CityLearn

CityLearn is a framework for the implementation of **multi-agent or single-agent** reinforcement learning algorithms for urban energy management, load-shaping, and demand response.

CityLearn uses building(9 buildings) hourly data from pre-simulated models and assumes that the indoor temperatures of the building do not change as a function of the actions of the controllers.

<u>Support for OpenAI Gym environment</u>

Inherited methods from OpenAI Gym

- Step()
- Get_ob()
- _terminal()

Input files

- Weather data
- Building energy models
- PV data
- Carbon intensity

User defined input parameters – we can change these parameters

- Building attributes and energy subsystem in buildings (heat pumps, batteries, thermal energy, storage tanks, electric heaters)
- Building state-action spaces
- Reward functions

data Provided – 9 buildings data (one medium office, one fast-food restaurant, one standalone retail, one strip mall retail and five medium multi-family buildings.)

CityLearn receives input restimulated building hourly data and allowed the control of the charging and discharging of the thermal energy storage devices installed in the buildings.

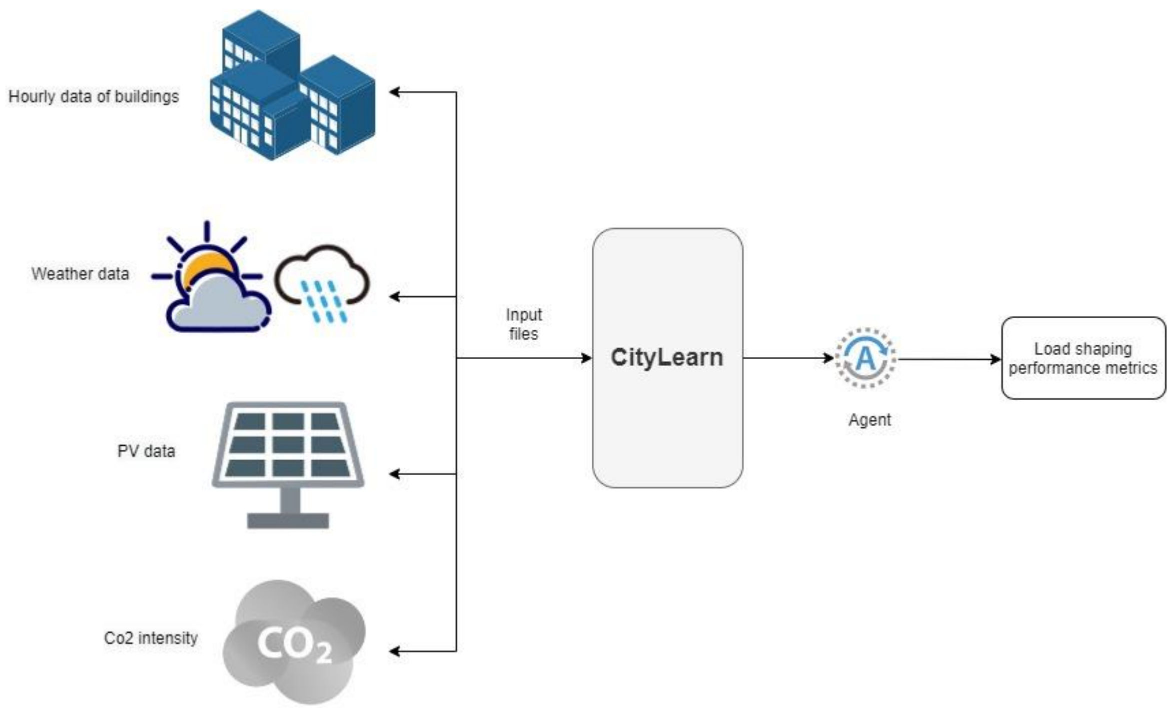
The heating energy was supplied by electric heaters while cooling energy was supplied by air-to-water heat pumps.

some buildings were equipped with a PV system to produce on-site energy.

The controllers were designed to manage the charging and discharging of cooling and DHW storages for the district of buildings, with the aim to minimize electricity costs and reduce electrical load as requested during the incentive-based DR events considered

Each building has an air-to-water heat pump, and most buildings also have an electric heater that supplies them with DHW.

All these devices, together with other electric equipment and appliances (non-shiftable loads) consume electricity from the main grid.



1. Environment

Action and State space

Continuous action and continuous state space

Actions

- Cooling storage
- Domestic hot water storage
- Battery storage

States

- Outdoor temperature
- Indoor temperature
- Outdoor humidity
- Indoor relative humidity

- Diffuse solar radiation
- Direct solar radiation
- Electricity consumption
- PV generation
- state of the charge of the cooling storage device
- state of the charge of the domestic hot water (DHW) storage device
- net electricity consumption of the building in the current time step

Reward

All the costs are normalized by the costs of a benchmark rule-based controller (RBC).
get_baseline_cost() - returns the costs of a Rule-based controller (RBC), which is used to divide the final cost by it.

- ramping – sum of electricity consumption in every timestep
- 1-load factor – average net electricity load / maximum electricity load
- Average daily peak
- Peak demand - maximum peak electricity demand
- Net electricity consumption - total amount of electricity consumed
- Carbon emission
- Total cost

2. How to Implement Simple Environment

1. Define parameters
    a. input files
    b. building ids
    c. simulation period
    d. agent type(single or multi)
    e. cost function
2. Create cityLearn environment using defined parameters
3. Simulate environment using env.step function.

GitHub - How to create simple environment –
https://github.com/anushaihalapathirana/CityLearn/blob/master/tSimpleEnv.py

GitHub – Centralized environment -
https://github.com/anushaihalapathirana/CityLearn/blob/master/tCenterlizeEnv.py

3. AI Algorithms

1. SAC – multi-agent RL controller
2. RBC for benchmarking – **this use to normalize the actual costs**. Daytime release stored energy and in early night time it store DHW and/or cooling energy.
3. MARLISA - Multi-Agent Reinforcement Learning with Iterative Sequential Action

GitHub – SAC - https://github.com/anushaihalapathirana/CityLearn/blob/master/tSAC.py

GitHub - RBC - https://github.com/anushaihalapathirana/CityLearn/blob/master/tRule-based-controller-agent.py

GitHub – MARLISA -

Results

Only for 1 building and simulation period is with 4 years data.

* - normalized using RBC

Algorithm	Cumulative cost	Average daily peak	Peak demand	Carbon emission	Net electricity consumed	Total cost
RBC	4 478 864 124.99	1.0	1.0	1.0	1.0	1.0
SAC	6 351 247 359.61	1.23*	1.40*	1.04*	1.03*	1.348*

References

1. GitHub Link - https://github.com/anushaihalapathirana/CityLearn