GENG4412/5512 Engineering Research Project

Final Report Preparation Guide
Semester 1, 2025

Dhrumil Bhanderi
24047059
School of Engineering, University of Western Australia

Supervisor: Hesam Mohammadi
School of Engineering, University of Western Australia

Co-Supervisor: Dr. Victor Shreeram
School of Engineering, University of Western Australia

Word Count - 6203

DECLARATION OF CONTRIBUTION

This report was successfully completed by using a number of peer-reviewed studies that were first supervised by the supervisor, Mr. Hesam Mohammadi, as well as those that were obtained through independent research. Every time content from these studies is used, the original authors are credited with the relevant in-text citations and references.

My contribution

Contributions the author believes to be original content include:

· Using general IEEE published standards, the author developed an algorithm for scheduling appliances using a machine learning approach;

· Choosing the right renewable source, battery type, and simulation in a Python environment; visualizing solar data, battery storage system, and appliance usage timing;

· Creating a detailed techno-economic model for battery system, PV cell, and Synergy electricity price using Python programming in Visual Studio code (based on reward function information obtained through multiple peer-reviewed research studies).

Use of AI tools

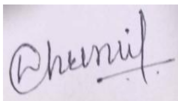
Students are permitted to use AI tools for general research purposes and to check and improve the quality of written English in their report. Students are not permitted to use AI tools to generate content.

In this section you must provide details of how you have used AI tools in your project and/or in the preparation of assessment items. If you did not use AI, then state this.

Studentdeclaration

In accordance with University Policy, I certify that:

The above information is correct, and the attached work submitted for assessment is my own work and that all material drawn from other sources has been fully acknowledged and referenced.

Student signature  Date 18/05/2025

Supervisor confirmation

Supervisor to indicate agreement or clarification of the comments above.

Supervisor to enter comments here (e.g., The above declaration of student contribution is correct.)

Supervisor signature Hesam mohammad: Date 18/05/2025

Project Summary

The rapid growth of energy demand and environmental concerns has underscored the importance of an efficient Building Energy Scheduling System (BESS). This report presents a novel data-driven framework for optimizing building energy management by addressing critical gaps in existing research, including limited real-world deployment, poor generalization, and inefficient training processes. In fact, this report proposes an approach applying reinforcement learning (RL) to optimize real-time energy scheduling for an integrated renewable-based smart building, considering solar and wind energies coupled with a real-time battery storage system as a backup and a time-of-use (TOU) tariff in a smart building.

This study will reflect the design, development, and compilation of a Q-learning algorithm within the Markov Decision Process (MDP) framework to schedule battery storage and appliance run time regarding cost savings per user. In this approach, agents work in the environment to constantly adapt and learn based on giving points. The feedback has rewards and punishments as positive and negative indices. For this project, building appliances are divided into three categories named shiftable, non-shiftable, and fixed load. The environment includes these three types of demands: battery storage and renewable-based resources' scheduling. Real-time renewable energy data from the next 15 days (solar and wind) will be used as an input for Q-learning, ensuring robustness against uncertainties. The reward function considers the prosumers costs and revenues. After getting results and values from rewards benchmarks, a comparison is inevitable, which indicates how the proposed model, in comparison to traditional buildings, is efficient. The future work can be building energy consumption prediction models for sustainable design and accurate energy management.

Acknowledgements

I would like to express my sincere gratitude to my dedicated thesis supervisor, Mr. Hesam Mohammadi, whose unwavering guidance and expertise have been invaluable throughout this research journey. Your mentorship has played a pivotal role in the success of this study.

List of Figures

List of Tables

Nomenclature

PV	Photovoltaic Cell
R-BAS	Retrofit-enabling Building Automation System
AI	Artificial Intelligence
LSTM	Long Short-term Memory
GRU	Gated Recurrent Unit
PICP	Prediction Interval Coverage Probability
MPIW	Mean Prediction Interval Width
AEMO	Australian Energy Market Operator
ARENA	Australian Renewable Energy Agency
STLF	Short term Load Forecasting
UHM	University of Hawaii
IoT	Internet of Things
RL	Reinforcement Learning

3. Introduction, Literature Review and Project Objectives

3.1 Introduction

Global energy demand, coupled with increasing environmental concerns, has highlighted the need for efficient and sustainable energy management in buildings. The project focuses on solving a specific problem: maximizing real-time energy planning for smart buildings that use renewable energy. Many current energy management systems in buildings have some disadvantages, such as limited practicality, inadequate generalization in different environments, and dependency on inefficient training methods. The systemincludes18 PV cells, battery storage devices, and a time-of-use electricity price (TOU). One of the core elements of the system is Q learning algorithms integrated into the Markov Decision Process (MDP) architecture. The design of MDP regulates the operating timetable for energy storage and equipment to reduce energy costs and improve efficiency.
The appliances of the building are divided into three categories: movement, non-moving, and fixed loads, each with different operational limitations. The RL agent interacts with the environment characterized by these appliances, real-time renewable energy supply, and battery status. The learning process is triggered by reward functions that balance energy costs and potential incomes for prosumers — people who consume and produce energy. Using real-time solar and wind data over 15 days to simulate real and dynamic weather conditions, the system will improve resilience to uncertainty related to renewable energy sources.

This research is important for various stakeholders. It contributes to the development of adaptive control strategies that improve the efficiency of intelligent energy systems for scientific and technical fields. This gives utility and supply companies such as Synergy, Horizon Power, AGL Perth Energy and legislators important information about how to integrate a prosumer-based energy model and how it affects energy demand management. This research work energy planning to save energy and increase the reliability of construction companies and customers. The project aims to compare traditional energy systems in buildings with proposed Reinforcement Learning - based models to demonstrate cost reduction, energy efficiency, and adaptability advantages. Future research may use these studies to integrate predictive models for building energy use and further improve sustainable design and operational planning strategies.

To reduce the cost of power for household users, this research suggests an optimal scheduling model for home energy management that considers dispatchable load and battery working status limitations.

To improve the efficiency of solving the optimization problem, a new off-policy RL-based algorithm is then used [3].

1) With dispatchable load and energy storage system operating status limitations, this paper suggests an optimal scheduling model for residential energy management that accounts for time-of-use pricing and real-time energy storage system control. The scheduling strategy can lower user electricity costs and energy storage system action times while maintaining battery safety by controlling the charging and discharging behaviour of the system based on the time-of-use electricity price and real-time energy storage system charging status [3].

2) The applied RL technique known as Q-learning is used to address the HEMS optimization problem, which is modelled as a Markov decision process (MDP). System state data is used for agent training in the data-driven proposed RL-based solution for HEMS. This method produces a more dependable energy management policy by doing away with the necessity for system modelling. This report's remaining sections are arranged as follows: The project process discusses the numerical model of the built-in home energy system [3].

