



tacwin

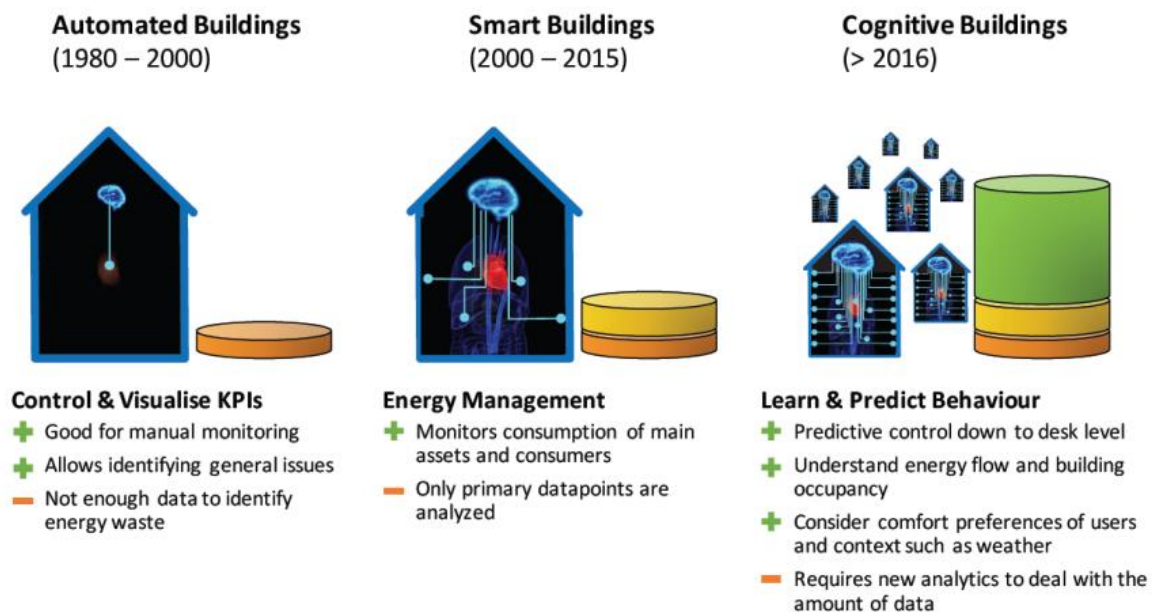
الگوریتم یادگیری سیستم کنترل

مقدمه

اصطلاح "هوش مصنوعی" به طور فزاینده ای با ساختمان ها و اتوماسیون ساختمان مرتبط شده است. سیستم های اتوماسیون ساختمان امروزی به طور عمده در پاسخ به برنامه های زمان ثابت یا پارامترهای کنترلی ساده به صورت «ایستا» عمل می کنند. کنترل دمای اتاق بر اساس دمای از پیش تعیین شده است که در طول روز یکسان است. روشنایی به صورت دستی، با سوئیچ ها یا بر اساس سوئیچ های حضور ساده کار می کند. هیچ کدام از اینها واقعاً «هوشمند» نیستند. بعد جدیدی که هوش مصنوعی می تواند به محیط اتوماسیون ساختمان اضافه کند، استفاده از تجزیه و تحلیل مستقل داده ها به عنوان مبنایی برای عملیات بهینه سازی شده است. بنابراین دینامیک گرمایش و سرمایش اتاق ها، پیش بینی آب و هوا، اشغال اتاق پیش بینی شده در طول روز همگی می توانند در عملیات گرمایش لحاظ شوند.

به عبارتی در ابتدا، «اتوماسیون ساختمان» نسبتاً «غیر هوشمند» بود. سیستم ها طوری برنامه ریزی می شدند که از مجموعه ای از قوانین ساده پیروی کنند، که امکان راه اندازی سریع سیستم و در نتیجه سهولت تعمیر و نگهداری را فراهم می آورد.

«ساختمان هوشمند» معمولاً بر اساس اتوماسیون ساختمان کلاسیک با سیستم های مدیریت مبتنی بر فناوری اطلاعات انعطاف پذیر ساخته می شود. رشد فزاینده داده های تولید شده توسط سنسورها در سطح مدیریت مبتنی بر فناوری اطلاعات، راه حلی نوین برای روش های پردازش داده پیشرفته تر فراهم کرده است. پردازش پیچیده داده های سنسورها منجر به تبدیل ساختمان هوشمند به ساختمان Cognitive شده است.



الگوریتم اجرایی یادگیری

اولین مرحله در هر فرآیند هوش مصنوعی، یادگیری سیستمی است. این می تواند به سه شکل باشد:

- Unsupervised Learning
- Supervised Learning
- Reinforcement Learning

در روش اول حجم زیادی از داده های سنسورها خوانده میشود و با یادگیری سیستم، در صورت وجود مغایرت در آنها به اپراتور اطلاع داده می شود. این روش تنها برای مانیتورینگ در ساختمان کاربرد دارد. به عبارتی تنها برای طبقه بندی کردن حجم زیاد داده استفاده میشود. در روش دوم بر اساس داده ها سیستم یاد میگیرد و با استفاده از آن پیش بینی انجام می دهد. به عنوان مثال برای شناسایی نقاط پیک مصرف انرژی با کمک داده های پروفایل مصرفی انرژی ساختمان.

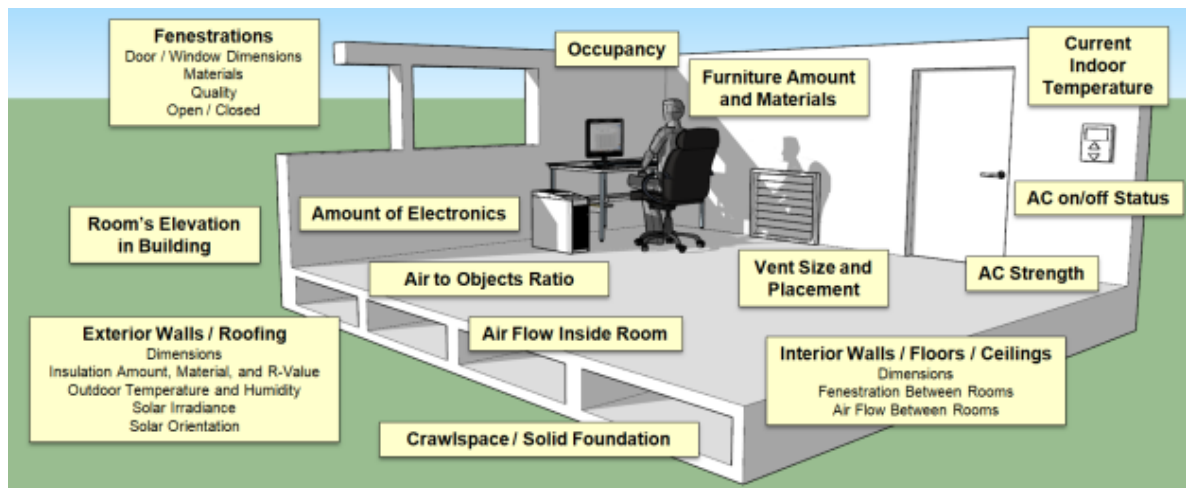
در روش آخر که تکوین پیاده سازی و اجرا کرده است، سیستم قابلیت این را دارد که هوشمندانه تصمیم گیری کند. برای تشخیص بهترین زمان روشن و خاموش کردن سیستم سرمایشی و نیز بهترین ست پوینت دما، با استفاده از سنسورهای دما-رطوبت دما و رطوبت داخلی ساختمان و نیز دما و رطوبت بیرونی ساختمان را میخواند و این داده ها در سیستم ذخیره سازی می شود. سیستم با این داده ها آموزش داده میشود. علاوه بر ارسال پیام در صورت مشاهده مغایرت در سنسورها، با استفاده از داده های دریافتی از سایت هواشناسی قابلیت تشخیص روزهای پیک بار حرارتی را دارد. داده هایی نظیر دما، جهت تابش خورشید، رطوبت نسبی و سرعت باد از داده های مهم موردنیاز می باشد که از سایت هواشناسی گرفته می شود.

