for smart buildings with deep reinforcement learning,” *J. Build. Eng.*, vol. 34, p. 101739, Feb. 2021, doi: 10.1016/j.jobe.2020.101739.

[2] A. Chehri and H. T. Mouftah, “FEMAN: Fuzzy-Based Energy Management System for Green Houses Using Hybrid Grid Solar Power,” *J. Renew. Energy*, vol. 2013, p. e785636, Sep. 2013, doi: 10.1155/2013/785636.

[3] C. Wan, J. Zhao, Y. Song, Z. Xu, J. Lin, and Z. Hu, “Photovoltaic and solar power forecasting for smart grid energy management,” *CSEE J. Power Energy Syst.*, vol. 1, no. 4, pp. 38–46, Dec. 2015, doi: 10.17775/CSEEJPES.2015.00046.

[4] K. Kostková, Ľ. Omelina, P. Kyčina, and P. Jamrich, “An introduction to load management,” Electr. *Power Syst. Res.*, vol. 95, pp. 184–191, Feb. 2013, doi: 10.1016/j.epsr.2012.09.006.

[5] H. L. Zhang, J. Baeyens, J. Degrève, and G. Cacères, “Concentrated solar power plants: Review and design methodology,” *Renew. Sustain. Energy Rev.*, vol. 22, pp. 466–481, Jun. 2013, doi: 10.1016/j.rser.2013.01.032.

[6] D. S. Kirschen, G. Strbac, P. Cumperayot, and D. de Paiva Mendes, “Factoring the elasticity of demand in electricity prices,” *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 612–617, May 2000, doi: 10.1109/59.867149.

[7] A. Barbato, L. Borsani, A. Capone, and S. Melzi, “Home energy saving through a user profiling system based on wireless sensors,” in *Proceedings of the First ACM* Workshop *on Embedded Sensing Systems for Energy-Efficiency in Buildings*, Berkeley California: ACM, Nov. 2009, pp. 49–54. doi: 10.1145/1810279.1810291.

[8] D. Li and S. K. Jayaweera, “Reinforcement learning aided smart-home decision-making in an interactive smart grid,” in *2014 IEEE Green Energy and Systems Conference (IGESC)*, Nov. 2014, pp. 1–6. doi: 10.1109/IGESC.2014.7018632.

[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, doi: 10.1109/TSG.2020.2971427.

[10] R. Lu, S. H. Hong, and M. Yu, “Demand Response for Home Energy Management Using Reinforcement Learning and Artificial Neural Network,” *IEEE Trans. Smart Grid*, vol. 10, no. 6, pp. 6629–6639, Nov. 2019, doi: 10.1109/TSG.2019.2909266.

[11] 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, doi: 10.1109/ACCESS.2020.2974286.

[12] “Home Page,” Typhoon HIL. Accessed: Jun. 02, 2023. [Online]. Available: https://www.typhoon-hil.com/

[13] “Mainflux — Full-stack Open-Source, Patent-free IoT Platform and Consulting services,” Mainflux Labs. Accessed: Jun. 02, 2023. [Online]. Available: https://www.mainflux.com/index.html

[14] Pedrasa, Michael Angelo, Ted Spooner, and Iain MacGill. "An energy service decision-support tool for optimal energy services acquisition." Centre for Energy and Environmental Markets, UNSW, the university of new south wales (2010).

[15] D. Majstorovic, I. Celanovic, N. Dj. Teslic, N. Celanovic, and V. A. Katic, "Ultralow-Latency Hardware-in-the-Loop Platform for Rapid Validation of Power Electronics Designs," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4708–4716, Oct. 2011, doi: 10.1109/TIE.2011.2112318.

[16] C. R. D. Osório, M. Miletic, J. Zelic, D. Majstorovic, and O. Gagrica, "Advancements on Real-Time Simulation for High Switching Frequency Power Electronics Applications (Invited Paper)," in *2021 21st International Symposium on Power Electronics (Ee)*, Oct. 2021, pp. 1–6. doi: 10.1109/Ee53374.2021.9628306.