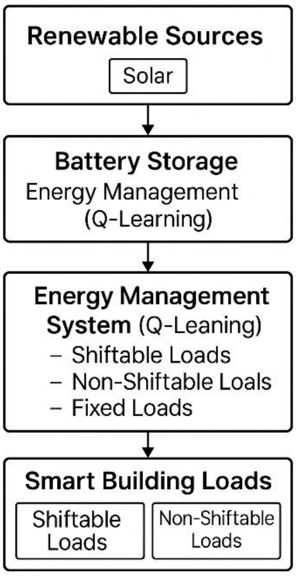


Figure 1 Energy management framework for smart building

3.2 Literature Review

3.2.1 Related Works

Home energy management systems (HEMS) have reached new heights in the demand response market. This is due to the increased demand for electricity and the development of smart grids. Home Energy Management Systems (HEMS) are tools consumers can use to change or lower their energy needs and improve how their home uses and makes energy [1]. A prominent direction of research involves the application of Artificial Intelligence (AI) and Reinforcement Learning (RL) to formulate demand-side management (DSM) strategies. In [2], a hybrid AI methodology was introduced that balances the trade-off between user comfort and energy costs. By leveraging a modified Elitist Non-dominated Sorting Genetic Algorithm II, the system optimized scheduling in the presence of battery storage, appliance priority, and variable pricing. This approach provided a multi-criteria decision-making model tailored for smart homes. Paper [3] further advanced this field by modelling the HEMS as a Markov Decision Process (MDP) and solving it using a Soft Actor-Critic (SAC) Deep Reinforcement Learning algorithm. Notably, the system demonstrated superior cost reduction—up to 2.91 Yuan—compared to baseline methods, and successfully converged after 8000 training episodes. This supports the feasibility of RL-based solutions in real-time operational settings. Supervised learning has also been explored as an effective strategy. In [4], a real-time scheduling framework based on supervised learning was implemented, considering solar PV, electric storage systems (ESS), and electric vehicles (EVs). This framework allowed the smart home to act as a prosumer, thereby actively participating in energy markets while optimizing internal energy efficiency. Deep reinforcement learning (DRL) continues to show promise across varied scenarios. For example, [5] proposed a hybrid model combining Q-learning with neural networks to control energy storage while ensuring user comfort. Each appliance (e.g., water management, air conditioning, and oven) was represented as an independent RL agent trained to maximize rewards. The multi-agent setup proved flexible in coordinating different household systems simultaneously. In [6], a Partially Observable MDP (POMDP) framework was used in a decentralized multi-agent DRL setup to not only manage DSM but also enable peer-to-peer (P2P) energy trading among homes. This represents a forward-looking approach as future smart communities may rely on such decentralized architectures for autonomous energy exchange and local grid stability.

The benefits of deep learning integration into RL were further highlighted in [7], which developed a hybrid DRL system capable of handling both single homes and aggregated buildings. The study confirmed that a properly trained single agent could manage complex energy optimization tasks in both isolated and collective environments. From a broader energy policy and sustainability standpoint, the effectiveness of DSM strategies was empirically validated in [8], where energy cost and peak load reductions reached 20% and 24% respectively. This work advocated for applying RL-based DSM beyond residential loads to upstream infrastructure such as distributed energy resources (DERs). Lastly, [9] addressed coordinated "source-load" energy management in coastal residential areas. A Q-learning algorithm was employed to minimize multi-objective functions including power balance, load fluctuation, and peak regulation costs. Simulation results confirmed the model's ability to stabilize energy supply while lowering operational expenses.

Table 1 : Summary of the related works and the recent work on machine learning based approach for smart home

No	Author	Year	Title	Focus Area	Techniques/Method	Learning Type	Performance / Outcomes
1	Helder Rocha	2021	An Artificial Intelligence based scheduling algorithm for demand-side energy management in Smart Homes	DSM using multi-objective optimization with comfort & cost tradeoff	NSGA-II + Support Vector Regression + K-means	Supervised & Optimization	51.4% energy cost reduction
2	Shengtao Xiong	2024	DRL-based HEMS considering ToU pricing and real-time control of ESS	Home load dispatching with ToU pricing & ESS coordination	Soft Actor-Critic (SAC) + MDP	Deep Reinforcement Learning	Cost reduction ≥15.87%
3	Truong Hoang	2023	Real-time energy scheduling for HEMS with ESS and EV using Supervised Learning	Energy demand forecasting + ESS/EV scheduling	Deep Neural Network (DNN) + MILP	Supervised Learning	Improved cost efficiency vs DRL-based methods
4	Sami Ben Slama	2023	Deep Learning model for intelligent	Cost-comfort multi-	Q-learning + Neural Networks	Reinforcement Learning	20% monthly cost

			HEMS using PV	objective scheduling			reduction vs ILP
5	Jiatong Wang	2023	Energy trading and load scheduling in P2P market	Decentralized DSM and energy trading	Multi-agent DRL + POMDP	Deep Reinforcement Learning	8.52–14.03% average reward gain over benchmarks
6	Elena Mocanu	2019	Online building energy optimization using DRL	Online DSM with real-time feedback to users	Deep Q-Learning + Deep Policy Gradient	Deep Reinforcement Learning	Validated on Pecan Street Inc. dataset
7	Aras Sheikhi	2016	DSM in multi-energy system for residential customers	Energy scheduling in electricity-gas systems	Reinforcement Learning	Reinforcement Learning	Up to 24% peak reduction, 20% cost savings
8	Chen Lingmin	2023	Q-learning for CCHP-based peak load control	Peak load control in coastal residential areas	Q-learning	Reinforcement Learning	Improved load stability & system economy
9	Hasan Izmitligil	2024	Online HEMS using Q-learning and DQL	Real-time smart appliance scheduling	Q-Learning + Deep Q-Learning	Reinforcement Learning	DQL better at peak reduction and cost savings
10	Zhe Wang, Tianzhen Hong	2020	Reinforcement learning for building controls: The opportunities	Comprehensive survey on RL for building control systems	Q-Learning, Deep Q-Learning, Actor-Critic,	Reinforcement Learning (review study)	Summarized 77 studies; discussed gaps, transfer learning,

			and challenges		Policy Gradient		multi-agent RL, policy/value -based methods
11	Muhammad Diyan et al.	2020	A Multi-Objective Approach for Optimal Energy Management in Smart Home Using RL	Human-appliance interaction-based scheduling for smart homes	Q-Learning with MDP, agent-based modeling	Reinforcement Learning	Reduced energy cost and discomfort vs Least Slack Time method

3.2.2 Gaps

Table 2 : Technical gaps in recent studies

Gap ID	Technical Limitation	Description	Sources
1	Simulation-Only Validation	Lack of empirical validation on physical systems limits the generalizability of simulation-based findings.	Wang & Hong (2020), Mocanu (2019)
2	Absence of Transfer Learning	Current RL models are not transferable across different building environments or user profiles.	Wang & Hong (2020)
3	Neglect of Real-Time Feedback	Most systems fail to integrate continuous comfort or behavioral feedback into control loops.	Diyan et al. (2020)

4	Underdeveloped Multi-Agent Frameworks	Limited exploration of multi-agent reinforcement learning for decentralized decision-making.	Wang et al. (2020), Mocanu (2019)
5	Scalability Issues	Existing models do not scale efficiently across varied home configurations or heterogeneous appliances.	Chen (2023), Mocanu (2019)
6	High Computational Overhead	Deep RL models are computationally intensive, hindering their deployment on edge devices.	Diyan et al. (2020)
7	Insufficient Multi-Objective Optimization	Few studies adequately balance multiple objectives such as cost, comfort, and environmental impact.	Slama (2023), Diyan et al. (2020)
8	Lack of Benchmarking Standards	No universal datasets or simulation environments exist for evaluating RL-based HEMS solutions.	Wang & Hong (2020)

3.3 Project Objectives

3.3.1 Objectives

The primary objective of this research project is to use machine learning-based algorithm and control to combine renewable source into building energy scheduling system. The initial goal is collected and examine data on solar irradiance, temperature, and daily sunshine availability that are unique to the location of the selected residential building.