با توجه به داده های خوانده شده از سنسورها و نیز داده دریافتی از سایت هواشناسی، برای زمان روشن خاموش کردن سیستم سرمایشی و دمای آسایشی سیستم تصمیم گیری میکند.

علاوه بر این در صورت مشاهده پیک ناگهانی در بار حرارتی (با استفاده از اختلاف مقدار دما و رطوبت سنسور محیط و نیز دمای setpoint، سیستم پیک بار را شناسایی می کند)، سیستم در طول زمان آموزش میبندد که آیا این پیک بار حرارتی تناوبی رخ می دهد و یا ناگهانی (به عنوان مثال همیشه ساعت 11 صبح در اتاق جلسه پیک ناگهانی وجود دارد و یا فقط بصورت تصادفی گاهی پیک بار حرارتی وجود دارد). و بر اساس این یادگیری برای پایین آوردن دمای setpoint فضای مربوطه و نیز در صورت نیاز شروع به کار زودتر، به صورت هوشمند بالانس حرارتی را انجام میدهد. لازم به ذکر است برای آموزش سیستم علاوه بر سایت هواشناسی، داده های مربوط به پروفایل occupancy به عنوان ورودی سیستم می باشد و همان طور که گفته شد بر اساس تغییر تعداد افراد موجود در فضاها بصورت هوشمندانه سیستم را کنترل می کند که از مصرف انرژی جلوگیری شود.

استفاده از الگوریتم یادگیری و هوشمند سازی سیستم

کنترل پیش‌بینی مبتنی بر مدل (MPC^1) در چند سال گذشته به دلیل پتانسیل اش برای صرفه جویی قابل توجه در انرژی HVAC محبوب شده است. MPC از یک مدل ساختمان برای پیش‌بینی عملکرد ساختمان در آینده استفاده می‌کند و به این صورت می‌تواند تصمیمات کنترلی بهینه را برای آینده اتخاذ کند. این سیستم ها برای پیش بینی و تصمیم گیری نیاز به یک مدل انرژی دقیق از ساختمان و سیستم HVAC دارند. به دلیل غیرخطی بودن و وابستگی به پارامترهای متعدد، مدل سازی حرارتی با چالش مواجه است. شکل زیر برخی از پارامترهای دخیل در تنظیم دمای محیط داخلی را نشان می دهد.



شکل 1- پارامترهای تاثیرگذار در مدل سازی انرژی ساختمان

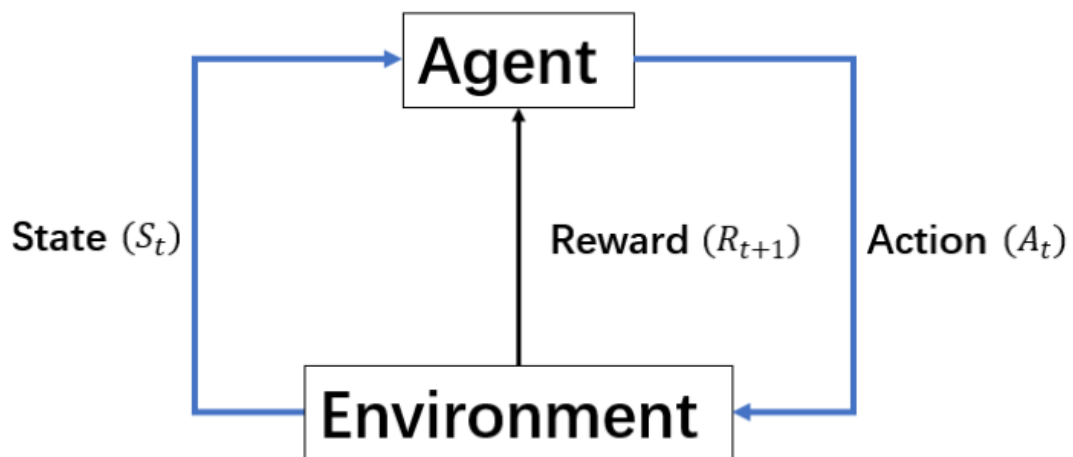
همانطور که از شکل مشاهده میشود، ابعاد داخلی اتاق و همچنین مواد تشکیل دهنده دیوار، کف و سقف نقش مهمی دارند. اشیاء، لوازم، افراد و برای دیوارهای داخلی انتقال حرارت بین اتاق ها باید مورد توجه قرار گیرد. باز و بسته بودن درب ها و جنس متریال استفاده شده در درب ها اهمیت دارد. همین اصل در مورد پنجره ها نیز صدق می کند - پنجره هایی که می توانند تک جداره، دو جداره، باز یا بسته باشند. دیوارهای خارجی برهمکنش انتقال حرارت با دمای بیرون و خورشید دارند، تعاملی که به موقعیت، جهت و عرض جغرافیایی خانه و همچنین میزان نور خورشید در دسترس بستگی دارد. هنگام مدل سازی یک ساختمان، هر اتاق دارای مشخصات حرارتی خاص خود است که با اتاق های مجاور و بیرون تعامل دارد و جریان هوا بین اتاق ها باید مورد توجه قرار گیرد. این فرآیندها یک شبکه درهم از تعاملات غیرخطی ایجاد می کنند.

چالش اصلی مدل سازی ساختمان و سیستم های HVAC این است که مشخصات حرارتی ساختمان در طول زمان تغییر کند و مدل MPC برای ساختمانی که امروز ساخته شده است، ممکن است یک سال بعد غیرقابل استفاده باشد. لذا برای حل این چالش از روش های هوشمند سازی با استفاده از هوش مصنوعی و الگوریتم های learning استفاده می شود که مستقل از مدل باشند.

¹ Model-based Predictive Control

یادگیری ماشینی (Machine Learning)، به ویژه یادگیری تقویتی (Reinforcement Learning) یا به اختصار RL، با آزمایش محیط به صورت تجربی و یادگیری تکراری، جایگزینی برای سیستم های مبتنی بر مدل ارائه می دهد. در این روش، هیچ پیش آگاهی از مقادیری مانند ضرایب عایق یا بار حرارتی داخلی لازم نیست. یادگیرنده به سادگی تصمیم می گیرد، دانش خود را به روز می کند و تلاش می کند تا بازدهی را به حداکثر برساند. به جای یک مدل شبیه سازی شده که شامل هر متغیر اندازه گیری شده است، آن ویژگی هایی که برای اندازه گیری مقرون به صرفه هستند در State گنجانده می شوند. گفته می شود الگوریتمی که بهینه سازی RL را با تشخیص الگوی یک شبکه عصبی (Deep Neural Network) ترکیب می کند، از یادگیری تقویتی عمیق (DRL¹) استفاده می کند. یک کنترل کننده هوش مصنوعی مجهز به چنین الگوریتمی را می توان طوری پیکربندی کرد که به طور نامحدود در داخل ساختمان یاد بگیرد و تا زمانی که کمترین هزینه عملیات HVAC را به دست آورد، اقدامات اصلاحی خود را انجام دهد.

