POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Optimal Management of community Demand Response

Master degree in Electrical Engineering

Ahmed Abdelrahim Mohamed Talha

Leiria, November of 2021

POLITÉCNICO DE LEIRIA | ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO

Optimal Management of community Demand Response

Master degree in Electrical Engineering

Ahmed Abdelrahim Mohamed Talha

Dissertation under the supervision of Professor Luís Neves

Leiria, November of 2021

Originality and Copyright

This dissertation report is original, made only for this purpose, and all authors whose studies and publications were used to complete it are duly acknowledged.

Dedication

I dedicate this dissertation work to the most important people in my life, my family and many friends, with special thanks and gratitude to my late mother, Dr.Buthaina, and my father, Dr.Abdulrahim Talha, for their unending love and encouragement; you will always be my biggest inspiration; to my lovely sisters Dr.Doaa, Dr.Alaa, Dr.Rawan, and the young Buthaina and Mohamed, thank you for everything you have done, and thank you for the support, to my grandmother and second mother who raised me well Thuraya, Without your ongoing support and assistance, my dissertation would not be possible.

I'd also like to dedicate this work to a number of people who have passed away, but whose memory will go on forever; as long as I live, you will be remembered.as long as I live you will be loved. Thank you to all my friends who have supported me throughout the process. I want to express my appreciation and gratitude to all of my teachers and professors; I will always appreciate what you have done for me. Lastly, this work is also dedicated to all “Decemberian” heroes, who had high hopes and dreams.

Acknowledgments

I'd like to thank my supervisor (Prof.Luís Neves) for making this work possible. His guidance, continuous support, encouragement, and patience guided me through this research. I would also like to thank the Laboratory for Advanced Computing at University of Coimbra (LCA) for allowing me to use their resources to perform my simulations. I would like to thank Polytechnic of Leiria for providing me with this wonderful opportunity. Finally, many thanks to the creators of this framework for their brilliant idea and for helping other students to experience this wonderful branch in engineering and science. This work was supported by projects UID/MULTI/00308/2017, and MAnAGER (POCI-01-0145-FEDER-028040).

Abstract

More than one-third of the electricity produced globally is consumed by the residential sectors [1], with nearly 17% of CO2 emissions, are coming from residential buildings according to reports from 2018 [2] [3]. In order to cope with increase in electricity demand and consumption, while considering the environmental impacts, electricity providers are seeking to implement solutions to help them balance the supply with the electricity demand while mitigating emissions. Thus, increasing the number of conventional generation units and using unreliable renewable source of energy is not a viable investment. That's why, in recent years research attention has shifted to demand side solutions [4]. This research investigates the optimal management for an urban residential community, that can help in reducing energy consumption and peak and CO2 emissions. This will help to put an agreement with the grid operator for an agreed load shape, for efficient demand response (DR) program implementation. This work uses a framework known as CityLearn [2]. It is based on a Machine Learning branch known as Reinforcement Learning (RL), and it is used to test a variety of intelligent agents for optimizing building load consumption and load shape. The RL agent is used for controlling hot water and chilled water storages, as well as the battery system. When compared to the regular building usage, the results demonstrate that utilizing an RL agent for storage system control can be helpful, as the electricity consumption is greatly reduced when it's compared to the normal building consumption.

Keywords: Demand Response (DR), Demand Side Management (DSM), Demand Aggregation (DA), Reinforcement Learning (RL), Optimization, Load Management.

Contents

List of Figures

List of Tables

List of Abbreviations and Acronyms

AS	Ancillary Services
CMP	Capacity Market Programs
CPP	Critical Peak Pricing
DB	Demand Bidding
DER	Distributed Energy Resources
DLC	Direct Load Control
DL	Deep Learning
DR	Demand Response
GHG	Green House Emissions
DSM	Demand Side Management
ED-CPP	Extreme Day Critical Peak Pricing
EDP	Extreme Day Pricing
EDRP	Emergency Demand Response Program
ESS	Energy Storing System
IBP	Incentive-Based Program
ML	Machine Learning
PBP	Price Based Program
RBC	Rule Base Controller
RL	Reinforcement Learning
RTP	Real Time Pricing
SAC	Soft Actor Critic
TOU	Time Of Use
VOLL	Value of lost load

1 Introduction:

Energy demand has increased in the last two decades [5]. Owing to a higher level of comfort. Which translates to an increase in energy consumption [6]. Keep in mind that, increased consumption means more CO2 emissions. Which is in conflict with the global goal of reducing greenhouse gas emissions (GHG) in general. The EU set a target to reduce GHG by 55 percent by 2030 compared to the levels of 1990 [7], and to achieve net zero greenhouse gas emissions by 2050. Which can place a significant societal and regulatory burden on the power industry. Because they need to decrease GHG emissions while taking increased demand into account.

According to [5], residential energy demand has risen steadily in recent decades. Posing a new challenge for electricity providers. One solution is to increase generation levels [8], by adding more conventional generation units. Which can help meet demand during on-peak periods. But the main issue, aside from the environmental impact, is that most of these units will be idle during off-peak periods, which is not a viable investment. Another option is to adapt renewable energy sources to meet this demand. But due to their intermittent nature, it is difficult to rely on their output. Necessitating the need for a stable and reliable source of energy to deal with system stress during on-peak periods.

Demand response (DR), with the huge advancement in smart metering infrastructure, has become an important and vital part of energy planning [9]. Because it can help in reducing demand, and it can help both customers and utilities reduce their energy price volatility. Besides offering a variety of operational and economic benefits, DR can offer wide-range market benefits, such as lowering wholesale electricity prices. Because it averts the need for high-cost conventional generation units. DR is considered a tool that can transform customers from non-responsive to responsive, and interactive customers, which can be considered as an extra flexibility source for the utility.

According to [10], DR can increase overall efficiency and energy utilization. Given that residential demand accounts for 30 percent to 40 percent of total energy demand globally [8], the need for energy efficiency applications, and load optimization has increased recently. An efficient use of energy can reduce a large part of demand and load consumption. It is referred to as "the fifth fuel" by [11], because it comes after coal, natural gas, nuclear power, and renewable energy. According to some energy experts, energy efficiency and load optimization are the "first fuel" because of their large influence on reducing demand and their cheap social costs [11].

Machine Learning (ML) has grown in popularity in recent years due to its ability to solve a variety of problems. In the energy sector, large corporations have adopted ML. In 2014, Google adopted a framework based on ML and AI called "DeepMind" and used it for energy optimization. According to [12], the cooling energy demand for its data centre was reduced by 40 percent as a result of using the framework. This research aims to manage a group of residential customers for use in a DR program. It does so by utilizing a ML branch known as Reinforcement Learning (RL) to optimize the customer's load shape and present it to the grid. Five chapters comprise the work: The first chapter contains an introduction, the second chapter contains a literature review and background information on DR, DR aggregation, and RL, as well as related papers and investigations developed by the authors in the literature. The third chapter contains the methodologies section, which discusses the procedure and technical review, the fourth chapter contains the case study and results section, which shows and analyses the customers' load diagrams, and the fifth chapter contains the conclusion and some advice for future work development.

2 Literature review:

2.1 Background:

2.1.1 Demand side management:

Since the electrical market was restructured from being vertically integrated to open market system, which is known as bidirectional power flow market. The system operation philosophy has evolved as well. From meeting energy and power demand whenever it occurred, to meeting energy and power demand while minimizing system fluctuations. These fluctuations typically refer to the balance between supply and demand. Which can be influence by a variety of factors, including generation, transmission, and distribution outages, and the constant change in electricity demand. Historically, the change in demand was the supplier's responsibility for a long period of time [13]. But after restructuring the electrical market, a new strategy for operation was introduced. The new strategy aims to reduce demand through the implementation of various managerial measures that will reduce demand and make electricity more efficient [14] [15]. These types of measures are known as demand side management (DSM). According to [16], the massive evolution of communication infrastructure, the unstable price of energy, and the oil crises are cited as the primary motivation and inspiration for these measures. Although the oil crises were mentioned as a rationale for moving toward this way of operation, it is vital to note that DSM was founded and developed in the 1970s, well before the oil crisis [17]. DSM is credited with significantly improving energy efficiency, utilization and sustainability.

Climate change mitigation has gained momentum in recent years, and there is widespread concern about lowering global carbon emissions. According to the Environmental Protection Agency in the US [18], greenhouse gas emissions (GHG) are described as the gases that trap heat in our atmosphere and are directly accountable for climate change. The primary source of GHGs is the power and energy generating sector [19]. And for all of these reasons combined, many policies and summits were introduced in order to reduce the carbon emissions, by the utilization of renewable energy sources (RESs).

However, it is critical to understand that RES sources are intermittent, which poses a risk to system operation. Therefore, in most cases, they view energy storage systems (ESS) as a complementary tool to RES operations, storing energy during off-peak hours, and the excess amount of energy, and then releasing it when needed. However, ESSs are not economically viable, as they require high investment costs and electrical energy cannot be stored on the scale required by large power systems [20]. Not to mention a complicated connection with the current (old) grid is required to ensure the successful deployment of distributed energy sources (DERs) in an efficient and cost-

effective manner [21]. As a result of these reasons, energy planners and decision makers began to rethink their alternatives and to prioritize demand side solutions above generating side solutions (like conventional storage, ESSs, increasing the generation capacity).

In DSM programs, customers will perform multiple actions in response to a signal issued by the system operator. They will arrange their consumption in an efficient manner in order to flatten the load curve and match demand to available supply. These programs are mainly classified as follows:

- **Load growth:** Load growth programs are those that encourage customers to consume more electricity during times of excess capacity. They are especially prevalent in areas that is supplied by wind power. But the new definition of DSM provided by [22] [23], describes the DSM as a tool to reduce the total demand, this means load growth programs can be excluded.
- **Energy saving:** It is a program designed to encourage customers to minimize and regulate their energy use via the use of precise measurement and control equipment [23]. Energy saving programs include those aimed at increasing energy efficiency. Energy efficiency programs can be defined as a collection of measures taken by a customer or even a municipality to minimize energy consumption. These efforts attempt to reduce load losses through the use of more efficient appliances and devices and the replacement of outdated equipment with newer ones [24]. The estimation for the amount of energy consumption reduction in the US is around 15-25 percent for water pumps, and 40-70 percent for street lighting, and 20-30 percent office building [25] [23].
- **Demand response:** DR programs are a method of altering consumption patterns. Often, this occurs in response to an increase in electricity prices or change in incentives by the operator [23]. In this research, and this chapter specifically the focus will be on DR.
- **Demand shifting:** Is a program that is used to increase the reliability of the power supply. The operator typically defines a threshold or a level of load demand. If the load demand exceeds this level, it is clipped and shifted to another period. This typically occurs during a peak period, and the load is shifted to an off-peak period. The conventional form of demand shifting is known as preventive load shifting (PLS), and it is when the load demand is shifted in an emergency and continuous manner. The other sort of load demand shifting is referred to as corrective load shifting (CLS), and it is utilized when the available capacity of the power supply is insufficient [26] [27].

DSM was developed in the late 1970s in the United States. It began as demand side load management [17], with the primary goal of peak shaving and load management [23]. Peak shaving is utilized in the application of energy storage to avoid the installation of additional capacity to meet

peak demand [28]. However, other applications have been added to the DSM programs since then to ensure client convenience and to prioritize it. According to [23] a successful DSM it is the one that is structured to make the customer comfort intact. Figure (2.1) illustrates all DSM programs, and different programs of DR and it is adapted from [23] by applying modification to the original figure. The original figure contains virtual power plant (VPP) as a tool for DSM. VPP is basically a portfolio of generation and renewable energy sources, and it is a program for the generation side, but I Performed modification to the figure and excluded it, because it is a program for the generation side. However, VPP can be used also in the demand side, as a tool and a program that provide system flexibility by offering flexible loads from its portfolio, to act as a DR aggregator.

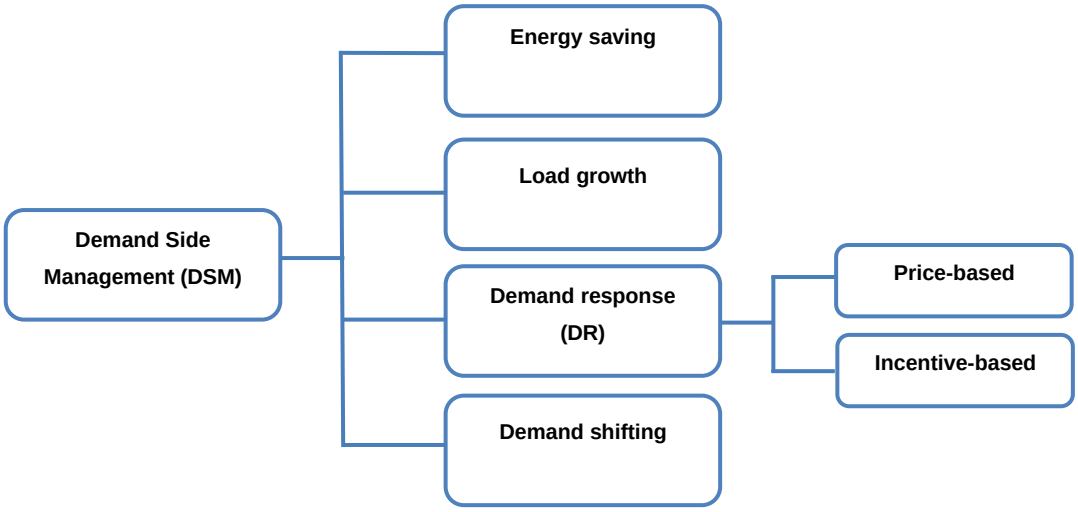

Figure 2.1 – Different operations of Demand Side Management

2.1.2 Demand response (DR):

2.1.2.1 Definition and types:

Demand response (DR) is defined as the change in the electricity usage by the customers, due to a change in electricity prices [29]. According to the US Department of Energy [30], DR is defined as a program or tariff that is implemented to incentivize changes in consumption patterns as a result of a change in electricity prices, when grid reliability is jeopardized. There are two primary types of DR: Incentive-Based Programs (IBP) and Price-Based Programs (PBP) [29]. It can be also named as system-led and market-led programs [31], or emergency-based and economic-based programs [32] [29], or stability and economic-based, and sometimes it is called reliability-based and price-based DR programs in the literature [32]. DR programs can be named differently, although all names are referring to the same type of programs. In this research we will use the term incentive-based (IBP) and price-based (PBP).