The development of predictive models for solar power generation necessitates the utilization of relevant data. Subsequently, reinforcement learning techniques, particularly the Q-learning algorithm, are employed to intelligently schedule household appliances, taking into consideration user comfort, electricity costs, and energy availability.

The model organized to adapt in real-time, renewable availability appliance and battery backup. In addition, the research work establishes a robust data architecture that effectively stores and manages energy-related information, integrating CSV data storage.

Simulations using Python packages like NumPy, SciPy, and pandas and matplotlib done to generate graphs and visualization. This research conducted with real-time scenarios, such as temperatures fluctuation and appliance ON/OFF timing, to ensure the framework's accuracy. A comprehensive master's thesis is founded on the detailed recording of all methods, findings, and insights, contributing to both academic research and practical implementation of smart home energy management systems.

3.3.2 Key Performance Measures

Table 3: Key Performance Measures

No.	Measures	Outcome
1	**Electricity cost reduction (%)**	Illustrates the real-time optimization of Q-learning for appliance scheduling and battery charge/discharge.
2	**System Efficiency (%)**	The percentage of energy consumption that comes from solar panels. Which indicates less dependency on the power coming from synergy and effective use of solar PV.
3	**Reward Functions**	Analyses the ML model rate such as learning, discount and reward.
4	**Accuracy of Prediction**	Measures and save data from “tutiempo” website ,the accuracy of forecasting energy demand and renewable availability.
5	**Efficiency of Battery Usage**	Assesses how effectively the battery charges/discharges in sync with TOU pricing and usage. Indicates smart storage integration.
6	**Rate of Renewable Utilization**	Percentage of total demand met directly by solar PV. Important for evaluating both environmental and economic impacts.
7	**Index of System Robustness (15-day simulation)**	Assesses performance stability under fluctuating demand and solar input. Based on variations in cost and efficiency over time.
8	**Runtime Conflicts Resolved (Appliances)**	Measures system's ability to resolve appliance schedule conflicts while respecting constraints. Shows scheduling effectiveness.

4. Project Process

4.2 Modelling Investigations

This section provides information about design and implementation of Building Energy Scheduling System particularly residential using next 15 days real time solar irradiance data, Q-learning based reinforcement learning, home appliance ON/OFF schedule and battery management system. The goal is to optimize appliance power management and battery storage characteristics to reduce electricity price per bill cycle under various Time of Usage while coordinating with real time PV cell data.

4.2.1 Model derivation and Configuration

Markov Decision Process is a framework that includes mathematical equations and architecture for decision making where part of it is random and it depends on decision maker. MDPs have various usage in various field such as robotics, Automated control, Economics and Manufacturing. Below example shows how reinforcement learning works in real world,

The software that manages a robot could be the agent, the real world serves as the environment in this instance and the agent uses a variety of sensors including touch and video sensors to investigate it one of its behaviours is to activate signalling motors when it gets closer to its goal it might be set up to receive positive incentives if it wastes time or goes in the wrong direction it might be set up to receive negative incentives[10].

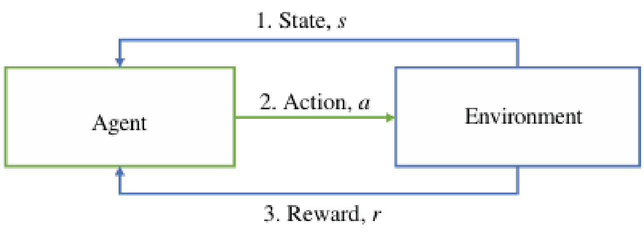


Figure 2 Reinforcement feedback loop[11]

Figure 2 depicts close loop that describes a typical RL process: The agent works in accordance with the policy $\pi(a|s)$ after observing the environment's current system states. The system then moves on to the following state, s′, and the agent is rewarded with r. The agent's to train the policy, experiences (s,a,r,s′) are used [11].

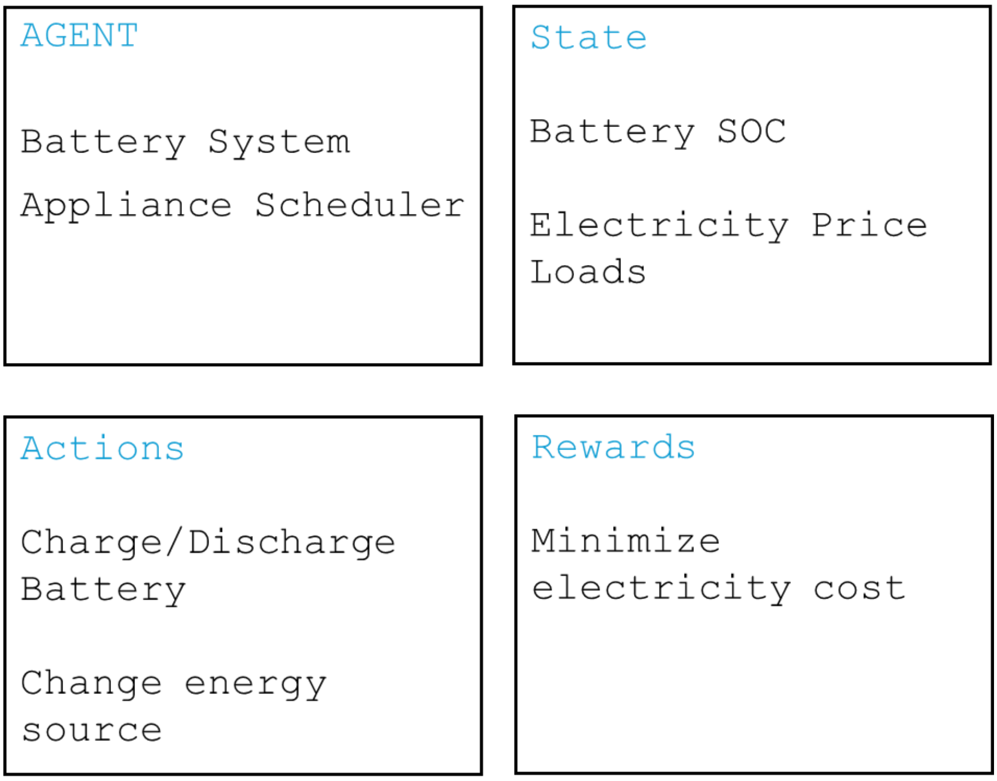


Figure 3 Key component of Reinforcement Learning for smart building

4.2.2 Software Tools and Implementation

The following software, packages and tools are used to achieve objectives:

- Python 3.13.2 - Primary programming language
- NumPy – Handles numerical patterns
- Pandas – Manage structured data like tables
- Matplotlib – Create plots and visualizations
- Random – generates random numbers or choices
- Custom python scripts – Developed for solar irradiance equation, Q learning, battery SOC model and appliance ON/OFF scheduling

Reason of selecting Python : It is a very popular with rich packages and vast machine learning libraries. In addition, it allows to manage and handle time series data, for instance, solar power, wind speed, temperature. Also, visualization gives extra attention to create boxplot, scatterplot and histogram for comparison, relation and distribution respectively. Active community support such as GeekforGeeks, PySlackers and Python Discord help in documentation and solutions. Reinforcement learning implemented to get better control over Q-learning algorithm and process.