در روش RL عامل یادگیرنده (learning agent) باید با محیط بصورت مراحل گسسته تعامل کند تا یاد بگیرد که چگونه پاداش (Reward) بازگشتی از محیط (environment) را به حداکثر برساند (شکل 2). تعاملات عامل - محیط در یک مرحله را می توان به صورت $A_t, S_t, S_{t+1}, R_{t+1}$ بیان کرد، که در آن S_t وضعیت (State) محیط را در زمان t نشان می دهد، A_t است اقدامی (Action) که توسط عامل برای تعامل با محیط در زمان t انتخاب می شود، S_{t+1} وضعیت محیط پس از اقدام عامل است و R_{t+1} پاداش دریافتی عامل از محیط است. در نهایت، هدف از کنترل RL، یادگیری یک خط مشی یا سیاست بهینه است که مجموع پاداش دریافتی را حداکثر کند. سیاست یا policy نحوه رفتار عامل یادگیرنده یا agent را تعریف می کند که یک نداشت بین موقعیت هایی که عامل می تواند در آن ها قرار بگیرد (state ها) و اقدامی که باید انجام دهد فراهم می کند. این روش کنترل RL به فرایندهای تصمیم گیری مارکوف (Markov Decision Process) شناخته میشود.



شکل 2- چهارچوب فرایند تصمیم گیری مارکوف

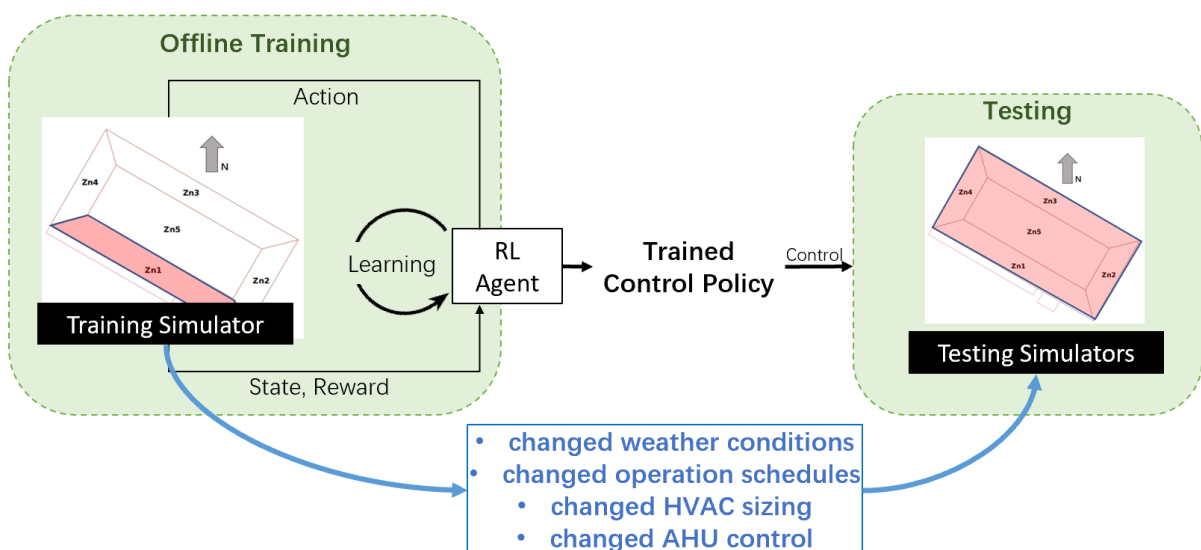
¹ Deep Reinforcement Learning

در شکل بالا ساختمان نقش Environment را دارد، همان کنترلر سیستم HVAC می باشد و State ها پارامترهای ورودی به محیط هستند. Action میتواند تنظیم نقاط کنترل دمایی (Setpoint Temperature) محیط و یا روشن/خاموش کردن سیستم سرمایشی/گرمایشی باشد. در روش های کنترل HVAC در ساختمان، قسمت پاداش یا Reward مینیمم کردن مصرف انرژی و حفظ شرایط آسایش دما-رطوبت محیط است. به عبارتی الگوریتم در هر مرحله (اینتروال های زمانی t)، با استفاده از داده ورودی که State است (می تواند دما-رطوبت بیرونی و داخلی و .. باشد)، و همچنین فیدبک دریافتی از سیستم که آسایش حرارتی محیط می باشد (با توجه به فاصله دمای محیط و Setpoint و زمان رسیدن به Setpoint و همچنین پایداری دمای محیط داخلی میتواند تشخیص دهد که Action اتخاذ شده در مرحله قبل مناسب بوده یا خیر)، میتواند Action ای را اجرا نماید.

روش بکار گرفته شده در تکوین

کنترل با استفاده از الگوریتم یادگیری RL یک روش کنترلی "بدون مدل" می باشد و عامل یاد گیرنده RL هیچ دانش قبلی در مورد فرآیند کنترل شده ندارد. RL یک استراتژی کنترل بهینه را با سعی و خطا یاد می گیرد. بنابراین، می تواند یک روش یادگیری آنلاین باشد که یک استراتژی کنترل بهینه را در طول عملیات ساختمان واقعی به دست می آورد. هر چند که چنین سیستمی میتواند در مرحله یادگیری باعث ایجاد action های ناپایدار شود و همچنین ممکن است به دلیل محدودیت داده های خام ورودی زمان زیادی برای یادگیری سیاست کنترلی پایدار نیاز داشته باشد (مثلاً 20 روز گزارش شده در [36]، 40 روز گزارش شده در [37] و بیش از 50 روز گزارش شده در [38]). بنابراین از روش یادگیری آفلاین برای فاز آموزش استفاده می شود.

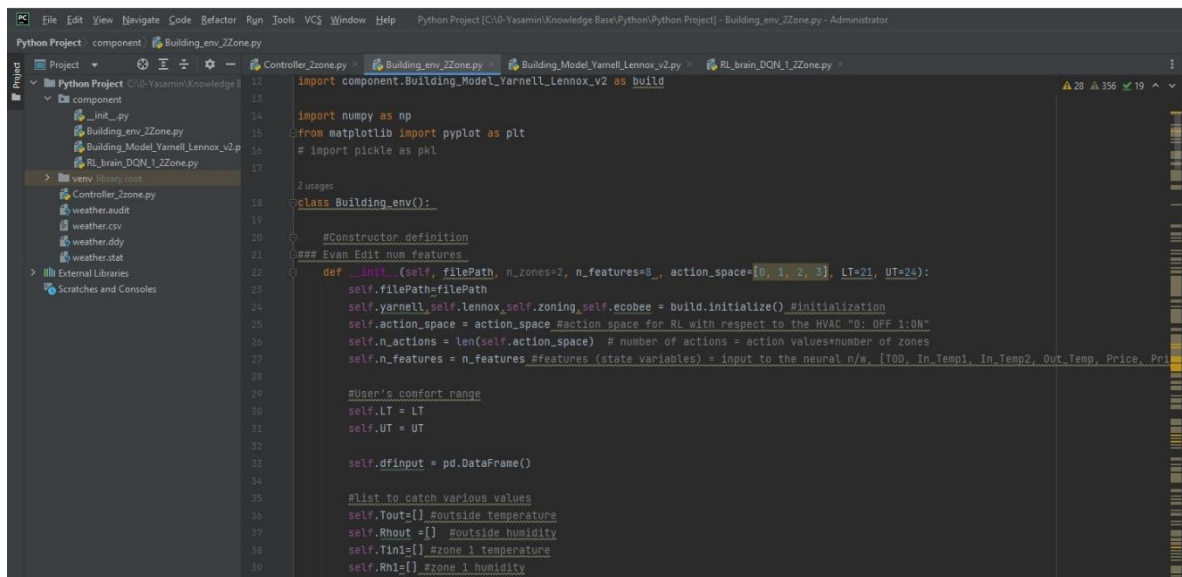
روش بکار برده شده در تکوین به این صورت است که ابتدا عامل یادگیرنده در RL با استفاده از مدل شبیه سازی شده ساختمان، یک سیاست یا policy کنترلی را یاد میگیرد و بعد از یادگیری، در ساختمان واقعی استفاده می شود. شکل زیر فلوچارت کلی روش اتخاذ شده را نشان می دهد که ابتدا بصورت آفلاین با استفاده از نتایج حاصل از شبیه سازی، سیستم RL اجرا می شود و بعد از یادگیری سیاست کنترلی، در ساختمان واقعی استفاده می شود.