[17] H. E. Toosi, A. Merabet, A. M. Y. M. Ghias, and A. Swingler, "Central Power Management System for Hybrid PV/Battery AC-Bus Microgrid Using Typhoon HIL," in *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, Jun. 2019, pp. 1053–1058. doi: 10.1109/ISIE.2019.8781277.

[18] M. Mohammadi, A. Al-Fuqaha, M. Guizani, and J.-S. Oh, "Semisupervised Deep Reinforcement Learning in Support of IoT and Smart City Services," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 624–635, Apr. 2018, doi: 10.1109/JIOT.2017.2712560.

[19] P. Lissa, C. Deane, M. Schukat, F. Seri, M. Keane, and E. Barrett, "Deep reinforcement learning for home energy management system control," *Energy AI*, vol. 3, p. 100043, Mar. 2021, doi: 10.1016/j.egyai.2020.100043.

[20] F. Ruelens, B. J. Claessens, S. Vandael, B. De Schutter, R. Babuška, and R. Belmans, "Residential Demand Response of Thermostatically Controlled Loads Using Batch Reinforcement Learning," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2149–2159, Sep. 2017, doi: 10.1109/TSG.2016.2517211.

[21] A. Prasad and I. Dusparic, "Multi-agent Deep Reinforcement Learning for Zero Energy Communities," in *2019 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*, Sep. 2019, pp. 1–5. doi: 10.1109/ISGTEurope.2019.8905628.

[22] E. U. Haq, C. Lyu, P. Xie, S. Yan, F. Ahmad, and Y. Jia, "Implementation of home energy management system based on reinforcement learning," *Energy Rep.*, vol. 8, pp. 560–566, Apr. 2022, doi: 10.1016/j.egyr.2021.11.170.

[23] S. Lee and D.-H. Choi, "Reinforcement Learning-Based Energy Management of Smart Home with Rooftop Solar Photovoltaic System, Energy Storage System, and Home Appliances," *Sensors*, vol. 19, no. 18, p. 3937, Sep. 2019, doi: 10.3390/s19183937.

[24] Y. Liu, D. Zhang, and H. B. Gooi, “Optimization strategy based on deep reinforcement learning for home energy management,” *CSEE J. Power Energy Syst.*, vol. 6, no. 3, pp. 572–582, Sep. 2020, doi: 10.17775/CSEEJPES.2019.02890.

[25] A. Forootani, M. Rastegar, and M. Jooshaki, “An Advanced Satisfaction-Based Home Energy Management System Using Deep Reinforcement Learning,” *IEEE Access*, vol. 10, pp. 47896–47905, 2022, doi: 10.1109/ACCESS.2022.3172327.

[26] C. Huang, H. Zhang, L. Wang, X. Luo, and Y. Song, “Mixed Deep Reinforcement Learning Considering Discrete-continuous Hybrid Action Space for Smart Home Energy Management,” *J. Mod. Power Syst. Clean Energy*, vol. 10, no. 3, pp. 743–754, May 2022, doi: 10.35833/MPCE.2021.000394.

[27] H. Li, Z. Wan, and H. He, “A Deep Reinforcement Learning Based Approach for Home Energy Management System,” in *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, Washington, DC, USA: IEEE, Feb. 2020, pp. 1–5. doi: 10.1109/ISGT45199.2020.9087647.

[28] B. V. Mbuwir, L. Vanmunster, K. Thoelen, and G. Deconinck, “A hybrid policy gradient and rule-based control framework for electric vehicle charging,” *Energy AI*, vol. 4, p. 100059, Jun. 2021, doi: 10.1016/j.egyai.2021.100059.

[29] J. Sun, Y. Zheng, J. Hao, Z. Meng, and Y. Liu, “Continuous Multiagent Control Using Collective Behavior Entropy for Large-Scale Home Energy Management,” *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 01, pp. 922–929, Apr. 2020, doi: 10.1609/aaai.v34i01.5439.