4.2.3 Input Model Parameters

- Location: 39 Gosford Meander, Ashby WA 6065
- Latitude: 31°44′19″S, Longitude: 115°47′43″E
- Solar irradiance data: 15 days real dataset from tutiempo website
- Battery type: Tesla Power Wall 2 (13.5 kWh capacity)
- Load: Residential appliance with categorized by following:

Table 4 : Classification of selected house based on feature and operating hours

Loads	Appliance	Ratings	ON/OFF Time	Notes
Shiftable	Washing Machine	2 kW	5 PM to 9 PM	2 cycles
	Dishwasher	2 kW	9 PM to 10 PM	Everyday
	Sweeping Robot	0.05 kW	8 AM to 9 AM	Everyday
Non-Shiftable	Lights	0.275 kW	4 AM to 7 AM	5 lights
			6 PM to 10 PM	14–15 lights
			10 PM to 11 PM	2 lights
	TV	0.05 kW	8 PM to 10 PM	Everyday
	Microwave	1 kW	1 hr (no fixed time)	-
	Oven	2.3 kW	30 min	15–20 mins for each person
	Toaster	1 kW	30 min	-
	Kettle	1 kW	30 min	-
Fixed	Air Conditioner	10 kW	1-2 hrs	-
	Refrigerator	0.28 kW	24/7	-

Additional input parameters;

Table 5: Synergy Home Plan A1 price per unit

No.	Timing	Cents per Unit
1	Peak – 3pm to 9 pm	52.5313

2	Off Peak – 9pm to 9am	23.1138
3	Super off Peak – 9am to 3pm	8.4050

The study conducted for a residential building with an average daily energy consumption ranging between 22 to 26 kWh, based on real data. The machine learning scheduling model implemented in Hp pavilion –15cs3006tx laptop equipped with 1.19 GHz processor.

4.2.4 Model testing and tuning

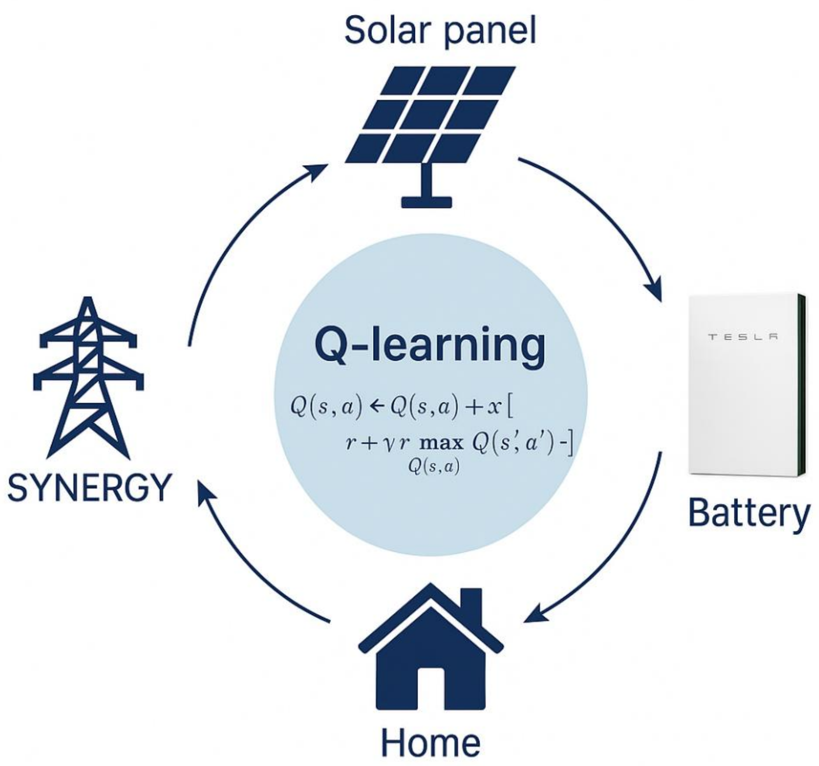


Figure 4 Q-learning cycle integrates Solar, Battery, Home Appliance and Synergy power

A series of debugging of python code conducted to ensures the Q-learning-based energy management model's convergence, stability, and performance. The proposed method used to clarify reinforcement learning model to practical scheduling objectives, like maximizing solar energy utilization, Tesla battery backup and lowering per unit costs, and accommodating customer comfort preferences.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right] \text{......................(1)}$$

Notation:

$Q(s_t, a_t)$: Current Q-value for acting a_t in state's$_t$

α: Learning rate ($0 < \alpha \leq 1$)

r_{t+1}: Reward received after acting a_t

γ: Discount factor ($0 \leq \gamma < 1$)

$\max_a Q(s_{t+1}, a)$: Maximum predicted Q-value for the next state's s_{t+1} across all possible actions

$\leftarrow$: Indicates the update (assignment) of the Q-value

The Q-learning update rule serves as the foundation for the model's learning behavior: The timing discrepancy between the expected and actual outcomes is referred to as TD error. The elements of the Q-learning update rule are shown in equation (1). By adjusting the process reward function design, the incentive structure was continuously enhanced to achieve a balance between lowering power costs, utilizing solar energy that was available, and preserving a suitable degree of user comfort.

Algorithm - Learning Procedure of proposed Q-learning method

1: Input: state of the agent s(t)

2: Output: action of the agent a(t)

3: Initialize: Q-table Q(s, a) with zeros or random values

– Learning rate α, discount factor γ, exploration rate ε

4: for episode = 1 to max_episode do

5: Initialize environment and get initial state s(t)

6: for time step = 1 to max_step do

7: With probability ε, choose random action a(t); otherwise, choose action that maximizes Q(s(t), a)

8: Execute action a(t), observe reward r(t) and next state s(t+1)

9: Update Q-table:

$$Q(s(t), a(t)) \leftarrow Q(s(t), a(t)) + \alpha [r(t) + \gamma \max\_a' Q(s(t+1), a') - Q(s(t), a(t))]$$

10: Set s(t) = s(t+1)

11: end for

12: Decay ε to reduce exploration over time

13: end for

4.2.5 Validation

- Without schedule of appliance: High peak rate usage and minimum use of solar, battery
- Q-Learning Implementation: Appliance ON/OFF timing shifted from high costing hours to low costing hours without compromising daily usage

- Comparison of battery and solar dependency versus traditional electricity usage from grid

4.2.7 A discussion of any limitations in the model or software

With positive findings, the current model has following drawbacks:

- Model does not include occupant comfort zones such as temperature condition and HVAC system
- No buyback pricing included as a reward function
- The whole system works on a static next 15 days solar irradiance data which might be slightly updates by time
- Python uses more memory size compared to low level languages, which may raise problem with large dataset. Also, python is not ideal for mobile applications because it needs light weight runtime.

4.3 Engineering Practice Investigations

A quantitative approach used in this project to collect data and investigation to model appliance behaviour, battery state of charge and solar availability. This approach is best suitable because of numerical based RL model for time-of-use pricing and solar data.

4.3.1 Data Collection Methods

- Real time solar data and TOU pricing from local electricity provider stored in CSV file for further usage in python code.
- Appliance rating and usage's excel sheet developed from household interviews and manufacturer datasheet.