شکل 3- فلوجارت جریان کار روش RL استفاده شده

همان طور که گفته شد، پارامترهای اصلی الگوریتم RL شامل State، Action، Reward، Environment و Agent است. در سیستم پیاده سازی شده کنترلر HVAC نقش agent را دارد که عامل یادگیرنده و تصمیم گیرنده می باشد و Environment همان ساختمان دارای سیستم سرمایشی سقف سرد و هواساز می باشد. برای Reward تنظیم بهینه دمای Setpoint فضای داخل و دمای Setpoint هواساز و نیز حداقل انرژی مصرفی در نظر گرفته شده است. پارامترهای مربوط به State شامل دما-رطوبت محیط بیرون، دما-رطوبت محیط داخلی، سرعت باد، جهت تابش خورشید، دما و رطوبت هواساز، دمای Setpoint و پرو فایل occupancy می باشد. تصمیم گیری برای Setpoint دمای محیط و نیز زمان شروع/خاموش سیستم سرمایشی و هواساز تحت Action تعریف شده است.

شکل 4 بخشی از برنامه پایتون را برای تعریف State ها نشان میدهد. با توجه به مقدار حداقل و حداکثر هر State مقادیر به عددی بین 0 تا 1 نرمالیزه میشوند. شکل 5 مازول ها و کتابخانه های استفاده شده در برنامه main را نشان می دهد.

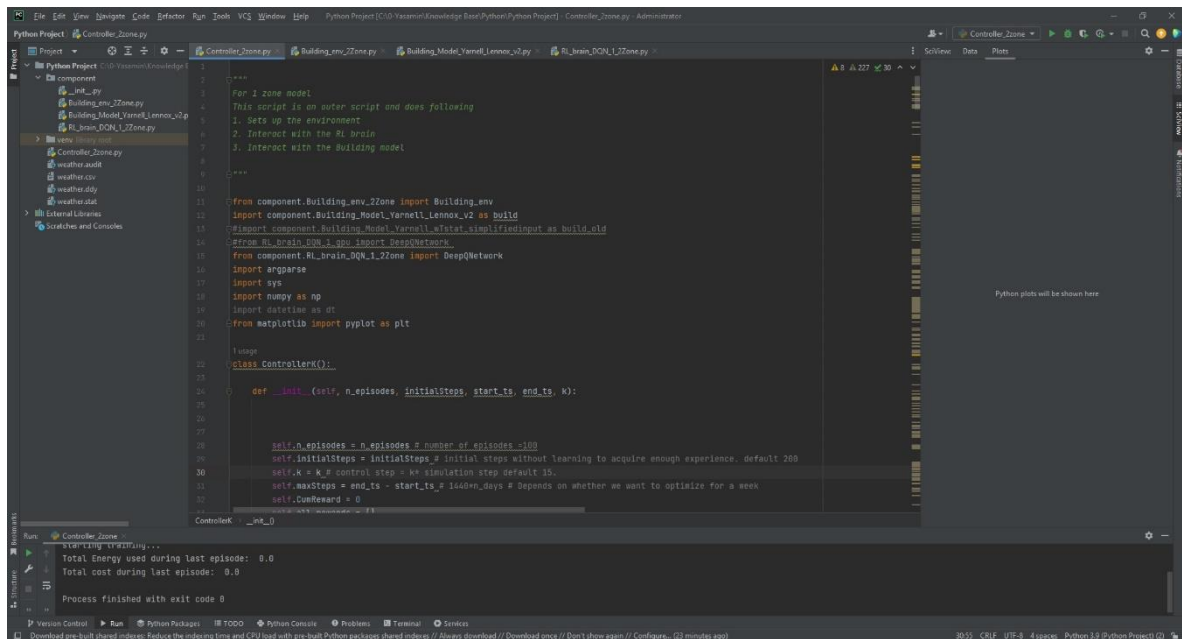


```

12 import component.Building_Model_Yarnell_Lennox_v2 as build
13
14 import numpy as np
15 from matplotlib import pyplot as plt
16 # import pickle as pkl
17
18 # usages
19
20 class Building_env():
21     # Constructor definition
22     ## Evan Edit num features
23     def __init__(self, filePath, n_zones=2, n_features=8, action_space=[0, 1, 2, 3], LT=21, UT=24):
24         self.filePath=filePath
25         self.yarnell,self.lennox,self.zoning,self.ecobee = build.initialize() #initialization
26         self.action_space = action_space #action space for RL with respect to the HVAC "0: OFF 1:ON"
27         self.n_actions = len(self.action_space) # number of actions = action values*number of zones
28         self.n_features = n_features #features (state variables) = input to the neural n/w, [TOD, In_Temp1, In_Temp2, Out_Temp, Price, Pri
29
30         #User's comfort range
31         self.LT = LT
32         self.UT = UT
33
34         self.dfinput = pd.DataFrame()
35
36         #list to catch various values
37         self.Tout=[] #outside temperature
38         self.Rhout=[] #outside humidity
39         self.Tin1=[] #zone 1 temperature
40         self.Rh1=[] #zone 1 humidity

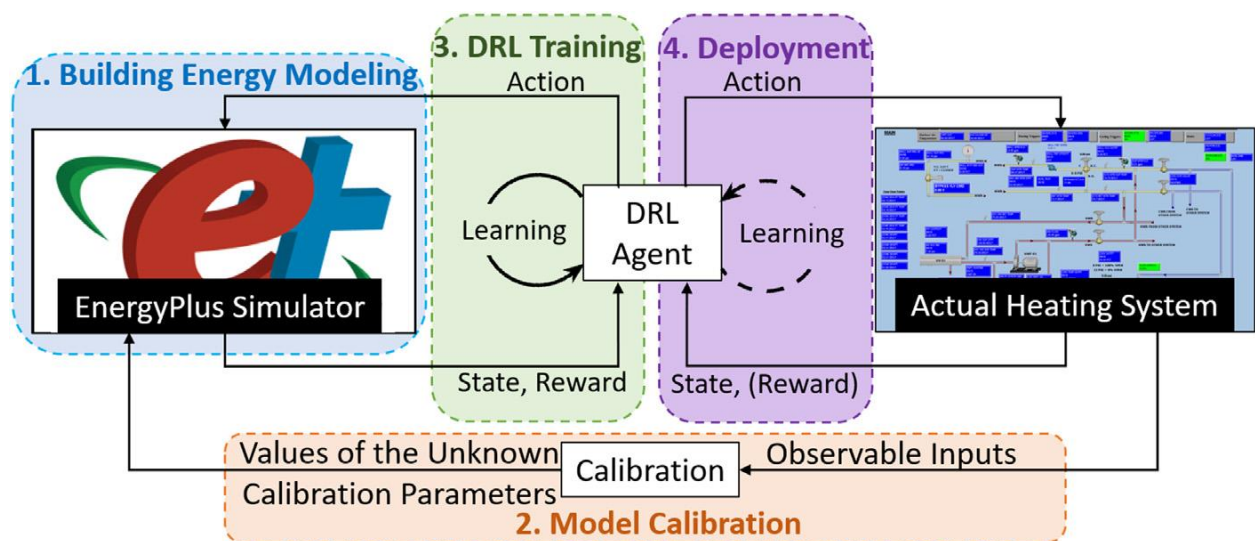
```