IBPs are classified into two broad categories: the Classical-IBP, which provides discounts or credits in exchange for participation, and the Market-IBP, which compensates customers for load reduction. The classical-IBP category includes two distinct types of programs: Direct Load Control (DLC) programs, in which power utilities have the ability to immediately shut down customers' equipment on short notice. Customers can be commercial or residential. And Interruptible/Curtailable Load programs, in which customers must reduce their load consumption to a predefined value or face penalties [29]. As mentioned in [33], in load curtailment, electricity consumption is reduced to pre-agreed levels in exchange for a benefit offered by the utility. Customers are typically commercial and industrial buildings. This typically occurs during the summer and can be extremely beneficial to customers, resulting in cost savings. Market-IBP, on the other hand, is separated into the following categories: Emergency Demand Response Programs (EDRP), Demand Bidding (DB), Capacity Market Programs (CMP), and Ancillary Services (AS).

In DB (also known as Buyback) programs, customers bid on load reductions, but the bid price must be lower than the wholesale market [29]. According to [34] DB programs target large customers, where customers define their own bid. If the bid is accepted, the customer must reduce his/her load by the amount specified in the bid. If the customer succeeds, the customer is rewarded. Otherwise, the customer faces penalties. Consumers will be compensated for reducing their loads during emergency conditions, therefore this is essentially a scheme that allows load bidding for customers only during emergency situations [35]. The Capacity Market Programs are utilized when a system crisis occurs (contingency situation), customers receive a day-ahead notice and are required to lower their loads to predefined levels [29], CMPs are regarded an efficient technique to assure supply security and to reduce residual peak loads by incentivizing consumers to act during a contingency [36].

Customers will bid for load curtailment in AS market programs. If their bid is accepted, they will act as an operating reserve for the utility and will be compensated at the spot market energy price if load curtailment is required [29]. The current AS can be frequency control services, which are used to restore nominal frequency levels following a deviation caused by supply and demand imbalances, or voltage control and reactive power supply services. Grid restoration is also considered a type of AS because it is provided by generation units following a system shutdown, this service is also known as black start operation. Keeping in mind that all of the aforementioned services reflect traditional AS. New types of services such as real power ramping, inertial response, and frequency response have emerged lately [37] [29].

The second type of DR is based on variable pricing and is referred to as Price based Programs (PBP). PBPs are divided into static and dynamic pricing. Static pricing system refers to a pricing method, where prices are predefined and remain constant for a specified period of time. Dynamic

pricing can be defined as a method of representing electricity tariffs other than flat rates and can be an effective method of representing the fluctuations in electricity prices over different time intervals [29]. According to [38], dynamic pricing can be defined as a method of disclosing the price of electricity prior to consumption, and it can be a critical tool for DR aggregation. Dynamic pricing includes a variety of different rates. Its main goal is to flatten the load demand curve by charging a premium for electricity during peak hours and charging less during off-peak hours. PBPs have a variety of rates, these rates include Time of Use (TOU), Critical Peak Pricing (CPP), Extreme Day Pricing (EDP), Extreme Day Critical Peak Pricing (ED-CPP), and Real Time Pricing (RTP).

The most fundamental sort of PBP is TOU rates, which are essentially a representation of the average rate during different time periods [29]. According to [39], It is commonly referred to as time-varying rates and refers to a sort of rate that the utility can establish and alter during the day, and it can fluctuate according to seasons, weekends, and holidays. There are two types of tariffs: static and dynamic. A static TOU price signal is fixed and predefined and remains constant throughout the day. A dynamic TOU price signal changes according to system conditions. In TOU, the day is divided into different time intervals: off-peak, on-peak, and occasionally mid-peak. The price of electricity is set for each time interval, with on-peak representing high demand and off-peak representing low demand. For an example of this rate see appendix A.

In dynamic TOU, the prices can fluctuate every hour or every quarter-hour or even less. The main difference between this type and the previous one (static TOU rate) is that in the static TOU, the periods and electricity prices are fixed and predefined. Whereas in the dynamic TOU, the on-peak and off-peak intervals can change regularly to simulate and reflect an accurate state of the energy market. In TOU, prices are higher during peak hours compared to off-peak hours. For example, when customers return home from work in the evening and begin turning on equipment (dishwasher, air conditioner ...etc.), this can result in increased demand on the power utility, resulting in higher prices during this time period. Typically, customers who own solar systems will rely on their solar system to compensate for the higher electricity prices during this time period [29].

CPP is a type of rate that is used to reduce load during high-cost hours. It provides customers with a year-round rate, and several studies demonstrate that residential users reduced load consumption using CPP more than any other TOU rate [40], CPP is considered a powerful rate for DR, particularly for small consumers. It can be an effective tool for reducing transmission and distribution congestion, as well as the need for Peaker plants to balance load and demand during peak hours [41]. CPP events are not frequent, and mostly designed in a dynamic manner. But CPP rates can be also superimposed on static TOU or standard flat rates [29]. CPP is often employed in the case of a system failure or when wholesale electricity prices are very high for a short number of days.

During event days, users are rewarded for every kilowatt reduction they make and punished for excessive energy consumption [41].

Another program similar to CPP is EDP. Both programs have higher electricity rates for a specified date or time period, but the difference is that EDP participants will be notified a day in advance, and the EDP price rate will be in effect for the entire 24 hours. Unlike CPP, which typically lasts only a few hours [29]. The inclining (or increasing) block rates are price structures in which the system charges a higher price for larger quantities of a commodity, in electricity market, the inclining block rate charges a higher rate per kWh at higher levels of energy usage (and a lower rate at lower levels of energy usage) [49].

Most of the electricity demand occur in just 1% of the hours of a year [42]. Considering how difficult it is to store the electricity in large quantities [20], and the high cost of dispatchable generation, particularly during peak hours, and the available generation that sits idle during the off-peak period. The supply cost of electricity is not constant and variable all the time, and only a few consumers notice this variation in electricity supply costs. This can result in massive amounts of over- and under-consumption during peak hours, and economic inefficiency during off-peak hours [43].

One option to combat this inefficiency is to implement a more time-varying pricing system, one that reflects the actual cost of electricity during usage. According to [43], the primary disadvantage of relying on time-invariant pricing systems is that they allow customers to consume electricity regardless of when they do so. One proposed solution is the Real-time pricing (RTP) mechanism, because it reflects the true wholesale price of electricity. RTPs are a category of DR programs that represent the real price of electricity in the wholesale market. Customers are notified of pricing on a day-ahead, hour-ahead, or even a few minutes-ahead basis through these programs [29]. RTP necessitates advanced metering infrastructure in order to facilitate DR. It is also widely regarded as the ideal method for usage in competitive power markets [29]. RTP can act as a link between the wholesale and retail markets, increasing price sensitivity and resulting in a more efficient allocation of resources and energy [44].

Numerous advocates have argued for the establishment of a system that represents the real cost of energy production in real time [45], and numerous research and studies have examined the genuine potential of RTP [46] [47]. RTP may assist in improving system efficiency and lowering emissions. Many environmental groups favour the implementation of RTP because it can help mitigate risk and harm to market power, as well as boost the adoption of more green sources [48]. The adoption of RTP can result in two types of benefits: long-term benefits and short-term benefits. The short-term benefits revolve around changing the generation types and patterns, which may result in reducing the emissions, and nuclear waste production, and fossil fuel usage. The long-term benefits

can include a shift in how production and consumption decisions are made to the reduction of peak load demand [48]. Reducing peak load demand has an environmental benefit since it reduces reliance on fossil fuels, Peaker plants, which may lead to a reduction in emissions caused by fossil fuels [45].

The increase in demand can be reflected in the price of electricity. If the customer is enrolled in an RTP tariff, the customer will face the real wholesale electricity prices. RTP acts as a conduit between the retail and wholesale electricity markets. The sudden increase and fluctuation in demand would undoubtedly affect the retail prices of electricity. Given that the retail prices of electricity are fixed (flat prices), this can cause customers to conserve less than they should during off-peak periods and more than they should during peak periods. Taking a hot day as an example, the wholesale electricity prices are significantly higher than the flat price of electricity. In a case of RTP, this will force customers to face the real wholesale electricity prices, which will cause them to conserve energy usage, which results in the system stability [48]. Currently, RTP is only available to large industrial and commercial users [47].

A case in point of a sudden increase in demand occurred in 2014 during the so-called "polar vortex" cold wave in the United States. The severe cold weather increased demand for natural gas while also increasing electricity demand. This resulted in the shutdown of numerous gas-fired power plants, affecting both supply and price of the service [50]. At a normal market rate, all of these losses would be borne by utilities and producers, which would be economically unviable for them and will result in increase of service prices for the customer.

Historically, electricity prices were flat and static, but dynamic pricing systems have begun to take over. Flat pricing systems mean that electricity prices do not reflect the true value of electricity at the time of purchase. This can increase customer comfort, but it can result in significant losses for energy retailers and producers. One of the reasons why flat rates formerly dominated, is because most policymakers saw flat rates as a mechanism to smooth out the market volatility [45]. Volatility of pricing may be described as the unpredictable fluctuations that occur over time in a process. In economics, it refers to a standard that is used to examine and assess the risk associated with owning an asset whose future is unclear [51]. Flat rates were seen as a tool to protect customers from potential bill shocks and to provide more price stability for low-income customers. The main impediment to implementing dynamic pricing was uncertainty about the ability of responding to dynamic price signals. As there are numerous concerns about customers' (residential customers') ability to respond to a DR event price signal, as well as numerous concerns about customer inconvenience when using dynamic pricing methods [45].

However, PBP’ dynamic pricing became a reality recently, and the concerns surrounding it has gone according to [46]. To substantiate this claim, a report published in 2020 stated that the number of customers enrolled in dynamic pricing programs increased by 311,300 in comparison to

2018, and the smart meter penetration was equal to (37.8 percent) in residential sector in 2018, and equal to (56.7%) in 2020 [52].

2.1.2.1.1 The costs of DR:

The cost of DR programs can be divided into two main types of costs [53]:

- Costs for participants
- Costs for the utility

The participant cost is divided into [53]:

- Initial costs
- Events-specific costs

Participants initial costs can be divided into [53]:

- Enabling the technology investments
- Enabling the response plan or strategy

By enabling technology investments, customers or participants can realize the full potential of DR programs. These technologies can be used to manage, control, and compute the customers' usage, as well as serve as a bidirectional communications link between the customers and the utility. The costs paid by participants will be repaid through incentives offered by the utilities [53]. The other expense is the cost of enabling the response plan or strategy, which can provide participants with technical assistance. The costs associated with certain events can be classified as follows [53]:

- The inconvenience costs
- The lost business costs
- Rescheduling costs
- Participant generator fuel and maintenance costs

The aforementioned costs are classified into two broad categories: financial costs, which include lost business, reschedule costs, and fuel and maintenance of the on-site generation unit. The second type is the abstract costs, or the value of the electricity service. Which can be quantified in terms of the inconvenience caused to customers by the constraints imposed [53]. In any response plan, customer comfort is a critical component, and it is typically quantified by what is known as the Value of lost load (VOLL). VOLL is defined as the amount that end-users are prepared to pay to avoid a disruption in their electricity service [54]. On the other hand, the system costs or the utility costs can be divided into [53]:

- The initial costs

- On-going “running” costs

The initial costs that the service provider must handle can be divided into [53]:

- The communication infrastructure
- Initial training to customers
- The system software

The cost of communications comprises the cost of cables or wireless connections, as well as the cost of connections made through a third-party telecommunications provider. The connection can also give information about rates and limitations [29]. Customer education has the potential to maximize the facilitation of DR's potential, and it must be given significant attention in order to comprehend the customer's wants [55]. The installation of system software and equipment is regarded a prerequisite for participation in any DR program. These costs will be recovered through rate increases on the customer's bill, while the customer education costs will be recovered through rate payers and public benefits sources [53]. The ongoing costs is divided into [53]:

- Marketing costs
- Management costs
- Payment for the participants
- Communications / metering costs
- The program evaluation

2.1.2.1.2 Demand response Benefits:

Due to significant advancements in energy modelling and information technology, DR became an efficient tool for increasing system flexibility, which led in an increase in system efficiency by allowing for more efficient energy use. This is consistent with the requirement for a method or tool to operate as a complement to the DER integration process. As DER, as previously stated, is intermittent [56], and it requires an additional method to ensure successful integration. The use of energy storing systems (ESS) is not considered economically viable. DR, with the flexibility it provides, can act as a tool to meet and fill the fluctuation caused by distributed energy resources (DER), potentially resulting in a higher penetration of these sources [57].

Using wind generation as an example, according to [58] [57], wind generation has two types of costs: essential costs and operating costs. The operating costs represent generation reserves that are used to compensate for the high fluctuations in wind power output. By utilizing the DR flexibility, wind power integration can be facilitated. DR can also create a valley filling effect, by mitigating the effect of the over-generation problems that occur during the night [59]. Keeping in mind that relying on conventional generation units as a source of flexibility is constrained by technical constraints such as ramp rates [59]. Ramp rates are defined as a rate in megawatts per minute that reflects the change

in available resources [60]. Because wind generation requires a large amount of generation reserves to act as a buffer and safeguard against system fluctuations, DR can mitigate these fluctuations and ensure the supply's stability and security through load curtailment and load shifting.

According to [61] responsive load is the most underutilized source of flexibility and can deliver a higher level of reliability than conventional generation. According to [62] the influence of a very small number of conventional generators is thought to be stronger than the effect of a greater number of responsive loads. Another advantage of DR is that the responsive load can provide a more efficient ramping rate than the conventional generator can supply, and the reason for that, is that power consumption from these types of loads can be adjusted instantly [57]. As demonstrated by appliances such as an electric heater or cooler, which produce energy services rather than power, and whose power consumption can be adjusted and shifted to another time with minimal impact on the energy they produce [57].

DR can benefit the entire market by lowering wholesale electricity costs. By removing the need-to-run power plants, which are regarded a pricey source of electricity. DR can assist in lowering the cost of produced energy [63], and it has the potential to generate long-term system benefits by requiring utilities to reduce their capacity requirements [64], which will impact the customer's electricity cost. Bearing in mind that even customers who do not move their loads or alter them in response to a DR event signal are likely to benefit from the overall reduction in electricity prices [63]. Another significant benefit is the reduction of price volatility. Finally, DR can help limit the market's dominance by large players [29]. Figure (2.3) is drawn from [53] and illustrates the broad benefits of DR on participants, the market as a whole, and market performance.