4.3.2 Information sources and Standards

- IEEE Smart Grid guidelines and ASHRAE for residential building
- Tesla Powerwall battery datasheet specification
- Synergy electricity provider documentation

5. Results & Discussion

5.1 PV cell, irradiance and power calculation

$$P_{PV}(t) = P_{PV,rated} \frac{G}{G_{STC}} \left(1 - k(T_c - T_r)\right) \qquad \text{.......................................(2)}$$

$where\ P_{PV,rated} = rated\ output\ of\ PV\ cell$

$k = power\ temperature\ coefficient$

$G = Light\ radiation\ intensity$

$T_c = Atmospheric\ temperature$

$$G_{STC} = 1000\ \frac{W}{m^2},\ T_r = 25°C$$

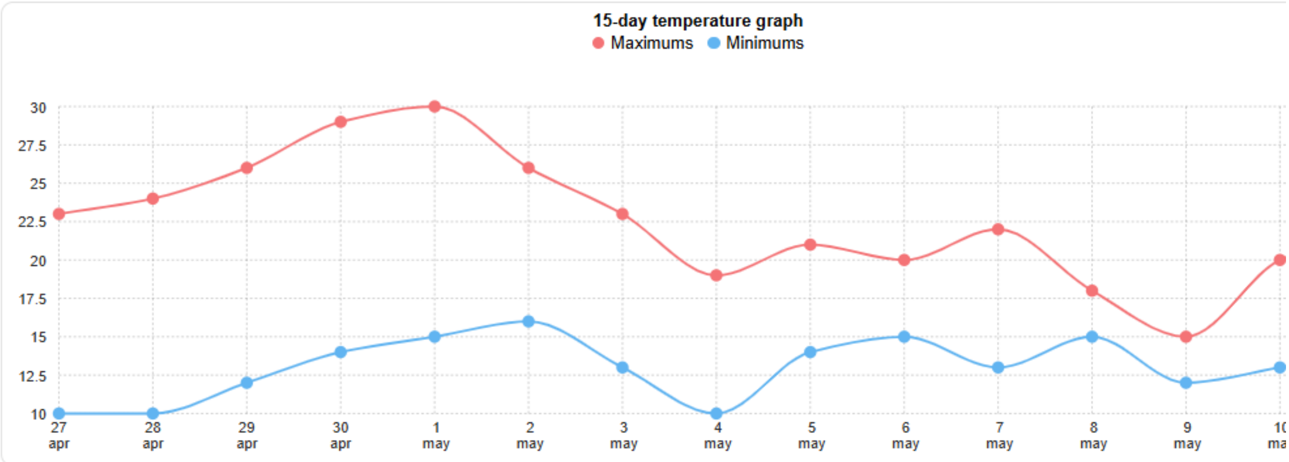


Figure 5 Atmospheric temperature Tc from 27 April to 10 May at selected location

Selected residential building has 6.3kWh solar capacity with 330 watt panel wattage and 19 panels. Here, Table 4 shows calculated energy from solar irradiance data which is save din CSV file from "tutiempo" website. Temperature and light radiations are decisive parameter to get better result from equation 2.

Using solar Equation 2, solar power generation for a 15-day was effectively calculated by Python programming . The realistic solar output was achieved by averaging and distributing the solar irradiance data into hourly segments from 7:00 to 17:00, depicted in figure 6.

Table 6 : Solar energy data (Watts) from 28 April to 12 May 2025

Table 6(a) - Hour 7 am to 12 pm

Date	7am	8am	9am	10am	11am	12pm
28-Apr	48.38	1070.50	2026.08	2751.84	3634.85	3749.76
29-Apr	45.60	1172.56	2436.31	3524.18	4234.23	4546.91
30-Apr	39.01	1157.29	2444.60	3491.36	4193.53	4505.61
01-May	774.14	1612.80	2580.48	3548.16	4515.84	4838.40
02-May	832.10	1664.21	2624.33	3584.45	4544.57	4864.61

03-May	892.58	1721.41	2677.75	3634.09	4590.43	4909.21
04-May	830.47	1660.93	2619.16	3577.39	4535.62	4855.03
05-May	809.55	1651.48	2622.94	3594.40	4565.86	4889.68
06-May	787.75	1641.15	2625.84	3610.53	4595.22	4923.45
07-May	845.21	1690.42	2665.66	3640.90	4616.14	4941.22
08-May	812.70	1657.91	2633.15	3608.39	4583.63	4908.71
09-May	780.19	1625.40	2600.64	3575.88	4551.12	4876.20
10-May	749.13	1595.98	2573.11	3550.24	4527.37	4853.08
11-May	726.26	1584.58	2574.94	3565.30	4555.66	4885.78
12-May	686.64	1536.76	2517.67	3498.58	4479.49	4806.46

Table 6(b) - Hour 1 pm to 5 pm

Date	1pm	2pm	3pm	4pm	5pm
28-Apr	3126.82	2479.68	2159.14	1427.33	447.55
29-Apr	4436.17	3908.52	3009.56	1810.95	495.08
30-Apr	4395.08	3868.45	2971.23	1781.44	468.12
01-May	4515.84	3870.72	2903.04	1935.36	967.68
02-May	4544.57	3904.49	2944.37	1984.25	1024.13
03-May	4590.43	3952.87	2996.53	2040.19	1083.85
04-May	4535.62	3896.80	2938.57	1980.34	1022.11
05-May	4565.86	3918.22	2946.76	1975.30	1003.84
06-May	4595.22	3938.76	2954.07	1969.38	984.69
07-May	4616.14	3965.98	2990.74	2015.50	1040.26
08-May	4583.63	3933.47	2958.23	1982.99	1007.75
09-May	4551.12	3900.96	2925.72	1950.48	975.24
10-May	4527.37	3875.95	2898.82	1921.69	944.56
11-May	4555.66	3895.42	2905.06	1914.70	924.34
12-May	4479.49	3825.55	2844.64	1863.73	882.82