شکل 4- بخشی از برنامه پایتون نوشته شده برای تعریف State ها



شکل 5- تعریف کتابخانه ها و ماژول هایی مورد استفاده در برنامه main

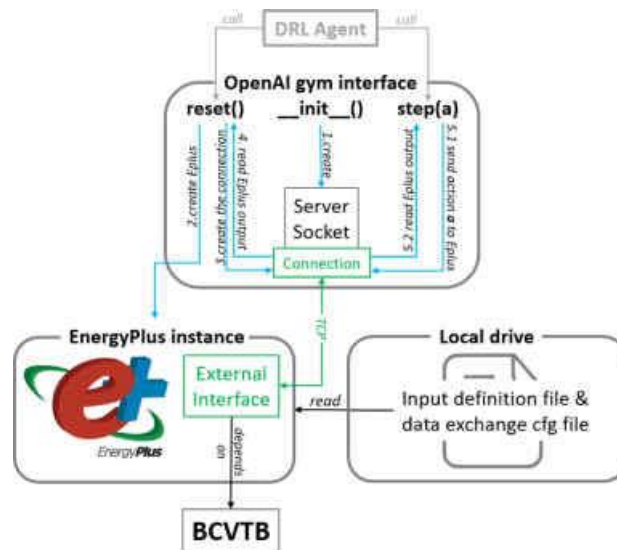
شکل زیر framework کنترلی پیاده سازی شده را نشان می دهد که از 4 مرحله تشکیل شده است. در مرحله اول با استفاده از نرم افزار مدل سازی Design builder مدل سازی انرژی ساختمان انجام می شود. در مرحله دوم که کالبراسیون مدل نام دارد، با استفاده از اطلاعات واقعی ساختمان خروجی شبیه سازی در صورت نیاز بهینه می شود. در مرحله سوم که فاز آموزش DRL نامیده شده است، با استفاده از نتایج شبیه سازی سیستم RL آموزش داده میشود تا بتواند سیاست بهینه کنترلی را فرا بگیرد. در مرحله آخر که Deployment نامیده می شود، سیستم واقعی تحت تست قرار میگیرد.



شکل 6- Framework کنترلی اجرا شده

برای اجرای سناریوی توضیح داده شده، از سیمولاتور EnergyPlus برای RL (EPRL)¹ استفاده شده است. این سیمولاتور از طریق اینترفیس OpenAI Gym دیتای خروجی شبیه سازی ساختمان (EnergyPlus) را میگیرد و با استفاده از آن یک سیاست کنترلی بهینه را یاد میگیرد. سیمولاتور EPRL از تابع ExternalInterface استفاده میکند.

به عبارتی اینترفیس OpenAI gym یک object پایتون با سه تابع `_init_`، `reset` و `step` است. ابتدا این object با فراخوانی تابع `_init_` شروع به کار میکند و یک سوکت سرور ایجاد میکند. تابع `reset` توسط RL agent که در واقع در کنترلر هوشمند پیاده سازی شده است فراخوانی میشود تا پروسه یادگیری آغاز شود. در این زمان یک instance از EnergyPlus با استفاده از فایل IDF و `cfg` که در درایو محلی ذخیره شده است، ایجاد می شود. در مرحله بعد gym object یک ارتباط TCP برای instance ایجاد شده از EnergyPlus برقرار میکند و سپس gym object برای خواندن مقدار اولیه شبیه سازی (خروجی شبیه سازی EnergyPlus) از این ارتباط TCP استفاده می کند و دیتای خوانده شده را برای RL agent می فرستد تا State و Reward را استخراج کند. RL agent در هر بازه زمانی تعریف شده، تابع `step` را فرا میخواند. در این هنگام gym object با استفاده از ارتباط TCP، خروجی تابع `step` که در واقع action می باشد را برای instance ایجاد شده از EnergyPlus می فرستد و نتیجه خروجی شبیه سازی را می خواند و مجدداً آن را به عنوان داده خام برای RL agent می فرستد و این داده برای استخراج State و Reward توسط RL agent پردازش می شود. شکل 7



شکل 7- معماری روال یادگیری آفلاین

¹ EnergyPlus simulator for reinforcement learning

نتایج حاصل از اجرای الگوریتم یادگیری در بیمارستان برکت

برای تست الگوریتم فوق یکی از فضاهای پروژه بیمارستان برکت لحاظ شده است. با استفاده از نرم افزار Design Builder مدلسازی انجام شده است. پارامترهای شبیه سازی در شکل های زیر آورده شده است.

Cooling Design Weather (0.4% based on dry-bulb temp.)	
Max DB temperature (°C)	38.7
WB temperature at Max DB (°C)	18.6
Min night design temperature (°C)	27.9
Cooling Design Weather (1% based on dry-bulb temp.)	
Max DB temperature (°C)	37.2
WB temperature at Max DB (°C)	18.3
Min night design temperature (°C)	26.4
Cooling Design Weather (2% based on dry-bulb temp.)	
Max DB temperature (°C)	36.2
WB temperature at Max DB (°C)	18.0
Min night design temperature (°C)	25.4
Cooling Design Weather (0.4% based on wet-bulb temp.)	
Max DB temperature (°C)	32.7
WB temperature at Max DB (°C)	22.2
Min night design temperature (°C)	21.9
Cooling Design Weather (1% based on wet-bulb temp.)	
Max DB temperature (°C)	33.2
WB temperature at Max DB (°C)	20.7
Min night design temperature (°C)	22.4
Cooling Design Weather (2% based on wet-bulb temp.)	
Max DB temperature (°C)	33.1
WB temperature at Max DB (°C)	19.7
Min night design temperature (°C)	22.3
Cooling Design Weather (0.4% based on dew-point temp.)	
Max DB temperature (°C)	30.5
WB temperature at Max DB (°C)	18.2
Min night design temperature (°C)	19.7
Cooling Design Weather (1% based on dew-point temp.)	
Max DB temperature (°C)	28.6
WB temperature at Max DB (°C)	16.1
Min night design temperature (°C)	17.8

➤ For external floor and roof

U-Value surface to surface (W/m ² -K)	0.164
R-Value (m ² -K/W)	6.249
U-Value (W/m ² -K)	0.160

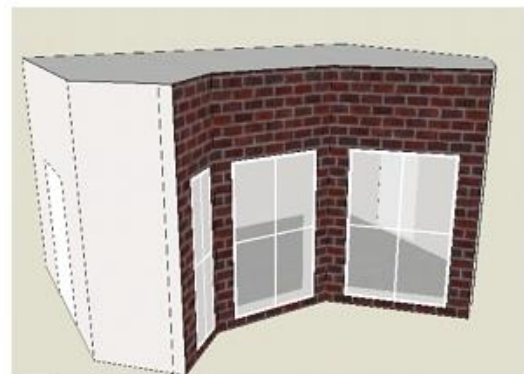
➤ For external wall

U-Value surface to surface (W/m ² -K)	0.258
R-Value (m ² -K/W)	4.011
U-Value (W/m ² -K)	0.249