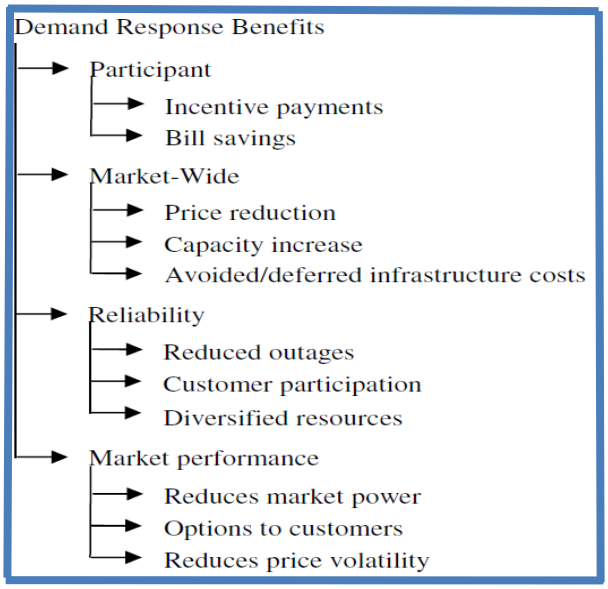

Figure 2.2 – The benefits of Demand Response

2.1.3 DR Aggregation (DRA):

DR Aggregation is described as the process of combining customers, producers, prosumers, and other energy sector participants into a single entity, for the purpose of trading power and selling their services to the system operator [65]. The European Parliament defines an aggregator as “a market participant that combines multiple customer loads or generated electricity for sale, purchase or auction in any organized energy market “ [66]. The aggregated load can act as a source of flexibility, hence enhancing the system's stability and reliability. According to [67], flexibility refers to changes in generation and/or consumption as a result of a price signal. The primary flexibility factors are according to [67]:

- Amount of power modulation
- The duration
- The rate of change
- The response times
- The location

There are numerous ways to provide flexibility, including centralized generation, which represents the current "common" paradigm, in which electricity is generated exclusively at large generation facilities and then distributed to customers via transmission and distribution systems. Decentralized or distributed generation, which represents a paradigm in which electricity is generated near its point of use [68]. Another source of flexibility is energy storage and demand side participation. However, only large customers, such as industrial customers, can provide and sell their flexibility in the flexibility market. However, demand aggregation enables small residential and commercial customers to leverage their flexibility potential.

Aggregators can be retailers or independent aggregators. An independent aggregator is defined as an entity that is not affiliated to a supplier or other market participant [69]. According to [70], the aggregator classifications can vary based on two factors:

- Resources optimization
- The flexibility they offer

Aggregators can be classified according to the resource they are optimizing into:

- Load aggregator
- Demand aggregator
- Production aggregator

The load aggregator is an entity that aggregates various types of loads from end users in order to manage their flexibility. Whereas the demand flexibility entity aggregates various resources

in order to manage the customers' flexibility. Finally, the production aggregator is an entity that utilizes a small number of end user generators in order to participate in a virtual power plant (VPP) and act as a prosumer. The aggregator types are classified according to the degree of freedom they provide [71]:

- An aggregator that consumes resources
- An aggregator that produces resources
- An aggregator with a bi-directional resource

The aggregator that consumes resources aggregates various types of loads and then organizes them to be used as a source of flexibility. While the aggregator that produces resources maintains a portfolio of various types of resources to generate electricity, which may include conventional or renewable generation. The bi-directional aggregator maintains a portfolio with static and dynamic energy storage devices, which are used to increase the portfolio's flexibility.

2.1.4 Integration regulatory barriers for DR and DRA in the EU market:

To maximize the potential of independent aggregators, the European Commission commissioned the Smart Grids Task Force (SGTF) to provide their suggestion on market flexibility. Which they did in 2015 [72], they discussed the role of the aggregator and the barriers it faces. As they noted in their recommendation, aggregation services can be provided by the supplier or by an independent aggregator (third party aggregator). However, in many Member States, the primary issue is that the relationship between the aggregator (as an independent third party) and the other market participants is acrimonious. For example, if the aggregator wishes to offer its services, it must enter into multiple contracts with each consumer [72], with the balance responsible parties (BRP), which represents the parties that is responsible to maintain supply and demand. Another contract needs to be established with the supplier, and yet another with the TSO and DSO. Each of these contracts may contain contradictory requirements, resulting in the aggregator's resources and potentials being blocked.

This issue, in particular, prompted the Smarty Grid Task Force (SGTF) to make a recommendation in its report [72], a fair communication is to be developed between the old and new parties for information exchange. They also proposed that, in order to eliminate entry barriers “an aggregator should never be obliged to negotiate its portfolio with the BRP or supplier of a consumer”, which might result in the aggregator's integration being made easier. On the basis of this report, the European Commission proposed article 17 on DR and aggregation [73], four distinct proposals were made, the first of which included the criterion “(a) the right for each aggregator to enter the market without the consent from other market participants”, this criterion grants the aggregator the ability to enter the market without the consent of any other party.

The second criterion is “(b) the existence of transparent rules that assign clear roles and responsibilities to all market participants”, and third is “(c) the existence of transparent rules and procedures for data exchange between market participants to ensure easy, equal and non-discriminatory access to data while fully protecting commercial data”, which ensures the establishment of a set of regulations that protects different market participants and facilitates the interchange of data between all market participants in a fair and transparent manner. The fourth criterion (d) regarding DR and DRA topics, has caused considerable controversy, as it refers to the aggregator's payment of compensation to the supplier and/or generator, equal to the amount of energy not consumed by the consumer after the aggregator triggered a DR event. The original criterion states “(d) the absence of any requirement that aggregators should pay compensation to suppliers or generators*”*, This sparked a major controversy. What is referred to as the supplier's open energy position [69], this position occurs when an independent aggregator initiates a DR event, which can result in either over-consumption (turn-up) or under-consumption by the customer (turn-down).

If the customer did not consume much electricity (turn-down), this can result in an excessive amount of available and unutilized energy being possessed by the supplier. If the supplier cannot sell back this energy, this results in a revenue loss. The other issue is that the customer is essentially selling unused energy during the (turn-down) event, because when the customer responds to a price signal via a call from its aggregator, the customer is essentially selling unused energy that has been purchased in advance by the supplier. This energy is sold on a DR form, and the consumer will not be charged by the supplier for energy that is not consumed. Taking all of these scenarios into account, the aggregator possesses considerable power and immunity from remunerations in the electricity market, which is frequently referred to as “free-riding” [69].

Due to the fact that the activation of a DR event can affect electricity prices, this change can affect other parties' revenue while the aggregator remains immune. This issue created a significant barrier to the integration of independent aggregators in the European market. The integration is currently limited to a few states (at the wholesale level), and as a result, there is no standard framework defining the aggregator's responsibility and relationship with suppliers and the BRP [74] [69]. However, this clause was amended later in 2017 by the European parliament, which included the wording "wholesale and retail marketplaces" in the first criterion (a) [69].

Another amendment was proposed to criterion (d). Originally, the criterion stated that no compensation for the supplier was required. However, after the modification, it now states that the supplier may be compensated in an amount equal to the amount of energy that the consumer did not use during a DR event [69]. The new criterion states “(d) transparent rules and procedures to ensure that market participants are remunerated for the energy they feed into the system during the DR

period. Were the conditions of remuneration are not agreed by market participants, they shall be subject to approval by the national regulatory authorities and monitored by the Agency".

On the other hand, The Regulatory Assistance Project (RAP) provided a comprehensive study outlining why the supplier should not be compensated [69]. The report was based on several reports from the French market, the German and Austrian markets, and the Nordic market. All of these markets operate on a day-ahead basis. Using data from these markets in 2013/14, 2014/15, and 2015/16, and taking into account various market characteristics, the report discusses the fact that the total benefits of DR programs will outweigh the total costs. The report also presents the enormous potential of DR in reducing wholesale market prices, and the societal nature of DR benefits [75].

There are two reasons why suppliers shouldn't receive compensation according to RAP [75]. The first reason is that, in order to design an incentive for DR that benefits the electricity market by reducing retail tariffs and reducing the cost of integrating intermittent resources, deployments of DR programs should be increased rather than decreased, through the development of laws that restrict demand side participation. According to RAP [75], The second argument is that a different method of compensation can be established that ensures the provider does not suffer financial loss. The French electricity market's current compensation mechanism is known as an administered arrangement is considered a reasonable choice. There are three distinct types of compensation in the French electricity market: the first is for regulated price contracts, the second is for a corrected model that is only applied to larger sites, and the third is for bilateral agreements between the aggregator and the supplier/BRP [76].

In conclusion, the primary obstacle to the successful integration of DRA into the market is a regulatory issue that can be resolved by adopting a model that incentivizes the DRA and participants to enrol in and participate in DR programs. As well as by limiting the laws that require the DRA to compensate suppliers and the entire market, given that DRA can generate enormous financial benefits for suppliers and the entire market [75]. The French model looks more promising. The main infrastructural barrier is the high cost of deploying advanced metering infrastructure (AMI) and the absence of a standardized communication tool protocol [77], which can act as a two-way communication channel between the customer and the service provider. Consumer acceptance of dynamic pricing can act as another barrier. Although enrolment in DR aggregation programs has increased in recent years, with the US alone estimating that there are more than 9 million customers in 2018 [52]. Another impediment is a lack of big analytical data and models to assess the feasibility of programs, in other words, a lack of business models.

2.1.5 Reinforcement Learning:

Reinforcement learning (RL) is a major branch in ML. It is concerned with optimizing the process of learning through experiences through trial and error [78]. RL makes use of environment data that has been gathered through observations. It begins with no knowledge of a task and learns through trial and error, keeping note of successful decisions. In disciplines other than artificial intelligence, RL is referred to as dynamic programming. It is based on behavioural psychology and incorporates numerous formulas and equations from economics [78]. According to [78], the main concepts of RL are:

- **Agent**: Also known as the controller, an agent is a type of entity whose primary function is decision-making. There are two types of agents: model-free agents and model-based agents. Both types will be discussed further.
- **Environment**: It is the environment in which the agent functions and operates, so it is essentially the "world". There are two types of environments: stochastic and deterministic. In a deterministic environment, we always end up in the same state after performing the same action. In a stochastic environment (also known as a probabilistic environment), we can end up in a different state each time we choose an action, and the same values can have a different outcome because they are inheriting a degree of randomness and a probability of an event [78]. In our situation, the environment that we're utilizing to optimize building energy and load management is called "CityLearn".
- **State (s)**: It corresponds to the location of the agent in the environment.
- **Action (a)**: This corresponds to the next move the agent is going to take.
- **Reward (r)**: A feedback the agent receives from the environment after performing an action. Keeping in mind that, the ultimate goal of the agent is to maximize the sum of “discounted “received rewards [78] [79], and the discount concept will be elaborated later.
- **Policy ($\pi(a|s)$)**: It outlines the agent's strategies for selecting actions and how the agent will behave in a particular state. A policy can be simple and straightforward or complex. In a simple policy, the effect of an action on the system is well known, which simplifies forecasting future states. Thus, maximizing the reward is simple. However, this rarely occurs. Most of the time, system dynamics are unknown, which makes forecasting the future state impossible. Take flipping a coin as an example, the system is deterministic "head or tail," but predicting the future state is impossible [79].
- **Value (v)**: It is the correlation between states and returns. It refers to the future rewards that an agent will obtain as a result of executing a future action. It is the difference between the value and the immediate reward [79].

- **Alpha**: The learning rate describes the amount of new information required to replace the older ones. Thus, the learning rate represents the speed of learning. The learning rate is a hyperparameter between ∈ (0,1) [80].

2.1.5.1 Markov Decision Process:

RL is modelled as a Markov Decision Process [78]. The environment is regarded as a partially observable Markov decision process (POMDP) [81]. It is defined as a model consisting of an array of n-tuples including states, actions, transition probabilities, an observation function, a reward function, and a learning rate: $\{S,A,O^1,T,\Omega,R,\gamma\}$ [82]. This indicates that we just need to be aware of the current condition of the environment and not of any previous states [78] [79]. MDP solves a system with a set of bounded (finite) states, actions and rewards. There can be a very large set of states, actions, and rewards but they are still finite. As mentioned previously, in stochastic (random) environments, and from the perspective of a controller (agent), the states and rewards received by the agent are random variables, which means they are associated with some degree of probability distribution [79], this distribution is defined in equation (1).

$$\sum_{s' \in S} \sum_{r \in R} p(s', r \mid s, a) = 1 \tag{1}$$

Which can be interpreted as: the probability of being in state prime (s') and receiving the reward (r) while our agent is in state (s) taking action (a), with considering the sum of all possible combination of states and the corresponding rewards is equal to 1 [79]. The expected value is equal to the outcome multiplied by the probability of the event, which can be written as shown in equation (2).

$$r(s,a) = \sum_{r \in R} r \sum_{s' \in S} p(s', r \mid s, a) \tag{2}$$

As previously stated, the ultimate goal of an agent in RL is to maximize the overall rewards obtained during an episode [79] [78]. This implies that we must specify the return value in our calculations. The return value (G_t) denotes the total rewards obtained by agents during an episode [79]. The return value (G_t) is defined as illustrated in equation (3).

$$G_t = r_{t+1} + r_{t+2} + \cdots + r_T \tag{3}$$

[1] *O* represents a set of sets of states, action, and observations

As demonstrated in formula (3), the return value increases with each time step, but this fact raises an interesting question: what if the episode is never-ending? would the return value continue to increase indefinitely? This concern is addressed by introducing a hyper-parameter known as gamma (γ), which has a value between 0 and 1 ($0 \leq \gamma \leq 1$), It refers to the notion of discounting future rewards. By employing gamma, the agent can prioritize immediate rewards above future ones, hence preventing the return value from increasing indefinitely [79]. The return values are usually called (Q) function. The modified formula is displayed in formula (4) after the discount component is introduced.

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \tag{4}$$

The agent must link and associate the received rewards with its states, as the returned value alone provides no information to the agent [79]. Linking those rewards to the states enables the agent to recognize and learn the most valuable states, allowing the agent to return to them in the future. This coloration of the states and returned rewards is referred to as the value function, which is also defined mathematically as the expected value of returns while the agent is in state (s) while assuming it is following a policy (π) [79], as shown in equation (5).

$$v_\pi(s) = E_\pi[G_t \mid S_t = s] \tag{5}$$

Another critical value is the action value function ($q_\pi(s)$), which is defined as a correlation between returns and state and action pairs when the agent is assumed to be following a policy π [79], as illustrated in equation (6).

$$q_\pi(s) = E_\pi[G_t \mid S_t = s, A_t = a] \tag{6}$$

All of the information above assumes we are dealing with a discrete environment rather than a continuous one. In reality, almost all environments are continuous, which is why we use deep neural networks as policy function and value function approximators. In other words, deep neural networks can make our environment behave like a continuous world [79].

2.1.5.2 Model-free agent & Model-based agent:

Solving MDP requires knowledge of the system dynamics, which we define as transition probabilities [80]. When the system dynamics are known, the problem is reduced to a planning problem rather than learning problem [80]. The planning problems can be either solved with value iteration or policy iteration [80] [83]. On the other hand, learning problems are solved through interaction between the agent and the environment. This interaction results in the agent learning the

system dynamics. There are two types of solutions presented for these types of problems: model-based approaches and model-free approaches. In model-based approaches, the agent first learns the model and then begins planning to solve the problem. In model-free techniques, the agent learns to correlate the highest-value actions (optimal actions) with its state without requiring the transition probabilities between the states to be linked (since it is already unknown) [80] [78].