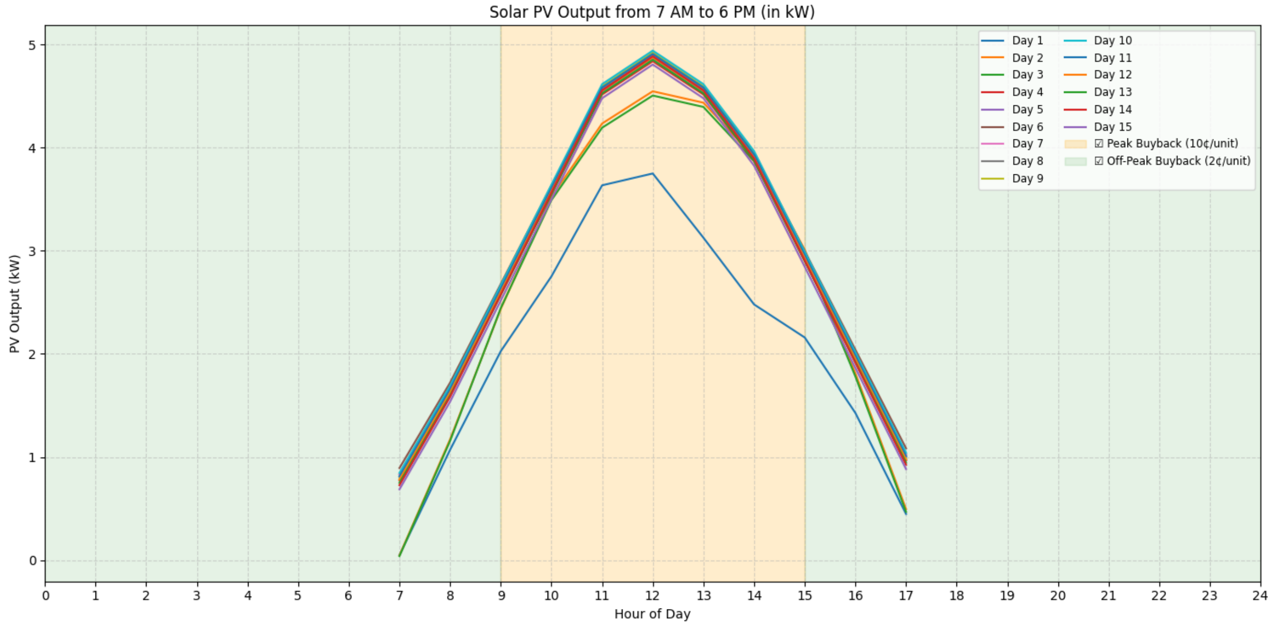


Figure 6 Hourly solar PV output after Python code implementation

5.2 Appliance load profile

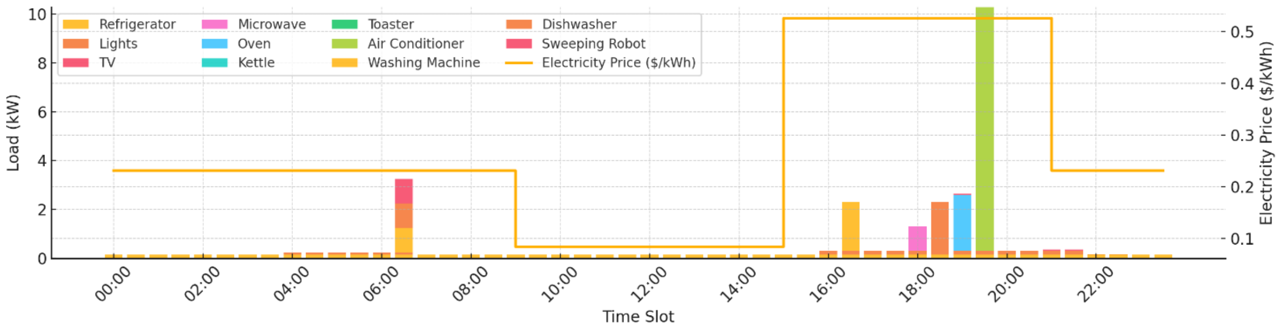


Figure 7 Appliance loads and Electricity price over 24 hours

Table 7 – Energy consumption by each load

Device	**Scaled Energy (kWh)**
Refrigerator	5.888380
Lights	2.192370
TV	0.087625
Microwave	0.876247
Oven	2.015368
Kettle	0.000000
Toaster	0.000000
Air Conditioner	8.762469

Washing Machine	2.628740
Dishwasher	2.628740
Sweeping Robot	0920059
Total	Nearly 26.00

According to the appliance load profile, energy consumption fluctuates throughout the day, with clear peaks in the early morning and evening. The refrigerator, lights, and TV all contribute consistently throughout the day, but the air conditioner produces a notable rise about 20:00. The microwave, oven, and kettle are all used simultaneously in the evening, which results in high loads as well. Because electricity prices fluctuate, the greatest rates are seen between 16:00 and 20:00, when most appliances are used.

5.3 Battery simulation [3][12]

To overcome intermittence of renewable energy source battery system used to give sufficient power to appliance. This research work highlights how Tesla Power Wall 2 implemented to energize loads during high peak period of electricity price plus less solar power available in system. Tesla battery simulation provides usable energy up to 13.5 kWh with 50 Hz grid frequency.

1. Battery SOC equation [3]

$$SOCB(t) = SOCB(t-1) + \left[(Pc(t) * \eta c/cap) * Xc(t) - \left(Pd(t)/(cap * \eta d)\right) * Xd(t)\right] * \Delta t$$

2. Charge/Discharge Limits

$$0 \leq Pc(t) \leq Pc, \max$$

$$0 \leq Pd(t) \leq Pd, \max$$

SOCB(t): State of charge of the battery at time t
Pc(t): Charging power at time t
Pd(t): Discharging power at time t
ηc, ηd: Charging and discharging efficiency
cap: Rated capacity of the battery
Xc(t), Xd(t): Binary variables indicating charging or discharging action
SOCB,min, SOCB,max: Minimum and maximum state of charge
Δt: Time interval

Table 8 - Tesla Power Wall 2 specification [12]

Specification	Value
AC Voltage (Nominal)	230 V
Feed-In Type	Single Phase
Grid Frequency	50 Hz
Total Energy	14 kWh
Usable Energy	13.5 kWh
Real Power, Max Continuous (Charge/Discharge)	5 kW
Apparent Power, Max Continuous (Charge/Discharge)	5 kVA
Maximum Supply Fault Current	10 kA
Maximum Output Fault Current	32 A
Power Factor Output Range	+/– 1.0 adjustable
Internal Battery DC Voltage	50 V
Round Trip Efficiency	90%
Warranty	10 years
Operating Temperature	–20°C to 50°C
Recommended Temperature	0°C to 30°C
Operating Humidity (RH)	Up to 100%, condensing
Storage Conditions	–20°C to 30°C; Up to 95% RH, non-condensing
State of Energy (SoE)	25% initial
Maximum Elevation	3000 m
Environment	Indoor and outdoor rated
Ingress Rating	IP67 (Battery & Power Electronics); IP56 (Wiring Compartment)
Wet Location Rating	Yes
Noise Level @ 1 m	< 40 dBA at 30°C
Dimensions	1150 mm x 753 mm x 147 mm
Weight	114 kg
Mounting Options	Floor or wall mount

Certifications	IEC 62109-1; IEC 62109-2; IEC 62619; UN 38.3
Grid Connection	Worldwide Compatibility
Emissions	IEC 61000-6-1; IEC 61000-6-3
Environmental Directives	RoHS Directive 2011/65/EU; WEEE Directive 2012/19/EU; Battery Directive 2006/66/EC; REACH Regulation
Seismic Standards	AC156; IEEE 693-2005 (high)

5.4 Q learning setup and training

Scenario 1:

Before applying Q-learning algorithm in existing system solar gives power to home appliances whenever usage increases and battery backup use in peak period of electricity price. Figure 8 shows solar output on 28 April 2025 with average usage of home appliance energy per day. Meanwhile, figure 9 shows tesla Power Wall charging and discharging time from 10 am to 9 pm. This framework does not rely on or follows how to shift energy source such as synergy, solar or battery backup whenever electricity price increase or decrease. Hence, reinforcement learning helps to determine timing and scheduling of home appliance without any human intervention and data records from history. Particularly, Q- learning rely on trial-and-error method to get maximum output for reward function which described in Scenario 2.