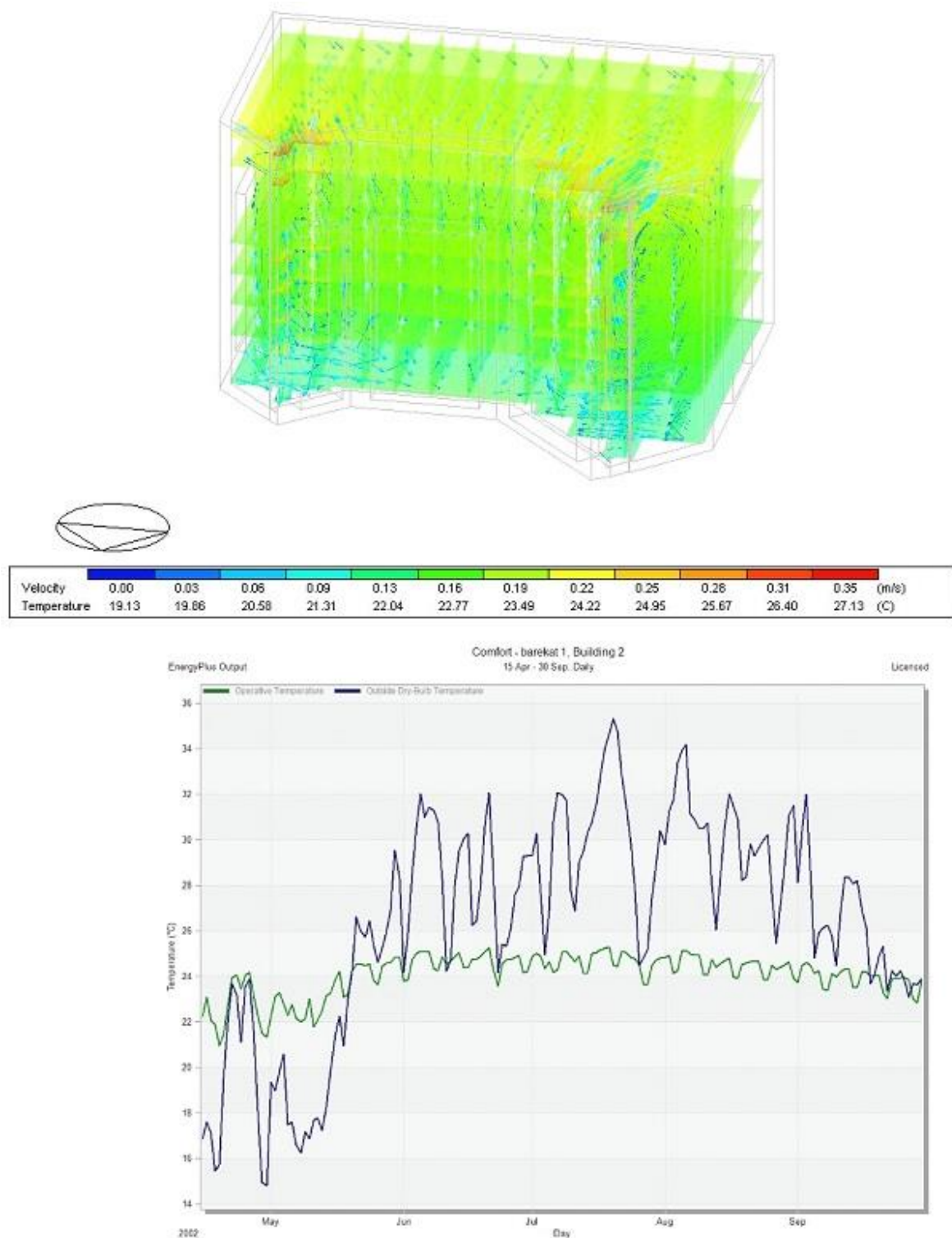
➤ For external Window (3 layer)

Total solar transmission (SHGC)	0.461
Direct solar transmission	0.377
Light transmission	0.637
U-value (ISO 10292/ EN 673) (W/m ² -K)	1.869
U-Value (W/m ² -K)	1.791

نقشه layout فضای موردنظر در شکل زیر آورده شده است.



خروجی شبیه سازی برای دما در 6 ماه و نیز CFD در شکل های زیر آورده شده است.

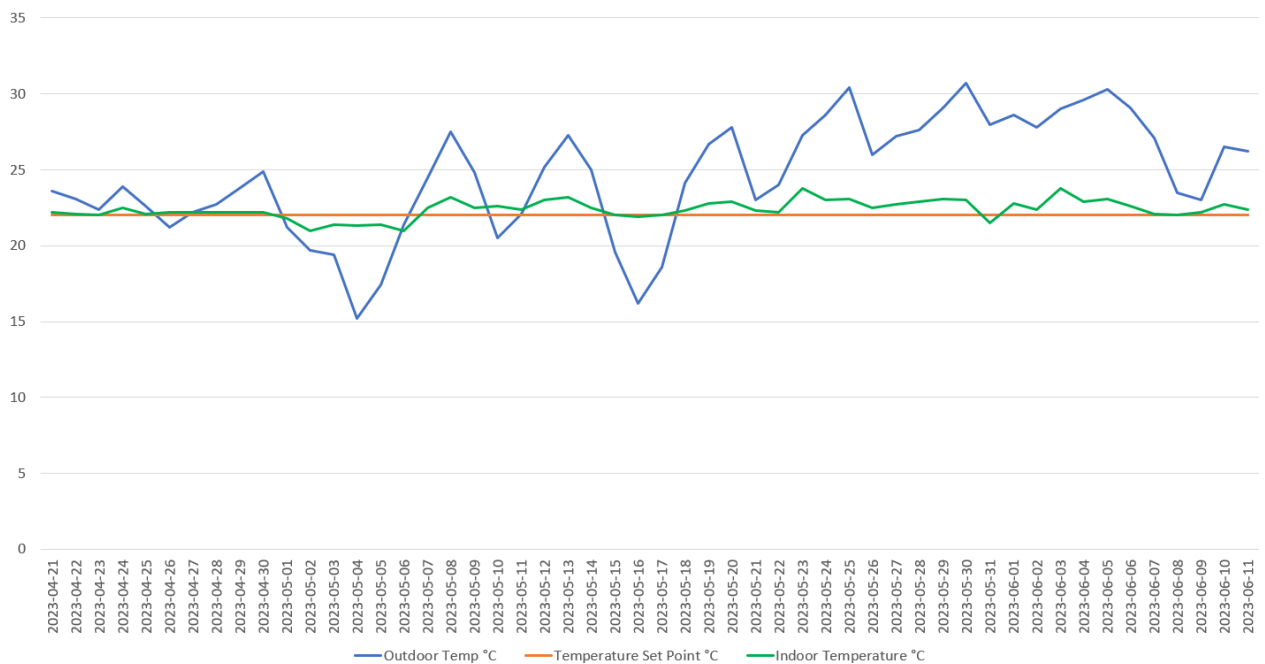


نتایج حاصل از الگوریتم یادگیری برای این فضا که با استفاده از یادگیری آفلاین انجام شده است در ادامه آورده شده است.