Q learning can be one of the most famous approaches for a model-free algorithms, model-free algorithms can be divided into policy optimization approaches and off-policy approaches. Off-policy algorithms (like Q learning) is learning the Q function by using stochastic gradient descent algorithm [80] [78]. As mentioned above, the Q function basically represents the expected returns. Q learning algorithm estimates these values by interacting with the environment. After collecting transition tuples containing states, action and rewards, it calculates its gradient. The Q learning method is considered much better than a policy-based algorithms in this point, but still Q learning is less efficient than model-based approaches. The Q-values are updated as shown in formula (7).

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left[r(s,a) + \gamma \max_{a} Q(s',a) - Q(s,a)\right] \tag{7}$$

Equation (7) is referred to as the Bellman equation, and it has numerous forms [78], but this is the version used in the Q learning algorithm. The Bellman equation is regarded as the fundamental formula for RL, and its solution is our ultimate objective [79]. In Q learning it is used to update the Q table, which is considered as the agent's brain [78].

While model-based approaches are considered to be more efficient than model-free approaches, they come at the cost of obtaining a suboptimal value when the system dynamics cannot be known (learned) due to the system's complexity. As a result, model-free approaches are typically less efficient, but can perform significantly better in those situations [84]. If we are working with a problem that has a limited number of states, we can define the transition using what is called as the Q table, this table contains state-action pairs which represents the entry for the table, the values inside the table are known as the Q values, they represent the cumulative sum of the discounted rewards after following an epsilon greedy policy [78].

The fact that the Q learning is an off-policy algorithm means that it updates its values using another policy, other than the one it used to collect those values, that means it uses two different policies. One for performing actions, it can be an epsilon greedy policy, or simple greedy ...etc., and another policy for updating the Q values. On-policy algorithms uses the same policy for taking action to update Q values, the most famous on-policy algorithm is SARSA (state action reward state action). Sometimes the difference between SARSA and Q learning can be very small, in fact sometimes Q learning is called SARSA-max [78]. This difference is dependent on the chosen policy. For example,

if the agent's policy was a greedy policy (chooses the highest valued action from the next state), both Q learning and SARSA will produce the same result [78].

On the other hand, model-based learning refers to learning in which the agent is aware of the environment's dynamics and how it changes in response to received feedback, with the agent's ultimate goal being to find the optimal policy that maximizes cumulative rewards. With the introduction of deep learning, the ability of model-based learning has increased [85]. The main distinction between model-based and model-free RL is that in model-based RL, there are fewer trial-and-error experiments to determine the best action that produces the highest rewards, as there are in model-free RL. Not to mention that model-free RL has a lower data efficiency due to the large number of trial-and-error experiments. On the other hand, model-based RL can increase data efficiency by obtaining near-optimal values after learning the model. The agents can use the learnt model for generalization and reasoning [85]. The figure below Figure (2.4), depicts the distinction between model-free and model-based learning processes, and it is taken from [86].

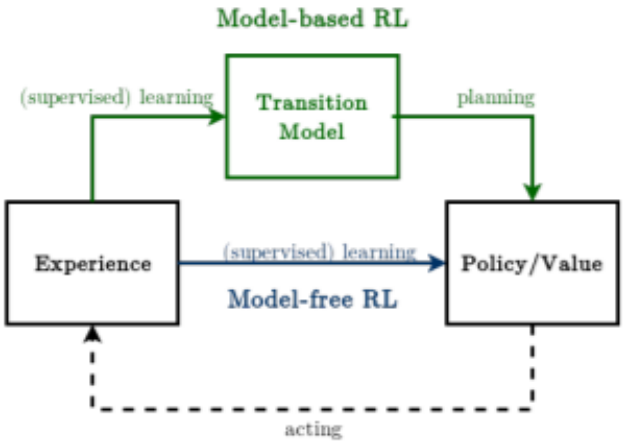

Figure 2.3 – Model-Based vs Model-free

2.2 Related studies:

This section will present various research papers on DR in residential, commercial, industrial, and market operations, as well as the energy community's fields. But the emphasis will be on using DR in residential communities and market operations, as it is the most relevant for this research. Additionally, this section will present various optimization methods, depending on the case and objective function of the problems, as well as various types of optimization approaches used for a better allocation of DR resources, such as: conventional mathematical optimization (MILP, MINLP ...etc.), and ML-based optimization approaches.

2.2.1 DR in market operations

DR can have a significant impact on various market operations and can be used to assess market performance and enhance system security [50]. The literature contains numerous contributions from various authors who used various modelling approaches and investigations to help raise awareness about the DR potential in the electricity market. Parvania, Fotuhi-Firuzabad, Shahidehpour (2013), developed a framework for DR in day-ahead market, the optimization process used Mixed Integer Linear Programming (MILP), and the objective function was to maximize the profits for the aggregator in the day-ahead market [87]. Mhanna, Verbič, Chapman (2016), presented a two-stage mechanism for demand realization and demand scheduling for the day ahead operations, to minimize the payment by the household agents. Their work covers DR for both residential and market operations [88].

Wouter L. *et al.* (2020), developed a multi objective procedure by using CES system. The first objective function aims to minimize the cost of the electricity, and the second one used to minimize the CO2 emissions and used both of them to find the trade-off between cost and emissions. A Pareto frontier of costs and emissions method is used in this framework [89]. Al-Awami, Amleh, Muqbel (2017), Introduced a framework that optimizes the bidding strategies and maximizes the VPP's profit on day-ahead and real-time bases. A fuzzy optimization approach is used, in order to maximize the VPP profit by using DR [90]. Nguyen and Le (2015), Developed a model for the microgrid aggregator, in order to determine the optimal hourly bids that the aggregator will submit in the day-ahead market. A Stochastic programming method is used to maximize the expected profit [91]. Longbo, Jean and Kannan (2012), created a DR with Energy Storage Management system, the system is used for general power consuming entities with finite energy storage and renewable energy. The objective is to minimize the energy costs by using Lyapunov Optimization [92].

Nguyen, Negnevitsky and De Groot (2012), developed a market clearing scheme, where the DR is treated as a commodity. The framework is called demand response exchange (DRX), the objective function aims to maximize profit, in order to achieve that, Nguyen *at el.* used a specific type of Hill climbing method known as tâtonnement & non-linear programming [93]. Gonzalez Vaya and Goran (2015) presented a model for the aggregator to bid in the day-ahead market while satisfying the plug-in electrical vehicle charging cost. The optimization procedure used in this work is MILP, and the objective function is divided into an upper bound and a lower bound. The upper bound aims to minimize the charging cost by the aggregator. While the lower bound aims to optimize the demand bidding process [94]. Henriquez, Wenzel, Olivares, Negrete-Pincetic (2018), presented an optimization model for an aggregator that manages a portfolio of different DR programs to participate in the wholesale market. A MILP method is used in order to maximize the aggregator profits [95].

Taşcikaraoğlu, Paterakis, Erdinç, Catalão (2019), designed a model of a direct load curtailment (DLC) of a HVAC system and energy storage, all of them are connected to the distribution system. A MILP method is used for Minimization of the energy demand of the customer & maximization of customer comfort [96]. Liu, Wu, Wen and Ostergaard (2014), developed a distribution congestion price based (DCP) market mechanism, to mitigate possible system congestions. A MILP for the minimization procedure [97]. García-Bertrand (2013) made an analysis for the different benefits of using DR programs from the retailer perspective. They used Mixed Integer Non-Linear Programming (MINLP) to maximize the retailer profit and minimize different market risks [98]. Zheng, Cai (2014), implemented a DR model that controls HVAC system to reduce the variation of non-renewable sources, and guarantee the customer comfortability. The model used a Lyapunov Optimization procedure in order to minimize the variation of non-renewable power demand and maximize the comfort of the customer [99].

2.2.2 DR in Residential communities

Numerous works have been developed in residential communities to improve load usage and reduce customer expenses. Yoshiki, Yutaro, Fumiya, Ryosuke, Koji (2014), presented a method to find a total optimized allocation of limited electric power during peak time. The model is using Generic algorithm &Sigmoid function. The objective function aims to minimize the comfort degradation of the designed DR program [100]. More recently, Gong, Jones, Alden, Andrew G (2020), presented a thermal model of a reference house and then calculated the optimal HVAC with satisfying humans comfort standard. The work is analysis based and the objective function was to minimize the peak demand and Max ramping for the distribution level [101].

Wang, Li, Ping Wang and Niyato (2013), presented a model predictive control (MPC) algorithm for EV. The work aims to schedule the charging and regulation processes and used a Quadratic programming optimization [102]. Chen, Wu, Fu (2012), developed a model for evaluation of residential price-based Demand response that can be embedded into the smart meters. They used a Stochastic & robust optimization to minimize the electricity bill for the customers [103]. Samadi, Mohsenian-Rad, Wong and Schober (2014), presented a model for pricing to reduce the uncertainty of the energy providers. They simulated the operation of customer responsiveness. The objective function aims to minimize the peak to average ratio (PAR) of the aggregated load demand. In Samadi *et al.* the model is divided into two systems. The first one is using Stochastic approximation for solving the pricing system. The second one uses dynamic programming for load control [104]. Good, Karangelos, Navarro-Espinosa, Mancarella (2015), presented a model for consuming energy and generating it using DR, based on thermal energy storage. A Stochastic optimization approach is used in order to minimize the overall electricity consumption, and the gas costs, and maximize the thermal comfort [105].

Dagdougui, Ouammi, Dessaint (2019), made an analysis and evaluation for a DR system, with onsite generation and stored energy in a building. A MILP method was used for Minimizing the peak load of the building [106]. Safdarian, Fotuhi-Firuzabad and Lehtonen (2014), presented a model for DR management, the model is used in coordination of residential customer's response to flatten the load profile. The work uses a Bi-Level optimization, and to solve it using iterative distribution algorithm, the model is casted into a single optimization problem. The objective function aims to minimize the energy expenses the deviation of the load profiles [107]. Shi, Li, Xie, Chu and Gadh (2014), made a Formulation for residential DR, they designed a distribution model for optimal demand response scheduling using a non-convex optimization method. It aims to maximize the aggregator utilities and minimize the losses. Their work is solving the problem as an optimal power flow problem (OPF), which can be solved by a convex optimization, but it is complicated to do so, that's why it got relaxed as non-convex problem [108].

Conejo, Morales and Baringo (2010), developed an algorithm that can be integrated into EMS system of a house or a small business, to reduce the energy cost and increase the utility with respect to the consumer minimum daily energy consumption level and load levels and ramping. The work is using Linear programming & Robust optimization [109]. Patnam, Kiran and Naran (2021), presented a control system of a building, that is integrated into a microgrid, and used a Particle swarm optimization (PSO) procedure, to minimize the capital cost and maintenance cost for the battery [110]. T. P. Imthias, S. Danish, Essam, Malik (2011), designed a model for a price-based demand response for optimal scheduling of loads. A Simulated Annealing algorithm is used, to minimize the electricity bill for the customers and minimize the maximum demand of the system [111].

Sibo, Ming, Gengyin (2018), developed a DR scheduling model for residential community. The optimization method used is MILP, and the objective function is to minimize the user's electricity consumption cost [112]. Terlouw, AlSkaif, Bauer, van Sark (2018), created a model for community energy battery system. The model is using MILP as an optimization method, the multi-objective function aims is to minimize the annual operating costs from grid electricity absorption, and to minimize the grid CO2-emissions [113]. Rahmani-Andebili, Abdollahi and Moghaddam (2011), designed a model for emergency demand response for Unit Commitment of thermal units. The EDRP problem is non-linear and non-convex and was solved using Simulated Annealing optimization method, and the objective function is to minimize the system production costs while satisfying load demand [114].

Qian, Zhang, Angela, Huang and Wu (2013) presented a real-time pricing model for peak to average reduction using demand response. Simulated-Annealing-based Price Control (SAPC) algorithm is the optimization method, and the objective function is to minimize the peak-to-average load ratio [115]. O'Neill, Levorato, Goldsmith, Mitra (2020), presented a residential DR Using

Reinforcement Learning. The authors developed an algorithm called CAES, and it reduces residential energy costs and smooths energy usage. The used method is Q learning, and the objective function is to minimize the user's electricity consumption costs [116]. Somer *et al.* (2017), presented a model that uses RL. The objective is to maximize the self-consumption PV production, by defining optimal scheduling of domestic hot water (DHW) heating cycles.

2.2.3 DR in Energy communities:

As a response to the climate issue, to reduce energy poverty in some areas, and to keep energy capitals local, a new trend known as energy communities has emerged recently. According to the European commission [117], energy communities can be defined as a citizen-driven actions to push for more energy green transition. DR plays a significant role in these communities because it acts as an energy management tool, a flexibility provider during peak demand, and can help reduce system imbalances. Recently, a new model known as community choice aggregation (CCA) emerged from these communities. It is defined as a model that enables local entities and governmental bodies to provide electricity services on behalf of customers, rather than power utilities [118]. O'Shaughnessy *et al*. (2019), presented an analysis study to evaluate the effectiveness of integration of CCAs into electricity market in the USA. The study uses a variety of data to explore the rise of community energies and community choice aggregation in the USA [118]. Michaud (2018) investigated the deployment of solar photovoltaic through community choice aggregation programs [119].

Kennedy and Rosen (2020) presented an investigation to evaluate the potential of CCA in California's market, and the integration of renewable resources in CCA. The investigation concluded with a recommendation of regulatory reform of the landscape that describes the relationship between the investor-owned utilities (IOU) and CCA, to procure a cooperative relationship, rather than a competitive one [120]. Huitema, Van Der Veen, Georgiadou, Vavallo, García (2020), made an investigation about demand-side flexibility in residential communities, and energy communities, and analysed the operation of Holistic Demand Response Optimization Framework (HOLISDER)[2] tool for DR. The work concluded with a great potential for HOLISDER products for residential and energy communities, but not for commercial [121]. Gjorgievski, Cundeva, Georghiou (2021), presented a study about the design and the social arrangement of energy communities. The study used several indicators in order to investigate the economic, environmental and technical impact of energy communities. According to the study these indicators are: Self-consumption rate (SCR), Self-sufficiency rate (SSR), Loss-of-load probability (LOLP), Load match index (LM), Electricity

[2] Is defined as a framework for buildings and residential sector, that can help in reducing energy costs in the consumer side.

exports, Primary energy. The study takes into account different types of energy communities, such as: shared solar PV, shared storage, multi- and hybrid energy systems, district heating and cooling systems [122].