[30] D. Azuatalam, W.-L. Lee, F. de Nijs, and A. Liebman, “Reinforcement learning for whole-building HVAC control and demand response,” *Energy AI*, vol. 2, p. 100020, Nov. 2020, doi: 10.1016/j.egyai.2020.100020.

[31] T. Zhang, A. K. G. S, M. Afshari, P. Musilek, M. E. Taylor, and O. Ardakanian, “Diversity for transfer in learning-based control of buildings,” in *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*, Virtual Event: ACM, Jun. 2022, pp. 556–564. doi: 10.1145/3538637.3539615.

[32] S. Thrun and M. L. Littman, “A Review of Reinforcement Learning,” *AI Mag.*, vol. 21, no. 1, Art. no. 1, Mar. 2000, doi: 10.1609/aimag.v21i1.1501.

[33] A. Kaplar, F. Savić, A. Kaplar, J. Vidaković, J. Slivka, and M. Vidaković. “Integration of Mainflux platform into a Multi-Agent based HEMS framework.” (2021): 168-172.

[34] A. Aleksić, M. Vidaković, J. Slivka, B. Milosavljević and A. Kaplar, 2020. Multi-Agent based HEMS framework. ICIST, pp.84-88.

[35] V. Mnih *et al.*, “Playing Atari with Deep Reinforcement Learning,” Dec. 19, 2013, *arXiv*: arXiv:1312.5602. doi: 10.48550/arXiv.1312.5602.

[36] Q. Fu, Z. Han, J. Chen, Y. Lu, H. Wu, and Y. Wang, “Applications of reinforcement learning for building energy efficiency control: A review,” *J. Build. Eng.*, vol. 50, p. 104165, Jun. 2022, doi: 10.1016/j.jobe.2022.104165.

[37] H. Van Hasselt, A. Guez, and D. Silver. “Deep reinforcement learning with double q-learning.” In Proceedings of the AAAI conference on artificial intelligence, vol. 30, no. 1. 2016.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” Aug. 28, 2017, *arXiv*: arXiv:1707.06347. doi: 10.48550/arXiv.1707.06347.

[39] R. S. Sutton and A. G. Barto, *Reinforcement learning: an introduction*. in Adaptive computation and machine learning. Cambridge, Mass: MIT Press, 1998.

[40] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz. “Trust region policy optimization.” In International conference on machine learning, pp. 1889-1897. PMLR, 2015.

[41] “Proximal Policy Optimization — Spinning Up documentation.” Accessed: Jun. 04, 2023. [Online]. Available: https://spinningup.openai.com/en/latest/algorithms/ppo.html#id8

[42] “Solcast \textbar Solar Api and Solar Weather Forecasting Tool.” Accessed: Jun. 02, 2023. [Online]. Available: https://solcast.com.au

[43] V. P. Chandran, S. Kumar, and P. S. Bhakar, “Comparative Study for Steady-State Operation of IAG in Stand-Alone mode using MATLAB and Typhoon HIL,” in *2018 3rd International Conference On Internet of Things: Smart Innovation and Usages (IoT-SIU)*, Bhimtal: IEEE, Feb. 2018, pp. 1–5. doi: 10.1109/IoT-SIU.2018.8519878.

[44] N. Parvez Farazi, B. Zou, T. Ahamed, and L. Barua, “Deep reinforcement learning in transportation research: A review,” *Transp. Res. Interdiscip. Perspect.*, vol. 11, p. 100425, Sep. 2021, doi: 10.1016/j.trip.2021.100425.

[45] Zhang, Junjie, Cong Zhang, and Wei-Che Chien. “Overview of deep reinforcement learning improvements and applications.” Journal of Internet Technology 22, no. 2 (2021): 239-255..

[46] Feriani, Amal, and Ekram Hossain. “Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial.” IEEE Communications Surveys & Tutorials 23, no. 2 (2021): 1226-1252.

[47] T. T. Nguyen and V. J. Reddi, “Deep Reinforcement Learning for Cyber Security,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 3779–3795, Aug. 2023, doi: 10.1109/TNNLS.2021.3121870.