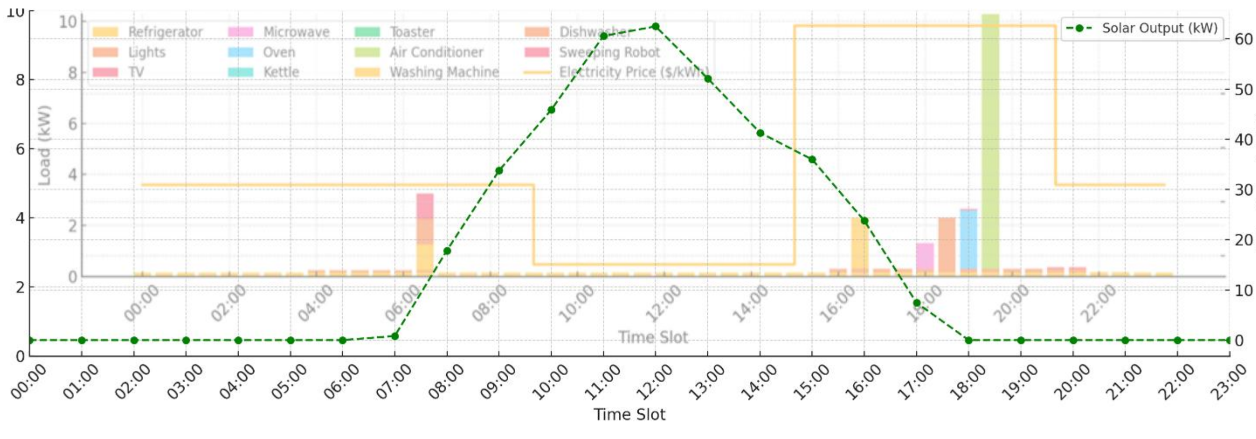


Figure 8 Appliance Load, Electricity Price and Solar Output on 28 April 2025

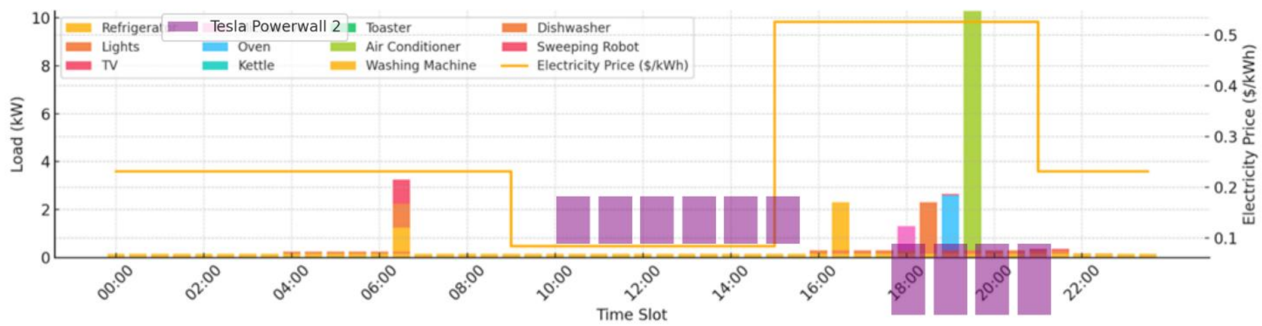


Figure 9 Tesla Power Wall discharging during peak period

Scenario 2:

It has machine learning capability to overcome challenges that highlighted in scenario 1 and get maximum close to reward function and decrease electricity cost in next billing cycle. Selected residential building provides information regarding average daily usage/unit and average daily cost with 4.7954 units and $2.65 per day respectively. The goal is to adjust ON/OFF timing of available shiftable appliances, maximum use of solar and usage of battery backup activation during peak hours of use which is 4:00PM to 9:00PM in this case and charging of battery when electricity price is low.

Table 9 – Q-Learning Parameters

Parameter	Value
Learning Rate (α)	0.1
Discount Factor (γ)	0.95
Exploration Rate (ε)	0.1
Training Episodes	1000

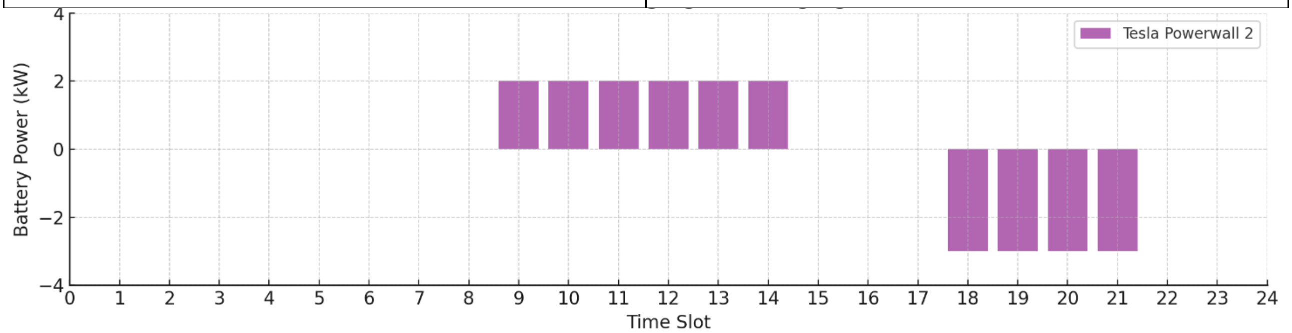


Figure 10 Tesla Power Wall 2 Charging/Discharging state after Q-learning

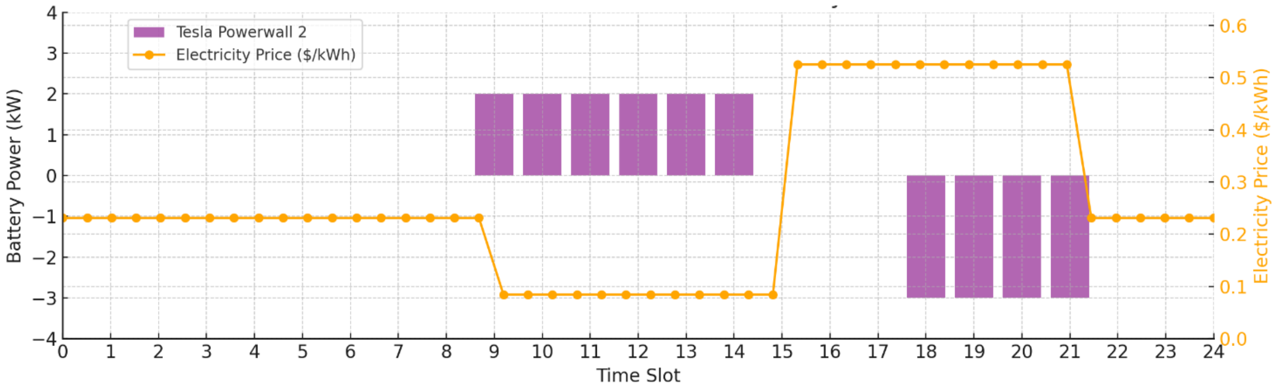


Figure 11 Tesla Power Wall Charging/Discharging with Electricity prices

Optimization Results:

Net savings: Over one day nearly $0.80 saved when prices are low and battery discharge when prices are high

Idea: Battery will charger during peak time of solar generation between 9:00am to 1:00pm as mentioned in figure 11 and then battery will discharge by the time of 6:00pm to 10:00pm when prices are high.