2.2.4 DR in Industrial and Commercial sector:

Although DR is extensively researched in the literature for industrial and commercial (I&C) applications, this research work focuses exclusively on DR applications for residential communities, and hence only a few brief research studies will be presented. Abdulaal, Moghaddass and Asfour (2017), introduced a two-stage approach for autonomous demand response used for large industrial consumers. The first stage uses Quadratic programming for optimization, and the second stage uses a modified form of genetic algorithm. The objective function for the first stage is to maximize the customer comfort, and the second stage aims to minimize the deviation of the lumped load [123].

Kerdphol, Qudaih, Mitani (2016), developed a model for optimal sizing of battery energy storage system (BESS), and used dynamic pricing DR as a tool to improve the system reliability. They used Particle Swarm Optimization (PSO) as a method for optimization. The objective function is to minimize the capital cost and maintenance cost for the optimal size of the battery. The work used DIgSILENT PowerFactory software, and it is a software used for industrial and commercial analysis [124]. Wen, O'Neill, Maei (2014), produced a model for EMS for residential and commercial buildings. The proposed model used Q learning algorithm, and the objective function aims to optimize the load scheduling process [120]. Huang et al. (2019), presented a scheme for industrial DR. The objective function is to minimize the energy costs, and to determine the optimal energy management scheduling policy. They used actor critic deep neural network [121].

3 Methodology

3.1 Overview

A 2016 study [125] discussed the issue of reproducibility. The study concluded that more than 70% of researchers failed to replicate the experiments of other researchers. And more than half of them failed to replicate their own experiments. These figures were derived from an online questionnaire completed by more than 1,576 researchers. One-third of respondents stated that their laboratories are attempting to resolve the reproducibility issue. One remedy offered is to standardize experimental methodologies. OpenAI gym is a Python library that was intended to address reproducibility and standardization issues, as well as to serve as a benchmark in the field of artificial intelligence. Gym provides a vast array of environments in which to work and test agent based RL algorithms.

CityLearn is a Python framework for implementing single or multiple agent RL algorithms. The agents are used for energy management, load shaping, and DR. According to the creators [2], the main objective of CityLearn is to create a standard OpenAI Gym framework. CityLearn is described as self-contained and independent of other energy simulation tools, such as EnergyPlus. It only requires a few Python libraries (mostly Pandas, JSON, Pytorch, NumPy, Gym). CityLearn uses hourly data from pre-simulated buildings (i.e., using EnergyPlus, Modelica, or real-world data). Buildings' indoor temperatures cannot be altered, as other comfort conditions. The controller actions are then supposed to result simply on energy consumption changes.

In general, there are two forms of energy storage: passive energy storage, which refers to the thermal mass of the building. The other type is active energy storage, which includes water heaters, cooling thermal storage, batteries, and schedules for electric vehicle charging [2]. As CityLearn is programmed to meet the building's cooling and heating demands regardless of the RL agent's actions. The agent takes any action that violates those constraints, the internal control system of CityLearn will override it. According to [2], this ensures the agent may concentrate on shaping the electricity curve without interfering with the residents' level of comfort. The architecture of CityLearn is composed of three major components: attributes, methods, and subclasses. The attributes are classified as follows:

- **input*:*** such as building ids, since we have 9 different building, and data path (different data path such as weather data, and PV loads data, ...etc.).
- **Internal:** represent internal component and structure.
- **Metric*:*** it represents the cost function that we are trying to minimize, such as: the ramping, and the (1-LF) and the avg. daily peak, net demand, etc.

- **RL***:* this one represents RL's different attributes such as states, actions, rewards.

The methods describe multiple techniques that are utilized within the environment for various reasons. Methods may be classified into two categories: those that are built-in to the Gym library and those that were created by the developers:

- **OpenAI Gym:** Such as; step (), _get_ob(), terminal(),seed()..etc.
- **Others:** Next_hour(), cost()

The third component is sub-classes, which indicates classes were created by the author [2], to assist in the creation of instances that handle environmental problems. There are five distinct types of energy models and sub-classes: Electrical heaters, heat pumps, energy storage, and batteries. Appendix B contains more illustrations of the CityLearn environment's main architecture. In CityLearn, there are two forms of agent control. The first is decentralized control, which is the default mode of control. Decentralized control is characterized in the RL literature as a system composed of several agents who interact with one another and share a common environment [126]. Different agents conduct the learning task by acting on the acquired data. The main advantage of this strategy is that it does not need extensive communication between the agents [127]. The second sort of control is centralized control, in which a single agent is responsible for all of the buildings.

The default agent mode in the CityLearn environment is the decentralized agent. When an action is performed, the environment returns a list of rewards, each of which corresponds to a different building, as well as a list of lists of states. This occurs whenever an action is performed in the decentralized mode. In this mode, the action is a list of lists, each of which corresponds to a different building. The other type of agent mode is called the centralized agent. In this environment there are three type of actions the agent can be performed.

- Acting on the chilled water tank
- Acting on the Domestic Hot Water Tank (DHW)
- Acting on the Battery system

When an action is performed, the CityLearn environment returns a single reward for all buildings and a list of states. The states represent only the unique states of each building. For example, the outdoor temperature appears only once because all buildings are treated as a single agent. Other unique states, such as Domestic Hot Water (DHW), and state of charge (SOC), appear as many times as the number of the buildings, since there are different DHW systems mounted in each building.

3.2 The environment

3.2.1 Data preparation

The first step in CityLearn is to insert the data required for the procedure. There are several types of data that are used in the environment. The first input is the climate zone. This can be done manually from the agent's main file. The tool supports five different climate zones [2], with the fifth being the default choice and the one used in the case study. The climate zones in the environment correspond to the US climate zones. The International Energy Conservation Code (IECC) classifies the US into eight distinct temperature-oriented climate zones [128]. These zones are primarily classified into three types of moisture regimes: A, B, and C. However, only five of these types of moisture regimes are employed in the environment: Humid, humid-hot, humid-warm, humid-mixed, humid-cold. The following table (Table (3.1)) summarizes the various climate zones and their associated minimum and maximum temperatures.

Table 3.1 – Climate zones

Whether name	Upper limit	Lower limit	information
Hot humid	23°C	19.5°C	–
Mixed-Humid	18.3333°C	7°C	–
Hot-Dry	–	7°C	–
Mixed-Dry	18.3333°C	7°C	5400-9000 hours of cold
Cold	18.3 C	–	–
Very Cold	18.333 C	–	9000-12600 hours of cold
Subarctic	–	–	More than12600 hours of cold
Marine climate	–	–	–

Another input is the building attributes file, which contains all of the buildings' states and actions. The buildings are numbered sequentially in this file, and each building contains a dictionary containing its attributes (state, action, and rewards), which can be modified by the user. The weather file represents another input data file, and it contains weather input data for the user-defined climate

zone. We also need to define solar generation, as photovoltaic systems are used to offset some of the demand, and it contains hourly solar generation in kilowatts. The data spans 35040 hours, or four years. The carbon intensity is a file that contains the carbon intensity of electricity. It indicates how much CO2 is emitted per kilowatt hour of electricity consumed[3].

The building's ids are an input. They display the building's id number. As previously stated, there are nine distinct buildings. The first is a medium office, the second is a fast-food restaurant, third is a standalone retail shop, and the fourth is a strip mall retail, the remaining five are residential multi-family buildings. Each building has its unique load profile to avoid them behaving similarly, and each building's solar energy is utilized to balance a portion of its power use but not all of it. Following the input step, the second step creates instances of all the buildings' sub-classes. These sub-classes are: Electrical heater, Heat pump, Battery, and Energy storage. The final instance is the building itself, which is the main instance. It accepts all the other energy models as input, as each of the nine buildings has its own air-to-water heat pump and some of the buildings have an electrical heater that supplies domestic hot water (DHW). All that is required is to unpack the energy models and apply each model's data to the appropriate building. The energy models and load categories are illustrated in Table 3.2.

Table 3.2 – Environment input data

Data	Description
Non-shiftable loads	All the electrical equipment’s in the buildings excluding (HVAC) in kWh
Cooling demands	Cooling demand for each building in kWh
DHW demand	Domestic hot demands for each building in kWh
Time parameters	All the time parameters information (day, month, hour): the day parameters are numbers from 1 to 8, with 1 represents Saturday and 2 is Sunday,7 is Friday, and 8 is a Holiday, the months are from 1 to 12 and the hours are in a 24-clock format
Unmet cooling set point difference (UCSD)	The difference between the thermal zone temperature and its set point. Average UCSD is the average of the building across all the thermal zones and weighted by their floor area
Indoor temperature	The indoor average temperature in all buildings, and it is weighted using the floor area, in (°C)

[3] https://carbonintensity.org.uk/

The outdoor temperature	The outdoor average temperature in all buildings, and it is weighted using the floor area, in (°C)
Indoor relative humidity	Between (0 – 100%) across all the thermal zones.
Outdoor relative humidity	Outdoor relative humidity (Rh) between (0 – 100%) across all the thermal zones
Indoor average temperature	The average indoor temperature in (°C) and weighted by their floor area, in (°C)
Daylight savings status	A Boolean variable to state whether an energy is saved during daylight or not (0, 1)
Diffuse solar radiation	It is the solar radiation that has been reflected and then lands on the earth surface [130]
Diffuse solar radiation prediction for 6h	A 6h prediction of diffuse radiation
Diffuse solar radiation prediction for 12h	A 12h prediction of diffuse radiation
Diffuse solar radiation prediction for 24h	A 24h prediction of diffuse radiation
Direct solar radiation	It is the direct solar beam that hasn't been reflected [130]
Direct solar radiation prediction for 6h	A 6h prediction of direct radiation
Direct solar radiation prediction for 12h	A 12h prediction of direct radiation
Direct solar radiation prediction for 24h	A 24h prediction of direct radiation

3.2.2 Determining the agent's limits:

This stage will size the building's energy usage in order to calculate the upper and lower boundaries of the observation space and action space. This will help the agent in acting within those bounds, simplifying the optimization process and making any function approximators more effective. It is critical to understand that both types of control have two distinct limits: centralized agent limits and decentralized agent limits. Each has its own observation space and action space. The observation space boundaries consist of the electricity consumed by each building. Which is essentially the total and pure energy consumption (in kWh). However, before computing the total electrical energy consumption, we must calculate the energy produced by each energy supply device:

- **Air to water ratio Heat pump:** It converts electrical energy from the grid to thermal energy by moving hot air from one side to the other and cooling it with refrigerant (Freon). A heat pump can also be used as a heater. If the thermostat is set to the heating mode, the heat pump will reverse the process by kicking hot air from the outside to the inside. The data we entered in the previous step pertains to the thermal energy consumption of the heat pump, not the electrical energy consumption. To determine the total energy consumption of the heat pump, we must obtain the thermal energy consumed by the heat pump and divide it by the coefficient of performance. As illustrated in equation (8).

$$E_t^{hp} = \frac{Q_t^{hp}}{COP_t} \tag{8}$$

As indicated in the equation (8), after dividing the thermal energy (Q_t^{hp}) by the coefficient of performance (COP_t), which represents a performance metric that is mostly utilized in marketing and technical applications [131]. It is defined as the useful amount of heat delivered per unit of electricity input. We obtain the electrical energy consumption of the heat pump (E_t). This energy represents the DHW electricity demand for each building, as well as cooling demand.

- **Electrical heaters**: It convert electrical energy to thermal energy via electrical resistance heating. It is 100 percent efficient because all electrical energy is converted to thermal energy with no losses. However, when compared to a heat pump, heat pump efficiency is significantly higher, reaching up to 300 percent in the case of the heat pump used in CityLearn. The electrical energy consumed by an electrical heater (E_t^{heater}) is calculated by dividing the thermal energy consumed by the heater (Q_t^{heater}) by its efficiency, as indicated in equation (9).

$$E_t^{\text{heater}} = \frac{Q_t^{\text{heater}}}{\eta_{eh}} \tag{9}$$

- **The solar panel's produced energy:** It can be estimated by multiplying the PV generation (P^{solar}) by the AC inverter's hourly data ($E^{\text{inverter output}}$). Because the data reflect the solar PV generation (P^{solar}), the equivalent energy is expressed in (kWh), as indicated in equation (10).

$$E_t^{\text{solar}} = P^{\text{solar}} \times E^{\text{inverter output}} \tag{10}$$

The total network consumption shown in equation (11) represents the total consumption of the building. It represents the upper bounds for both centralized and decentralized agents. In both cases, the lower bound is equal to zero. In the case of a decentralized agent, the observation space

contains two categories: the total energy consumption shown in equation (11), and the other building's attributes. The lower and upper limits for building attributes such as indoor temperature and relative humidity, average unmet set point, solar generation, etc. are their minimum and maximum values in the data cheat. Excluding the energy storage state of charge (SOC), which will be calculated later. However, in the case of the centralized agent, the observation space is divided into three categories: inner characteristics of the building (e.g., the indoor temperature, the solar generation, the indoor relative humidity, the average unmet setpoint). The second is the total energy network consumption, and the third is the state of charge (SOC) of energy storage (tanks). The reason for this is that because all buildings are controlled by a single agent. There is no point in using the outdoor temperature for each building, as it is assumed to be constant across all buildings, which is not the case for the decentralized agent.

$$E_{total} = E_{appliances} - E_{PV} + \left(\frac{E_t^{heater}}{\eta_{eh}}\right) + \left(\frac{E_t^{heat\ pump}}{COP}\right) + C_{cooling} + C_{DHW} \quad (11)$$

In general, the agent can do three sorts of actions: Acting on cooling storage, acting on DHW storage, and acting on electrical storage (Battery). The action space is determined by the capacity of those storages (Tanks & battery). Capacity is a measure of the energy storage's size in comparison to the building's maximum hourly energy demand. The purpose is to establish a limit that the agent cannot exceed. This limit is set to be between $\left(\frac{-1}{\text{Capacity}}\right)$ and $\left(\frac{1}{\text{Capacity}}\right)$. Thus, the energy storage system cannot deliver more energy than the building will ever use. The upper and lower action bounds are illustrated in equations (12) and (13) correspondingly.

$$a_{low} = max\left\{\left(\frac{-1}{Capacity}\right), -1\right\} \quad (12)$$

$$a_{high} = min\left\{\left(\frac{1}{Capacity}\right), 1\right\} \quad (13)$$

According to [2], this should accelerate the learning process and make it more stable and effective than just establishing the limits between (-1 and 1), e.g., if the chilled water tank capacity is three times the annual maximum hourly demand, the agent action will be bounded between $\left(\frac{-1}{3}\right)$ and $\left(\frac{1}{3}\right)$. The observation space of the decentralized agent contains twenty-eight values for each limit (upper bound and lower bound). Each building has its own observation space and action space, but this is not the case for the centralized agent, whose observation space contains ninety-one states for all buildings. There are thirty states available for each building, but some are excluded by default, namely: the day light saving status which is set to False (excluded), and the average unmet set point is also set to False (excluded), resulting then in twenty-eight possible states for each building.