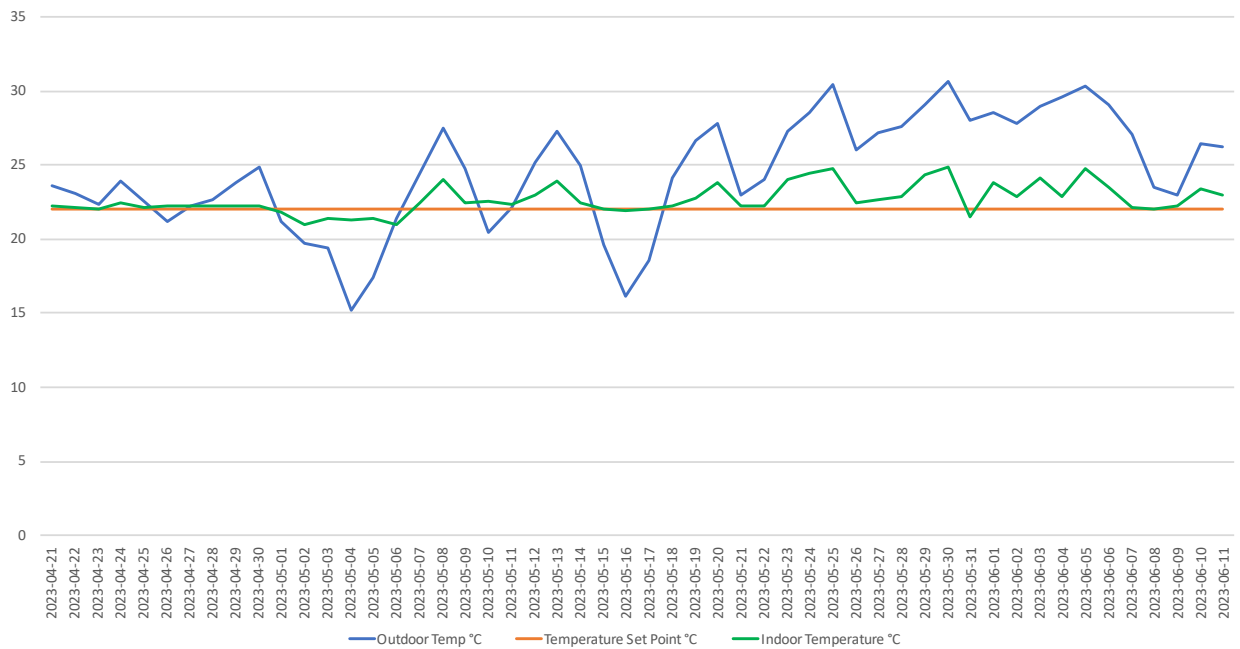
در نمودار اول که دمای بیرونی و داخلی فضای موردنظر را در بازه زمانی اول اردیبهشت تا 21 خرداد را نشان میدهد، مشاهده می شود، با برنامه ریزی زمان روشن شدن سیستم، دمای آسایش حرارتی که روی 22 درجه تنظیم شده است، قابل حصول می باشد. منحنی آبی دمای بیرون (با استفاده از سایت هواشناسی) و منحنی سبز دمای داخل را بر حسب درجه سانتی گراد نشان میدهد. دمای آسایش محیط منحنی نارنجی می باشد که روی 22 درجه تنظیم شده است.

نمودار دوم نشان دهنده سیستم بدون تنظیم شروع به کار هوشمند می باشد که همان طور که از منحنی مشاهده می شود، سیستم در روزهای پیک بار حرارتی از نقطه آسایش حرارتی دور است. به عنوان نمونه در 30 خرداد که دمای بیرون 30.7 درجه سانتی گراد می باشد، با تنظیم زمان روشن شدن سیستم، دمای داخل 23 درجه (نمودار شکل اول) و بدون تنظیم زمان روشن شدن سیستم دمای داخل 24.9 درجه (نمودار شکل دوم) قابل مشاهده است. در واقع سیستم در نمودار اول به عنوان مثال، با شناسایی روزهای پیک بار، سیستم سرمایشی را بجای ساعت 8 صبح، ساعت 6:45 روشن می کند تا بار حرارتی را جبران نماید.

(21 April-11 June)- Temp-With adjusting the start of time of HVAC

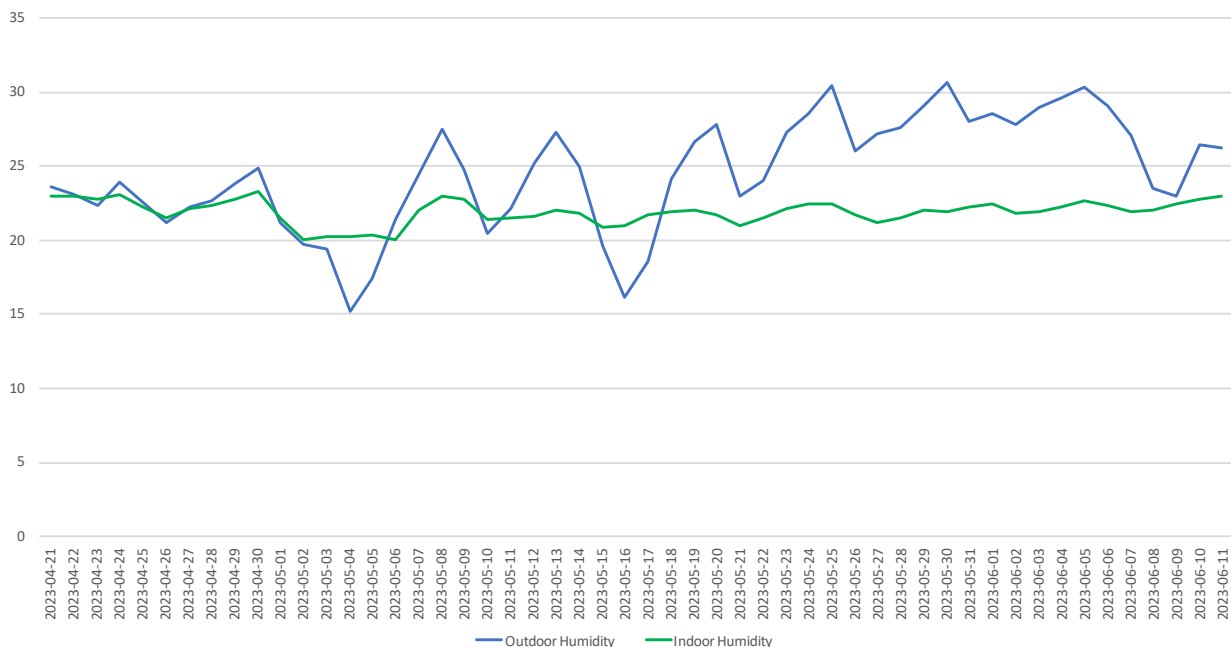


(21 April-11 June)-Without adjusting the start of time of HVAC



منحنی زیر رطوبت نسبی فضای بیرونی ساختمان (منحنی آبی) و رطوبت نسبی فضای داخلی ساختمان (رنگ سبز) را نشان می دهد. همان طور که مشاهده میشود رطوبت حدود 22 درصد است.