6 Conclusion and Future work

This research successfully optimizes the application of machine learning to schedule ON/OFF time of home appliance and battery backup, specifically give attention to coordination of Solar, Electricity price, Tesla battery. The key domain of this work is:

1) Optimal behaviour of battery: Q-learning able to manage battery data and update next state during off-peak and peak period of electricity price which leads to reward function and net daily energy cost.
2) Lesser dependency on electricity grid and more dependency on solar power plus battery during appliance activity.
3) The study's objective, which involves scheduling energy and use of reinforcement learning mostly met. By the figure, Q-learning significantly reduce energy cost while operational constraints.
4) However, Battery degradation time, efficiency losses and inverter efficiency were not included during parameter tuning.

Future Work:

1) Battery Health Monitoring: Degradation effect, charge/discharge efficiency and adding more episodes in Q-learning increase accuracy of close value to reward function.
2) Buyback price: Add buyback price to save energy and generate revenue
3) Real time data logging from website to avoid renewable energy intermittency. Also, implement real hardware model such as raspberry pi and Arduino for live feedback from sensor's output.

In a nutshell, this research work presents a strong baseline for Building Energy Management System (BEMS) with potential of integrating smart grid model and benefits prosumers.

7 References

[1] Liu, Y., et al., 2022. A home energy management system incorporating data-driven uncertainty-aware user preference. Appl. Energy 326, 119911. http://dx.doi.org/10.1016/j.apenergy.2022.119911.

[2] H. R. O. Rocha, *Home energy management systems (HEMS) in the demand response market*, 2023.

[3] S. Xiong, *Energy demand planning in smart homes using a multi-objective optimization algorithm*, 2023.

[4] T. H. B. Huy, *Optimized scheduling model for HEMS with Deep Reinforcement Learning*, 2023.

[5] S. B. Slama, *A supervised-learning-based HEMS framework for real-time scheduling*, 2023.

[6] J. Wang, *An intelligent home energy control algorithm combining Q-learning and neural networks*, 2023.

[7] E. Mocanu, *Multi-agent Deep Reinforcement Learning for DSM and P2P energy trading*, 2023.

[8] A. Sheikhi, *Deep Reinforcement Learning for online optimization of electricity scheduling*, 2023.

[9] C. Lingmin, *Source-load coordinated energy management using Q-learning in coastal residential areas*, *Electric Power Systems Research*, vol. 214, 2023.

[10] Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.

[11] Kotevska, O., & Andelfinger, P. (2021). Reinforcement learning for intelligent building energy management system control. In *Proceedings of the 2021 International Conference on Smart Energy Systems and Technologies (SEST)* (pp. 1–6). IEEE. https://doi.org/10.1109/SEST50973.2021.9543545

[12] Tesla. (n.d.). *Powerwall 2 AC datasheet (Australia)* [PDF]. Tesla, Inc. https://www.tesla.com/sites/default/files/pdfs/powerwall/Powerwall%202_AC_Datasheet_en_AU.pdf

8 Appendices

Q-Table

Hour	SOC	Charge	Discharge	Idle
0	0	-0.45009	0.801347	0.496682
1	2	-0.095	0.120629	0
2	4	-0.05	0.075	0
3	6	-0.05	0	0
4	8	-0.05	0	0
5	10	-0.05	0	0
6	12	0	0	0
7	12	0	0	0
8	12	0.00285	0	0
9	12	0	0.057	0
10	12	0	0	0
11	12	0	0	0
12	12	0	0.057	0
13	12	0	0.03	0
14	12	0.058076	0	0
15	12	0	0.611325	0
16	12	0	0	0
17	12	0	0	0
18	12	0.000677	0	0
19	12	0.01995	0	0
20	12	0	0.210375	0
21	12	0	0	0
22	12	0	0	0
23	12	0	0	0

Solar Power Generation Program

```
import numpy as np
import matplotlib.pyplot as plt
import random

############################################
# SOLAR PV DATA
############################################

# Solar PV output data (in watts) for 15 days
pv_data = [
    [48.384, 1070.496, 2026.08, 2751.84, 3634.848, 3749.76, 3126.816, 2479.68, 2159.136,
1427.328, 447.552],
    [45.599, 1172.556, 2436.311, 3524.182, 4234.23, 4546.912, 4436.17, 3908.52, 3009.56,
1810.948, 495.079],
    [39.01, 1157.285, 2444.602, 3491.359, 4193.532, 4505.609, 4395.082, 3868.452, 2971.231,
1781.438, 468.115],
    [774.144, 1612.8, 2580.48, 3548.16, 4515.84, 4838.4, 4515.84, 3870.72, 2903.04, 1935.36,
967.68],
    [832.104, 1664.208, 2624.328, 3584.448, 4544.568, 4864.608, 4544.568, 3904.488, 2944.368,
1984.248, 1024.128],
    [892.584, 1721.412, 2677.752, 3634.092, 4590.432, 4909.212, 4590.432, 3952.872, 2996.532,
2040.192, 1083.852],
    [830.466, 1660.932, 2619.162, 3577.392, 4535.622, 4855.032, 4535.622, 3896.802, 2938.572,
1980.342, 1022.112],
    [809.55, 1651.482, 2622.942, 3594.402, 4565.862, 4889.682, 4565.862, 3918.222, 2946.762,
1975.302, 1003.842],
    [787.752, 1641.15, 2625.84, 3610.53, 4595.22, 4923.45, 4595.22, 3938.76, 2954.07, 1969.38,
984.69],
    [845.208, 1690.416, 2665.656, 3640.896, 4616.136, 4941.216, 4616.136, 3965.976, 2990.736,
2015.496, 1040.256],
    [812.7, 1657.908, 2633.148, 3608.388, 4583.628, 4908.708, 4583.628, 3933.468, 2958.228,
1982.988, 1007.748],
    [780.192, 1625.4, 2600.64, 3575.88, 4551.12, 4876.2, 4551.12, 3900.96, 2925.72, 1950.48,
975.24],
```

```
    [749.133, 1595.979, 2573.109, 3550.239, 4527.369, 4853.079, 4527.369, 3875.949, 2898.819,
1921.689, 944.559],
    [726.264, 1584.576, 2574.936, 3565.296, 4555.656, 4885.776, 4555.656, 3895.416, 2905.056,
1914.696, 924.336],
    [686.637, 1536.759, 2517.669, 3498.579, 4479.489, 4806.459, 4479.489, 3825.549, 2844.639,
1863.729, 882.819],
]

pv_avg_w = np.mean(pv_data, axis=0)

pv_output_w = np.zeros(48)
pv_output_w[14:25] = pv_avg_w  # from 7:00 to 17:00
pv_output_kw = pv_output_w / 1000  # convert to kW



###########start######

# Electricity prices
prices = [0.2311] * 9 + [0.0845] * 6 + [0.5253] * 6 + [0.2311] * 3
prices = np.repeat(prices, 2)

hours = np.linspace(0, 24, 48)

############################################
# APPLIANCES DATA
############################################

# Non-shiftable appliances
lights = np.zeros(48)
lights[8:14] = 99/1000
lights[32:44] = 154/1000
lights[44:46] = 30/1000

tv = np.zeros(48)
tv[42:44] = 50/1000
```

```
microwave = np.zeros(48)
microwave[36:37] = 1000/1000

oven = np.zeros(48)
oven[38:39] = 2300/1000

kettle = np.zeros(48)
kettle[13] = 1000/1000

toaster = np.zeros(48)
toaster[13] = 1000/1000

air_conditioner = np.zeros(48)
air_conditioner[39:40] = 10000/1000

# Fixed appliance
refrigerator = np.full(48, 280/1000/2)  # half load


##END###
```