The centralized agent's action space contains twenty-five actions. Each building has three available actions, and there are nine buildings. Thus, the action space should equal twenty-seven, but the third and fourth buildings lack DHW storage (Tanks). On the other hand, the decentralized agent's action space contains three distinct actions for each building. Except for the third and fourth buildings, which have only two actions for the same reason. The following step is to calculate the COP for both the heat pump and the electrical heater. The COP of the heat pump is dependent on several factors, including the target temperature, technical efficiency, and outdoor air temperature. If the heat pump is used to supply heating demands, the COP will be calculated as shown in equation (14).

$$COP_c = \eta_{\text{tech}} \cdot \frac{T^c_{\text{target}}}{T_{\text{outdoor air}} - T^c_{\text{target}}} \tag{14}$$

Additionally, if a heat pump is utilized to supply the cooling demands, the COP may be determined using the formula stated in equation (15).

$$COP_h = \eta_{\text{tech}} \cdot \frac{T^h_{\text{target}}}{T^h_{\text{target}} - T_{\text{outdoor air}}} \tag{15}$$

Typically, when a heat pump is used for cooling, the target temperature (T^h_{target}) is between 7 and 10 degrees Celsius. But when it is used for heating, the target temperature (T^h_{target}) is 50 degrees Celsius. (T^h_{target}) represent the logarithmic mean of the water temperature on the storage (Tank) and the water temperature returning to the heat pump [2]. The thermal energy generated by the heat pump may be estimated using the state of charge and capacity, as well as the thermal demand, as illustrated in equation (16).

$$Q^{hp}_{t+1} = C \times (SOC_{t+1} - SOC_t) + Q^{dem} \tag{16}$$

(Q^{hp}_{t+1}) represents the quantity of thermal energy that the heat pump will deliver. (C) represents the DHW tank's capacity. (SOC_{t+1}) represents the state of charge (next time step). (SoC_t) represents the current state of charge, and (Q^{dem}) represents the thermal demand. In general, two forms of energy storage exist: domestic hot water (DHW) and chilled water tanks, which serve as energy supply devices (either a heat pump or an electrical heater). It provides the energy storage with a thermal heating or cooling energy. The output of the energy storage (Q^{sup}_{out}) represent the thermal energy demand of the building (Q^{hp}_t). The output of the energy supply device (Q^{sup}_{out}) represents the thermal energy of the storage system (Q^{heater}_t). This process is calculated in equation (17).

$$Q_{out}^{sup} := Q_t^{hp}; \quad Q_{out}^{\text{sup}} := Q_t^{\text{heater}} \tag{17}$$

The (SoC_t) can be calculated as shown in equation (18).

$$SoC_{t+1} = SoC_t \times (1 - e_{loss}) + Q_{in}^{sto} - Q_{out}^{sto} \tag{18}$$

In equation (18), (e_{loss}) is referred to as the thermal loss coefficient, which represents the stored thermal energy that is lost each hour. ($Q_{\text{in}}^{\text{sto}}$) represents the thermal energy inside the storage, and ($Q_{\text{out}}^{\text{sto}}$) represents the thermal energy leaving the storage. To calculate the thermal energy leaving the storage, the thermal energy entering the storage is divided by the root of the efficiency, as shown in equation (19). To calculate the thermal energy leaving the supply device, the thermal energy entering the storage will be divided by the root of the efficiency as shown in equation (19).

$$Q_{\text{out}}^{\text{sto}} = \frac{Q_{\text{in}}^{\text{sto}}}{\sqrt{\eta_{\text{eff}}}} \; Q_{\text{out}}^{\text{sup}} = \frac{\text{Q}_{\text{in}}^{\text{sto}}}{\sqrt{\eta_{\text{eff}}}} \tag{19}$$

The thermal energy leaving the storage can also be calculated by dividing the building's thermal demand (which equals the thermal energy in the storage) by the efficiency's root, as shown in equation (20). The entire system, including the storage systems and thermal energy supply devices, is illustrated in figure (3.1).

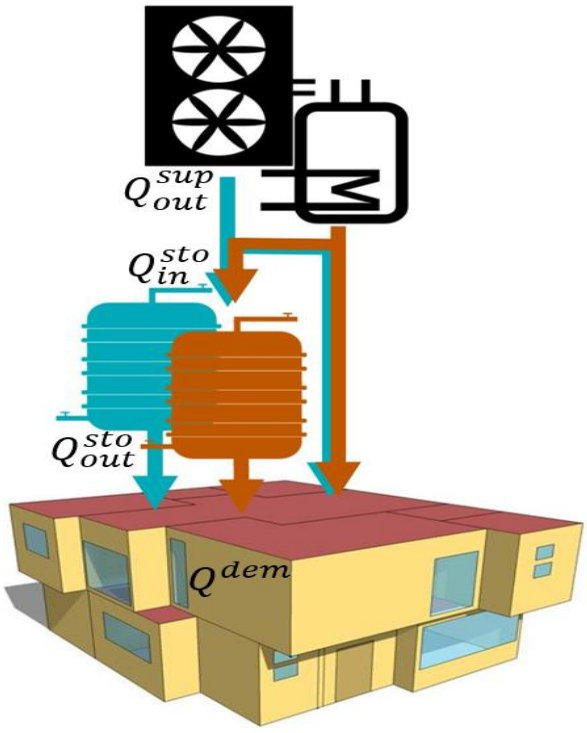

Figure 3.1 – System Component

$$Q_{\text{out}}^{\text{sto}} = -\frac{Q^{\text{dem}}}{\sqrt{\eta_{\text{eff}}}} \quad Q_{\text{out}}^{\text{sup}} = \frac{Q_{\text{in}}^{\text{sto}}}{\sqrt{\eta_{\text{eff}}}} \tag{20}$$

The nominal power for the supply devices is then calculated. With the purpose of ensuring that the thermal energy supply device is always capable of meeting the maximum DHW/cooling demands. This is accomplished by dividing the maximum thermal energy demand by the COP in the case of the heat pump, or by the efficiency in the case of the electrical heater, as shown in equations (21) and (22).

$$E_{\text{nominal}} = max\left\{\frac{Q_t^{hp}}{COP_t}\right\} \tag{21}$$

$$E_{\text{nominal}} = max\left\{\frac{Q_t^{\text{heater}}}{\eta_{eh}}\right\} \tag{22}$$

The capacity of both the heat pump and the electrical heater was then computed. As previously stated, capacity is a measure of how large the energy storage is in comparison to the building's maximum hourly energy consumption. According to this definition, capacity equals:

$$DHW_{\ storage} = max\left\{\frac{Q_t^{hp}}{COP_t}\right\} \times C_{DHW} \tag{23}$$

$$Chilled\ water_{\ storage} = max\left\{\frac{Q_t^{\text{heater}}}{\eta_{eh}}\right\} \times C_{cooling} \tag{24}$$

The next step is to define the correlations between all the buildings. Because each building can have an effect on the others. This process can aid in coordination, i.e., if the controller's action is greater than the building's electricity demand, the excess electricity is fed into the micro grid and can be used by other buildings (if needed). As a result, it is necessary to develop relationships between all buildings, as the controlling agent is a multi-agent system.

The correlation referred to in this environment is called Pearson's product-moment and it is a method for determining linear correlation between two distinct sets of data [132]. It is essentially an indicator of the degree to which two variables vary together. The result is a two-by-two array, but we are only interested in a single value answer. So a single value correlation answer is used. The

answer is saved in order to be used later by the agent. Correlations will be calculated for three distinct areas: DHW demand, cooling demand, and non-shiftable loads.

On the other hand, unlike thermal energy supply devices, batteries have a capacity defined in kWh and a nominal power defined in kW. During each charge and discharge cycle, a portion of the battery capacity is lost, defined by a coefficient known as the lost capacity coefficient (c_{loss}). The new battery capacity can be expressed as shown in equation (25).

$$C_{\text{new}} = d \times (\#\_\text{of\_cycles}) \times C_0 \tag{25}$$

While (#_of_cycles) equals to:

$$(\#\_\text{of\_cycles}) = \frac{|E_{\text{in | out}}|}{2 \cdot C} \tag{26}$$

And by substituting its value:

$$C_{new} = c_{loss} \times C_0 \times \frac{|E_{\text{in | out}}|}{2 \times C} \tag{27}$$

(C_0) is defined as the original capacity of the battery in (kWh). (C) as the current capacity of the battery. ($E_{\text{in | out}}$) as the energy that has been charged or discharged. The maximum charging power of the battery is dependent on its state of charge (SOC). When the CityLearn environment is being constructed, one of the characteristics of the buildings we used was the capacity power curve. This curve represents the maximum power provided by the battery at any given time as a function of SOC. For example, if the capacity power curve is equal to [[0., 1], [0.8, 1], [1. ,0.2]], the curve can be interpreted as: if the SOC <= 0.8, the battery is going to charge at the nominal power rate. But if the SOC > 0.8, the battery going to charge at 0.2 of the nominals. The state of charge for the next time step is defined as shown in equation (28).

$$SoC_{t+1} = SoC_t \cdot (1 - e_{\text{loss}}) + E_{\text{in | out}} \tag{28}$$

Equation (28) specifies the charge state that will exist in the future (SoC_{t+1}), by taking the current state of charge (SoC_t) and the quantity of energy spent or stored into account ($E_{\text{in | out}}$), and the loss coefficient, (e_{loss}). Which represents a user-defined parameter. It is a number that may be ignored due to its low value, and it indicates a ratio of the energy lost during standby. After the agent performs an action, the battery is discharged. If the quantity of energy released exceeds the overall electrical demand of the building. The extra energy is injected into the micro grid, and then utilised

by adjacent buildings. So reducing the quantity of energy drawn from the main feeder. The entire behaviour may be described using the equation (29).

$$E_{net}^{\text{microgrid}} = \sum_{i=0}^{n} \left(E_{b_i} + E_{\text{bat}} \right) \tag{29}$$

Whereas the (E_{b_i}) represents the total net electricity consumption by the buildings. The (E_{bat}) represent the energy consumed by the battery and adding those two values results in the total energy consumed by the microgrid ($E_{net}^{\text{microgrid}}$). The energy provided by the battery can be expressed as shown in equation (30).

$$E_{bat} = \frac{E_{\text{in | out }_i}}{\sqrt{\eta_{\text{eff}}}} \quad if \quad E_{\text{in | out }_i} \geq 0 \tag{30}$$

Whereas (η_{eff}) represents round-trip efficiency, which is a value assigned by the user, there are two possible values: either a constant value, which can be specified in the building attributes section, or a function of the charging and discharging rate P, which is similar to the battery capacity and power curve but is referred to as the power efficiency curve. The data used in the CityLearn case study contains the single value.

3.3 The agent

This part presents several agents structures and concepts. In this study three distinct types of agents are employed. The first one is the Rule based Controller (RBC), which provides the baseline results which is going to be used as a reference to compare other agent performances. Second agent is called multi agent RL with iterative sequential action selection (MARLISA). It represents an algorithm that enables any non-policy algorithm to be transformed into a decentralized multi agent algorithm for coordination. The last agent is Soft Actor Critic algorithm (SAC). It represents a model-free algorithm that aims to reduce sample complexity and increase the performance of the agent.

3.3.1 Rule base Controller

The rule-based controller (RBC) serves as a benchmark for measuring performance and comparing results obtained from other agents. The RBC is manually tuned by the developer [133]. The (RBC) is a greedy algorithm whose primary goal is to reduce the building's energy consumption by storing energy during certain times. For example, energy is likely to be stored at night and released in the morning because the COP is higher at night. The RBC is divided into six different periods:

- From 7 to 11 at mornings the action is set to be equal to −0.02.
- From 12 to 15 afternoon the action is set to equal to -0.02
- From 16 to 18 evening the action is set to equal to −0.044
- From 19 to 22 at night the action is set to equal to -.024
- From 23 to 24 at midnight the action is set to equal to 0.034
- From 1 to 6 at morning the action is set to equal to 0.05532

The returned action is an array of actions selected from the action space. The size of this array is dependent on the dimensions of the action space of the buildings (typically three except for two buildings). This action is calculated by manually tuning it and then selecting the values that produced the best results. In the analysis chapter, a comparison between (RBC) and MARLISA is made, which also uses RBC for exploration and then SAC for exploitation. The MARLISA action selection mechanism is dependent on the mode we are in. Whether we are using "safe exploration" or not. "Safe exploration" is a Boolean variable whose value is determined by the users. If we are using "safe exploration," that means we will use RBC for action selection in MARLISA. If we are not, the action will equal the action space multiplied by a scaling factor of 0.5 (31). While "safe exploration" is selected by default, both modes will be compared in the result section for the decentralized agent.

The RBC has been calibrated so that it charges 9.1 percent of its maximum capacity between 10pm and 8am. It releases (discharges) approximately 8% of its maximum capacity between 9am and 9pm. The RBC's primary structure is a loop. This loop should return actions based on the time of day, as explained previously. In a decentralized agent, the actions are three per building, whereas in a centralized agent, the actions are three for all buildings. As illustrated by equation (31), which shows another action selection method is used if the exploration mode is disabled.

$$a = coef \times rand\{a_i\} \tag{31}$$

The $(coef)$ is equal to 0.5 and it is predefined by the author. $(rand\{a_i\})$ is a random sample from the action space. ($rand\{a_i\}$), represents a random action sample taken from the action space, and $(\boldsymbol{a})$ is the action.

3.3.2 Soft Actor Critic (SAC)

The use of neural networks in the field of RL for function approximation faces two distinct challenges. To begin, sample complexity, because the model-free learning process requires a large amount of data collection. Even basic activities take millions of steps. Much alone complicated behaviour and tasks with high-dimensional observations, which will require even more data. The second issue is setting the hyperparameters, which include the learning rate and exploration

constants. These parameters must be tuned extremely precisely to achieve the best outcome [134]. As a result, this has an effect on the adoption of off-policy approaches in practice. As off-policy methods typically rely on prior values (values from the past).

Adopting non-policy techniques and approximating non-linear functions with neural networks might create complications. These complications are resolved by employing a separate actor network [135]. There are three critical components to the SAC algorithm: Actor-critic architecture with distinct value and policy networks (separated). And a formulation that permits the utilization of earlier data, referred to as the replay buffer. The replay buffer is a basic data-generating mechanism in off-policy techniques. It has been demonstrated that it can improve sample efficiency by saving the most recent transitions collected [136]. To enhance the stability and exploration, maximum entropy RL is used to optimize the policies, which results in maximized of the anticipated returns and entropy of the policy. This improves the exploration and robustness of the error estimates [135].

After encoding all of the state's variables, a principal component analysis (PCA) technique is performed. PCA is often used to bring out patterns by suppressing variances, and it is also used to clean our data sets and minimize their dimensions. PCA is based on two mathematical theories[4]:

- Variance and Covariance.
- Eigen Vectors and Eigen values.