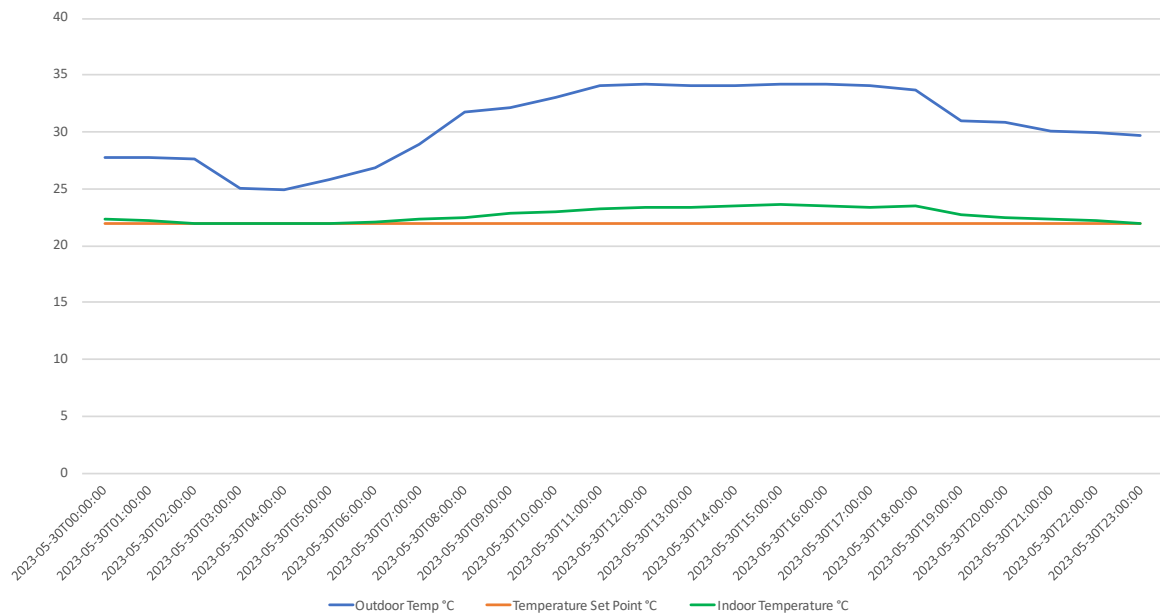
(21 April-11 June)- Relative Humidity %



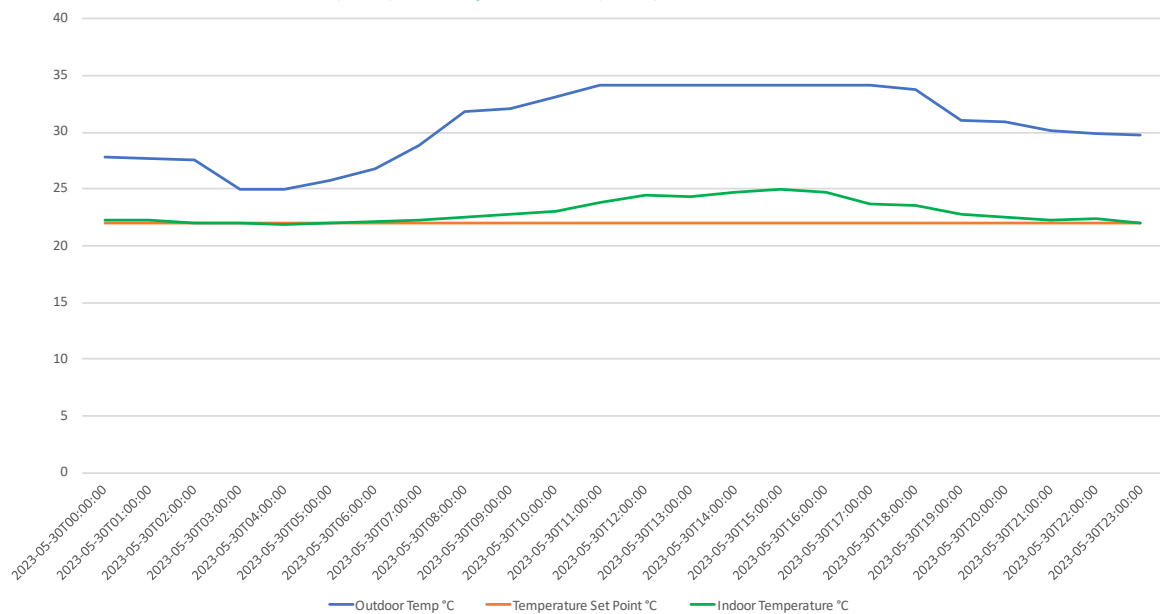
نمودارهای زیر روز پیک بار حرارتی 9 خرداد را نشان می دهند. نمودار اول با تنظیم هوشمند شروع به کار سیستم سرمایشی و نمودار دوم بدون تنظیم شروع به کار هوشمند سیستم می باشد. منحنی آبی در دو نمودار دمای بیرونی و

منحنی سبز دمای داخلی ساختمان را نشان می دهد. در روز فوق سیستم سرمایشی زودتر شروع به کار کرده است تا بتواند پیک بار حرارتی بین ساعت های 11 تا 6 را تامین کند.

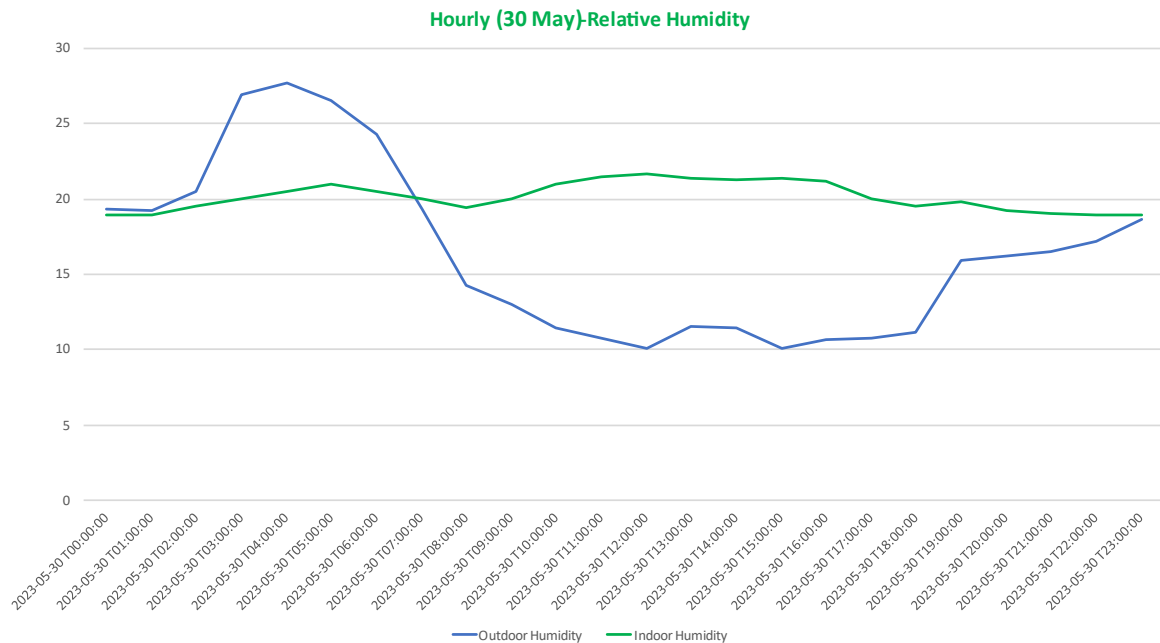
Hourly Temp(30 May)-With adjusting the start of time of HVAC



Hourly Temp (30 May)-Without adjusting the start of time of HVAC



منحنی رطوبت نسبی بیرون و داخل ساختمان در روز پیک بار حرارتی 9 خرداد در نمودار زیر نشان داده شده است. رطوبت نسبی داخل ساختمان حدود 22 درصد می باشد.



در صورت مشاهده پیک بار حرارتی در زمان خاصی از روز به عنوان نمونه ساعت حدود 11 تا 1 اتاق جلسه، (در نمودار شکل زیر قابل مشاهده است) سیستم با توجه به شناسایی پیک بار و علم بر اینکه دمای setpoint فعلی قادر به تامین بار حرارتی نمی باشد، بصورت هوشمندانه setpoint فضای مورد نظر را نیم ساعت قبل از زمان پیک پایین تر آورده تا سیستم سرمایشی بتواند مازاد بار حرارتی را جبران کند و مجدد بعد از زمان پیک، setpoint را به مقدار قبلی برگرداند.

در نمودار زیر منحنی بار سرمایش با رنگ سبز، دمای فضای موردنظر با آبی و دمای setpoint با رنگ نارنجی نشان داده شده است. دمای setpoint در بازه زمانی 11 تا 1 بصورت هوشمند یک درجه کاهش پیدا میکند تا بار حرارتی سرمایشی مازاد که حدود 1.5 کیلو وات می باشد را جبران نماید.