PCA is performed directly in this work using the Sklearn Python library, which eliminates the need to worry about algorithm procedures. What is required is obtaining both action and state dimensions for all the buildings. Which is accomplished by considering all the encoded data as a state dimension and the action space we obtained as an action dimension. The state dimensions are interpreted as the number of components that will be used in the PCA. As previously stated, we will receive a tuple of returns for each action. This tuple will comprise the present state (s) and the action (a) and the reward (r) and the future state.

As previously discussed, the primary objective of RL is to maximize expected returns. However, when SAC is used, the objective changes slightly because the algorithm employs a maximum entropy architecture. The new objective will generalize the standard objective by increasing it by the entropy term, as illustrated in equation (32).

$$\pi^* = arg\,\max_{\pi} \sum_{t} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[r(s_t,a_t) + \alpha\mathcal{H}\left(\pi(\cdot|\, s_t)\right)\right] \tag{32}$$

[4] https://towardsdatascience.com/pca-eigenvectors-and-eigenvalues-1f968bc6777a

Whereas the (α) indicates the temperature parameter, which is distinct from the learning rate. It demonstrates the relevance of entropy in relation to the reward, and so is utilized as a control parameter for the stochasticity of the optimum policy. The ($\mathcal{H}$) is the entropy, which relates to the predictability of the agent's activities. The entropy may be used as a proxy for the policy's certainty. If the policy is certain, the entropy will be low, and vice versa. The entropy is also used as a scale factor for incentives. The entropy value is given in equation (33).

$$\mathcal{H}\big(\pi(.\mid s_t)\big) = \mathbb{E}_{a\sim\pi(.|s)}\big[-log\big(\pi(a\mid s)\big)\big] \tag{33}$$

As can be seen, the maximum entropy technique has a different objective function than the standard one. However, when the temperature parameter is equal to zero ($\alpha \rightarrow 0$) the standard (traditional) RL objective is obtained as stated in equation (7). Additionally, a discount factor can be introduced if the formula is to be used in infinite horizon issues [134] [135]. The negative element in the entropy formula works as a barrier, by removing unsuccessful explorations for a certain state [137].

The critic network, also known as the policy evaluation network [135], has the main objective of computing the value of our policy while taking into account the maximum entropy objective. To accomplish this, a Bellman backup operator ($\mathcal{T}^{\pi}$) will be applied to our estimated (Q) values as shown in equation (34).

$$\mathcal{T}^{\pi}Q(s_t, a_t) \triangleq r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim p}[V(s_{t+1})] \tag{34}$$

Whereas the value of ($V(s_t)$) can be calculated as shown in equation (28).

$$V(s_t) = \mathbb{E}_{a_t\sim\pi}[Q(s_t, a_t) - \alpha log\ \pi(a_t\mid s_t)] \tag{35}$$

The ($V(s_t)$) represent the soft value function, and the soft (Q) is produced using the bellman backup operator repeatedly. The objective is to reduce the temporal difference error and then back-propagate it via gradient descent, as is the case with other DL algorithms. The authors of SAC published their first paper in 2018 and it included a separate value network and a separate critic network for ($Q\ (s, a)$), as well as a third actor network. However, in the second version of SAC, they used a double critic trick similar to the TD3 algorithm (another deep RL technique). The purpose of taking the minimal or "pessimistic" limit of these networks is to avoid overestimation bias. Which can result in greater probability associated with suboptimal actions and utilizing pessimistic limits can help decrease overestimation bias [135]. The formula in the second edition has been changed to be equal to equation (36).

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t,s_{t+1})\sim\mathcal{D}}\left[\left(Q_{\theta_1}(s_t,a_t) - y_t\right)^2 + \left(Q_{\theta_2}(s_t,a_t) - y_t\right)^2\right] \quad (36)$$

Whereas (y_t) equals to:

$$y_t = r + \gamma\mathbb{E}_{a\sim\pi(\cdot|s_{t+1})}\left[\min_{i=1,2} Q_{\bar{\theta}_i}(s_{t+1},a) - \alpha log\left(\pi(a \mid s_{t+1})\right)\right] \quad (37)$$

The first network initialized during the implementation process is the replay buffer network. There are two types of replay buffers or experience buffers. The first is the model replay buffer. Because off-policy methods typically use replay buffers to store transitions or state-action-rewards tuples [2]. This helps improve sample efficiency [136]. The second buffer is the regression buffer. The algorithm gathers and processes data from the CityLearn environment in order to train the RL agent and a regression model [133]. Both buffers have been designed with a capacity of 100000 state, for the first and 30000 state for the second.

Then, as mentioned previously in the second version of SAC [137], the authors modified and used two different critic networks. The networks are constructed using three fully connected or linear layers. Each of which contains three major partitions. This version is used in the SAC implementation in this work. The first is the input layer, which represents a summation of the total number of possible states for each building and the total number of actions that each building is capable of performing (equal to 3 in most cases). The output of the first layer serves as the input for the second layer, which is called the hidden layer. The hidden layer is used to process complex data. It contains nodes/neuron, and the true value of the nodes in the hidden layer is unknown. The input and output neurons of the layer are equal to (400,300), and this is a value chosen by the author by a manual tuning process.

The second network is the hidden layer network. Both input and output are taken from the dimension mentioned above. The third network takes an input from the hidden layer. After constructing the main structure of the networks, a normalization layer called "LayerNorm" was introduced. Layer normalization is a technique for normalizing and scaling the distribution of intermediate layers. It helps in boosting training time, allowing a smoother gradient, and improves generalization accuracy [138]. The final step is to initialize the neural network's weights, which is a critical step because the purpose of using neural networks and DL is to find the optimal set of weights that can produce the desired results. Because the DL algorithm is iterative in nature, incorrect initialization of the weights can result in problems such as vanishing/exploding gradients.

The gradient vanishes if the error propagates backward from the output layer to the input layer. In this case, the gradient shrinks until it equals zero and vanishes, and the weights remain unchanged. The explosion occurs when the gradient grows larger, leaving the weight updates at very large values. Typically, weights are initialized around zero but not exactly zero. In our case, the

network was initialized with a value between $(-3\ e^{-3}, 3\ e^{-3})$ using a uniform distribution. The bias will proceed in the same manner. The bias is essentially an additional input to the next layer that is unaffected by the preceding layer's output. The bias ensures that even if all of the networks' inputs are equal to zero, the network will always be activated.

The bias was initialized following a uniform distribution $(-3\ e^{-3}, 3\ e^{-3})$. The entire structure is depicted in Figure (3.2), which is derived from [139], and it is quite similar to the one utilized in the model we are employing. Every network employs a feed forward method, as information always flows forward from the input to the hidden layers and then to the output. There are no cycles. The feed forward method treats the input as state and action, and after an input is assigned to the network, an activation function is used. The activation function is critical in neural networks, as it adds non-linearity. Without the activation function, the model can be reduced and discretized into a simple linear model. The activation function utilized in this study is Rectified linear units ("ReLUs"). The "ReLU" function can be represented as stated in equation (38). It activates the positive part of its argument.

$$y = f(x) = max(0, x) \tag{38}$$

When the input is larger than zero (>0), the output of the (ReLU) function (y) will always grow. This prevents the gradient from decreasing to zero (gradient vanishing). But when the input is equal to a negative value, both the output and the gradient will equal zero.

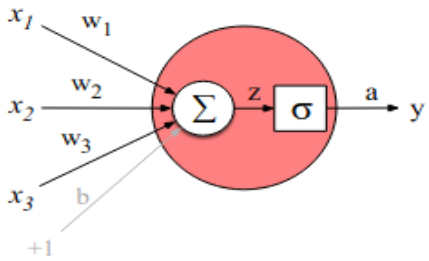

Figure 3.2 – Network activation

Figure (3.2) depicts a similar situation to the networks used in the algorithm, which consist of a number of inputs, (x_1), (x_2), and (x_3). Each with its corresponding weights, and a bias (b). When the sum of all the networks equals 1, the sum of all the networks is activated using a sigmoid function (in our case, (ReLU) to produce an output (y). Another network inside the SAC architecture is the network of action-value and value function. Which is formed identically to the critic network. The policy optimization network, or actor network, is distinguished from the critic network in that the

first layer contains the input. In this case is the total number of states, not the sum of states and actions. The output of the first layer is the first dimension of the hidden layer, which is equal to 400. The second layer contains the hidden layer, with the input and output equal to the provided dimensions of the hidden layer. This layer's input and output are equivalent to the dimensions specified for the hidden layer, which are (400,300), respectively.

The third and fourth layers are identical. One will represent the log standard deviation, which defines the width of the distribution. The other will represent the mean, which defines the distribution's centre point. Both will be required for the reparameterization trick. The weights and bias of both layers will then be initialized similarly to the critic networks. The network operates in a feed forward fashion, with each node triggered by a Rectified linear unit ("ReLU"). The next step is to apply a clamp on the standard deviation, as we do not want our policy's distribution to be arbitrarily broad. Rather, we want to confine it to a certain range, which is defined as (-20, 2). These numbers were picked based on experiment by the author.

Another method used within the actor network is the sample method. If we are dealing with a discrete environment. We will assign a probability to each discrete action, and the sum of all the probabilities should equal one. However, if we are dealing with a continuous environment, we will require some kind of distribution for our action space. And because we are dealing with a continuous space, the actor network will use a Gaussian distribution (normal). This distribution is applied to the output of the log standard deviation and mean, and then a sample from the Gaussian distribution is obtained.

At this point, the original creator of the SAC algorithm employed a technique dubbed the reparameterization trick, which will be briefly described because it falls outside the scope of this paper. The trick introduces another source of noise (ϵ). This introduces stochasticity into the distribution. Allowing it to backpropagate through a random node, assisting in the node's transformation into a deterministic node [140]. Using the PyTorch library, this is accomplished directly when selecting a sample. All that is required is to use the method (rsample()) rather than ("sample()"), to select a random sample from the distribution, as the latter represents a noise-free random sample. The random sample will represent the controller's action. This action will be triggered by a Tan hyperbolic activation function (Tanh). Tanh's output value ranges between (-1,1), which is advantageous when we require both positive and negative output values [141]. Equation (39) expresses the Tanh activation function. Finally, the log probability is used to generate the loss function, which is used to update the weights of the neural networks.

$$y = f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{39}$$

3.3.3 MARLISA

MARLISA is a coordinated DR algorithm that enables any non-policy algorithm to be transformed into a decentralized multi agent algorithm for coordination. MARLISA stands for multi agent RL with iterative sequential action selection. Its primary inspiration is the asymmetric multi-agent RL algorithm [142]. The agent is capable of forecasting future electricity usage and then sharing this knowledge with others via a leader-follower schema. To do this, it uses a combination of a baseline controller designed manually by the author and the Soft Actor Critic (SAC) algorithm.

To begin, energy coefficients for buildings are calculated, which provide a rough estimate of how much energy the building consumes over the course of a year. This process aids in the action selection step when using the SAC algorithm. The coefficients are equivalent to the total energy demand. Which includes DHW via an electrical heater or a heat pump, cooling demand via a heat pump and non-shiftable loads, and solar PV capacity generation. It is divided by the total annual hours (so it will be transformed into energy) and given that the yearly performance of a photovoltaic system is 16.66 percent of the maximum achievable performance. This estimate is regarded optimistic, since photovoltaic systems can produce less energy than this. The operation is depicted in equation (46).

The second stage is to encode all data about the environment. The developer used five different types of custom designed data encoders. Because different types of data are used in the environment. The first type is hot encoding. The second is periodic encoding. Third is normalized. Fourth is by assigning zero in cases where the state is excluded, and the final and fifth type is removed features, which removes unwanted features. The one-hot encoding approach is a well-known technique for categorical features. The basic premise is to construct extra features based on the unique values of the features. In our situation, the one-hot encoding technique was utilized for the days.

The periodic encoding is used for cyclical features such as hours and months, wind speed, and so on. The cyclical variables are mapped onto a circle, and the internal circle component is computed using sin and cosine. The internal circle component is then normalized by dividing it by the feature's maximum value. The third encoding is a normalizing procedure in which the mean of the characteristics is subtracted and then divided by the standard deviation. The fourth encoding phase is to zero-out any states that are not utilized by the user (two states), and the last one is to eliminate undesirable features. Table (3.3) shows every state with corresponding encoder.

Table 3.3 – States & Encoders

State	Encoder
Month	Periodic normalization
Day	One hot encoding

Hour	Periodic normalization
Daylight savings status	One hot encoding (in case TRUE value)
T out	Normalize
T out prediction 6h	Normalize
T out prediction 12h	Normalize
T out prediction 24h	Normalize
RH out	Normalize
RH out prediction 12h	Normalize
RH out prediction 12h	Normalize
RH out prediction 24h	Normalize
Diffuse solar rad	Normalize
Diffuse solar rad prediction 6h	Normalize
Diffuse solar rad prediction 12h	Normalize
Diffuse solar rad prediction 24h	Normalize
Direct solar rad	Normalize
Direct solar rad prediction 6h	Normalize
Direct solar rad prediction 12h	Normalize
Direct solar rad prediction 24h	Normalize
T in	Normalize
Average unmet set point	Normalize
Rh in	Normalize
Non shift able load	Normalize
Solar gen	Normalize
Cooling storage soc	Normalize
DHW storage soc	Normalize
Electrical storage soc	Remove features
Net electricity consumption	Normalize
Carbon intensity	Periodic normalization
Month	One hot encoding

Day	Periodic normalization

If a building does not have a certain state, flags can be added to indicate that it should be deleted or handled later. For example, if a building does not have a solar photovoltaic system placed on its roof, it cannot have any state associated with solar generation or solar radiations, including diffuse and direct radiations. If the DHW and cooling yearly demand are equal, the state of charge of the thermal energy supply device, such as a heat pump or an electrical heater, should likewise be equal to zero. If it is not, a flag should be set, and those features removed. This can help in the construction of the regression model and initial estimates for the method. For this purpose, a new encoder called the regression encoder is built to estimate state transitions and transformations. It comprises the same components as the standard encoder.

The algorithm iterates until a solution is discovered during the action's selection stage. Prior to picking an action, the environment must be reset in each episode. Keeping in mind that the case study has a total of one episode. The environment reset method resets the environment's state and returns a state. Keeping in mind that the state (observation) reflects one of the environment variables, and it will be restored after performing an action. But the one used in the main file was created by the developer. After resetting the environment, an action must be made to interact with it. This interaction will be chosen iteratively. In this technique, the time step (which symbolizes the hours) will grow by one whenever the agent performs a step. If the value of the time step is smaller than a specified threshold known as the exploration period, the action will be picked (the threshold is 7500 hours, which represents 85.6 percent of total hours during a year). Then, the agent will enter exploration mode. In this phase, the RBC will be employed.

If the time step equals another threshold known as the regression period, the agent will utilize the regression model for the 500-hour period. The agent will then receive the environment variables. As time passes, the environment variables will be updated, the environment variables are:

- **The next state**: And called the observations, and it refers to the new state we are in.
- **The reward**: And it refers to the amount of the reward obtained after performing an action.
- **A variable**: Which tells if the optimal solution is found or not.
- **The information variable**: which used for debugging purposes.

In this study, two types of agents are used: centralized and decentralized agents. The decentralized agent performs actions and establishes relationships between the various components of the action array. If the building has a battery system for storing electrical energy and a cooling storage, a method for calculating the cooling storage would be used. The action represents the amount of cooling energy stored in that particular time step and is calculated as a ratio of the energy storing

device's maximum capacity. There are two potential outcomes: either the action is greater than zero, or less or equal to one (0<action<=1). In this instance, the energy storage system will discharge energy into the building, lowering its state of charge (SOC). The second scenario is when the action is more than or equal to minus one but less than zero (0>action>= -1), In this situation, the energy storage unit receives energy from the energy source, which is often a heat pump, and the state of charge increases proportionately. There are three distinct types of constraints on actions:

- The power capacity of the cooling supply device.
- The cooling demand of the building, as this constrains the quantity of cooling energy that the storage unit can offer.
- The energy storage device's state of charge (heat pump in this case).

Then, the cooling power of the device is then calculated and subtracted from the cooling demand provided. This is accomplished by multiplying the nominal power of the device by the coefficient of performance. Which takes the maximum amount of power that the heat pump can consume and returns the maximum amount of cooling energy that the heat pump can provide (COP). Both the nominal power of the device and the coefficient of performance (COP) had been determined at that time step using equations (14), (15), (21) and (22). This calculates the cooling power supplied to the storage device in order to enhance its state of charge (SOC). This cooling power relates to the available cooling power, to which the device will be charged an equal amount.

After calculating heating and cooling demand at a specific time step, solar generation and non-shiftable loads will be calculated in order to define the building's electricity consumption. Which is equal to the non-shiftable loads added to the heating electrical demand, cooling electrical demand, and battery electrical demand, and then subtracted from the solar generation. For a more detailed explanation, see equation (41).

Buildings share a limited amount of information. The information sharing variable is a Boolean variable. If True, the RBC action selection technique is used. The features (X) and labels (Y) are defined by stacking all the current states and concatenating them with the building's actions. The result of this step will be the variables (X) regression, which represents the features, and (Y) regression, which represents the predicted net electricity consumption using linear regression. This can serve as an initial estimation for the agent. If there is no information sharing between the buildings, the action selection method will be identical to the one shown in equation (33).

If the simulation time exceeds 500 hours, another variable known as coordination variables will be introduced. This variable is an array with two values. The first value is (C), which represents the sum of the capacity sent by each building. This capacity may be written as indicated in equation (40).

$$C = \sum_{i=0}^{n} \left(\frac{E_{coefficients}}{N} \right) \quad (40)$$

Whereas ($E_{coefficients}$) is an approximate estimate of annual energy consumption that may be determined using the formula in equation (41), (N) is the total number of hours in a year.

$$E_{coefficients} = \left(\frac{A}{\eta_{\text{eff}}}\right) + \left(\frac{B}{COP}\right) + (C) - \left(\left(\frac{D}{8760}\right) \times \left(\frac{1}{6}\right)\right) \quad (41)$$

Where (A) is the heating thermal energy produced by the electrical heater, divided by the heater's efficiency to obtain electrical energy. (B) is the cooling thermal energy produced by the heat pump, divided by the heat pump's coefficient of performance to obtain electrical energy. (C) are the building's non-shiftable loads. (D) is the solar photovoltaic output divided by the total number of hours in a year, which is then divided by an estimate of the amount of energy generated by the photovoltaic panel. The second value of coordination variables is the total of expected demand. This prediction was made using linear regression. This value is scaled to account for dispatched capacity and may be stated as indicated in equation (42).

$$E_{predict} = \frac{\sum_{i=0}^{n} (P_{total} - P_i)}{C} \quad (42)$$

There are two types of buffers used in our model: one with a large capacity for model variables and another with a smaller capacity for regression variables. Because the regression period will not begin until 500 hours of random exploration. All environment variables will be placed into the regression buffer in order to be fitted to the regression model. The main goal of fitting is to identify the best fit, which is a straight line with the least divergence between linked and dispersed data points. After 600 hours of random exploration and after storing more values on the replay buffer than a predefined batch size. Batch size refers to the number of training iterations performed in a single iteration, in our case 256. We normalize all current states and rewards to have a mean of 0 and a standard deviation of 1, and then apply the PCA algorithm that was initialized earlier. Finally, a new buffer was established to hold all of the normalized and compressed values, in order to replace the previous one.

The mean can be calculated directly using the NumPy library, but keep in mind that the mean is calculated on the first axis (axis = 0), and with a buffer capacity of (256,38) in the first iteration, the final result will be an array with the shape (,38). Since we are only calculating the mean on the first axis, the final result will be an array with the shape (,38). The standard deviation is then

determined. It is equal to the square root of the average of squared deviations from the mean and may be stated as given in equation (43).

$$\sigma = \sqrt{\frac{\sum_{i=0}^{n} (x_i - \mu)^2}{N}} \tag{43}$$

In equation (48), (σ) denotes standard deviations, (x_i) denotes the sample size (population), (N) denotes each value in the population, and (μ) is the mean. The standard deviation will be determined in the same manner as the mean (axis equals 0). The current state (s) will then be normalized by removing the sample mean (μ) and dividing it by the standard deviation, as seen in equation (44).

$$s_{norm} = \frac{(s - \mu)}{\sigma} \tag{44}$$

The same steps will be used to calculate the mean and standard deviation of the rewards, but the normalization step will be different this time. Normalization will be accomplished by dividing the calculated standard deviation by a predefined reward scaling factor, which equals 5, as specified in [133]. The normalized rewards can be expressed as shown in equation (45).

$$r_{norm} = \frac{\sigma}{Scale_R} \tag{45}$$

Whereas (σ) is the standard deviation of the rewards and ($\mathrm{Scale_R}$) denotes the scaling factor for the rewards. Then PCA will be used to compress our data. Each row will represent a vector. The primary goal of PCA is to compress data in such a way that the output is long, not tall. In other words, has fewer rows than the input data, and the columns equal or exceed the number of components specified previously. In our case, where the number of components is equal to 36, the PCA fitting method is essentially learning how the output would be suitable for the supplied number of components, and then transforming and projecting each row of our data into the learnt vector space.

Each row of input data corresponds to one row of output with a number of columns equal to or less than the number of components. PCA is typically employed when the input data contains a large number of columns, and we need to minimize the number of columns. As a result, a new buffer will be created that will be used to hold the current state, the next state, rewards, and actions. The final stage of the training phase is to train the DL neural network. To do so, we will take a random sample from the buffer with a size equal to the batch size (256). The random sample contains the following variables: current states, next states, rewards, actions, and an indicator variable that

indicates whether or not optimal solutions were discovered. The network's basic structure and initialization procedure are already illustrated.

3.4 Optimization schedule

The optimization schedule is used to train the neural network. The model parameters are tuned using a technique called backpropagation. In which the loss function is pumped from the last to the first layer. The backpropagation method relies on the chain rule of differentiation and gradient descent. The weights are then updated at each layer to minimize the loss. The updated weights are dependent on the learning rate. The entire process of updating the weights is referred to as the optimization schedule. There are numerous optimization techniques available for this task. The one utilized in the model is termed Adaptive Moment Estimation (Adam), and it is considered the most famous and widely used optimization schedule. However, before discussing (Adam), and because its functionality is dependent on the operation of other optimization schedules. A quick review of the optimization schedules' functionality will be provided. All information here is sourced from [141]. There are numerous optimization schedules available, including the following:

- **Stochastic Gradient Descent (SGD):** It is used to perform update for the model parameters, as follow:

$$\beta = \beta - \alpha \times \frac{\delta L(X, y, \beta)}{\delta \beta} \tag{46}$$

  Whereas (β) is a parameter that requires optimization, (X) is the input data, and (y) denotes the labels, (L) denotes the loss and (α) is the learning rate. In (SGD) parameter updates are performed on each pair of (X, y), and we can see that we are using a single learning rate for all parameters, yet individual parameters may require a different learning rate.

- **Adagrad**: The previous method updates the parameters on a per-(X, y) pair basis. This optimization schedule, the update is per-parameter, and the learning rate is not constant, as in the (SDG). Because we may need to update the parameters at a different rate, especially with sparse data. The (Adagrad) can be expressed as shown in equation (47).

$$\beta_i^{t+1} = \beta_i^t - \frac{\alpha}{\sqrt{SSG_i^t + \epsilon}} \times \frac{\delta L(X, y, \beta)}{\delta \beta_i^t} \tag{47}$$

There are multiple parameters in this method, which is why the subscript (*i*) is used, and it refers to the *ith* parameter and the superscript (*t*) to the iteration's time step. (SSG) represents the sum squared gradients for the *ith* parameter, which will be updated in each iteration. Keeping in mind that (ϵ) represents a small value to be added to the (SSG) to avoid division by zero. Due to the fact that the parameters are constantly changing. This strategy ensures that the learning rate will be slower than the prior way.

- **Adaptive Moment Estimation (Adam):** The prior approach (Adagrad) used an increasing denominator for the learning rate. Which resulted in the learning rate disappearing as it continued to decrease. This is why it was improved by introducing a decay factor, which computes the average of the previous gradient. This is referred to as Adadelta. Adam is another scheduling approach that can create a customized learning rate for each parameter. It, like the Adadelta method, uses a decaying average. Adam can be stated as indicated in equation (48).

$$\beta_i^{t+1} = \beta_i^t - \frac{\alpha}{\sqrt{SSG_i^t + \epsilon}} \times SG_i^t \tag{48}$$

Whereas (SG_i^t) is the sum of gradients and is similar to the first moment of gradient estimation. Thus, this method is referred to as adaptive moment estimation (Adam). For the purpose of estimating the first moment gradient, the decaying average is determined as shown in equation (49).

$$SG_i^t = \gamma' * SG_i^{t-1} + (1 - \gamma') \times \frac{\delta L(X, y, \beta)}{\delta \beta_i^t} \tag{49}$$

The (SSG^t) represents the total squared gradient, which is theoretically identical to the second moment of gradient estimation. The decaying average can be determined similarly to the Adadelta approach, as shown in equation (50).

$$SSG_i^t = \gamma * SSG_i^{t-1} + (1 - \gamma) \times \left(\frac{\delta L(X, y, \beta)}{\delta \beta_i^t}\right)^2 \tag{50}$$

The values of (γ) and (γ') are typically close to 1, which indicates that both (SG_i^t) and (SSG^t) can have zero initial values. To reach to a solution, we must apply a correction factor to both (SG_i^t) and (SSG^t):

$$SG_i^t = \frac{SG_i^t}{1-\gamma'} \tag{51}$$

And sum squared gradient can be expressed as:

$$SSG_i^t = \frac{SSG_i^t}{1-\gamma} \tag{52}$$

Then, we correct the equation by substituting revised values (48). Adam is widely regarded as one of the most successful optimization schedules for training complex DL networks. Adam's three main hyperparameters are as follows:

- Learning rate
- The two decaying rates for the gradients and the square gradients.

3.5 Reward function

In RL, the reward function serves as a guide for the agents. It assists agents in learning through trial and error. As mentioned previously, the goal is to maximize the total rewards over time, as shown in equation (4). Rewards are used as an incentive mechanism to communicate to the agent which actions are correct and which actions are incorrect. All of this can be accomplished through the use of rewards and penalties. There are numerous reward functions in the CityLearn framework, and they vary according to the type of agent, the number of agents, and the correlation information between buildings. If the agent is centralized, the reward is determined by the network's electricity consumption. If the agent is decentralized, the reward is determined by the building's correlation information. This framework contains three distinct reward functions, the one used in this case study is the one given in equation (53) and (54).

$$r_i^1 = min\{0, e_i\} \tag{53}$$

$$r_i^2 = -sign(e_i) \times min\{0, e_i\}^2 \tag{54}$$

Whereas ($\boldsymbol{e_i}$) reflects the net electricity consumption. If the building consumes more power than it creates, the electricity demand will be negative, which is why the sign is utilized in equation (55). However, it is critical to remember that the superscript on the left is not exponential, but rather refers to the reward function's number.

$$r_i^3 = min\{0, e_i\}^3 \tag{55}$$

If we employ the SAC method, we will use the implementation in equation (55). This reward function assumes that all agents are autonomous and do not share knowledge. According to [133] increasing the exponential increases performance. However, exponentials greater than three have no effect.

$$r_i^{MARL} = -sign(e_i) \times {e_i}^2 \times min \left\{0, \sum_{i=0}^{n} e_i\right\} \quad (56)$$

If we use a multi-agent algorithm, such as MARLISA, we will use the reward function in (56). This reward function multiplies the district's net electricity consumption by the net electricity consumption of individual buildings. This will assist in reducing the district's and individual buildings' electricity consumption. Because this reward function is non-linear, the penalty can grow polynomially with net electricity consumption. Greater demand values contribute more than lower demand values [133].

3.6 Evaluation Metrices

This section addresses the framework's objective function, which represents the function that the agent is attempting to reduce. It has six distinct cost functions, each expressed as a function of total net electricity consumption [2]:

- **Ramping**: It represents the net electricity consumption at every time step, and it has a non-negative value, ramping can be expressed as shown in equation (57).
- **1-load factor**: It is defined as the ratio of real electrical energy kWh consumed during specified periods divided by the maximum electricity load, or in other words, divided by the total amount of energy that might have been consumed simultaneously by the user. A high load factor implies efficient power use, whereas a low load factor shows wasteful electricity consumption. Because the measure is 1-load factor, not load factor, the purpose is to decrease, not maximize, the load function (58).
- **Peak demand**: Represents the maximum peak electricity demand.
- **Average daily peak**: It represents the average daily peak net demand.
- **Net electricity consumption**: It is the total amount of electricity consumed.
- **Quadratic**: It is the square of the total sum of the net electricity usage. This cost function is implemented on the initial implementation but is never used or examined.
- **Carbon emissions**: It represents the amount of carbon emissions produced by the building.

- **Coordination variables**: Is the mean of all the metrics except the net electricity consumption, and the carbon emission, because buildings exchange some information, but not all of it.
- **Total**: The mean value of all the other metrics.

$$Ramping = \sum_{i=0}^{n} |E_i - E_{i+1}| \quad (57)$$

Whereas (E_i) denotes the current time step's net electricity consumption and (E_{i+1}) denotes the future time step's net electricity consumption. The approach will calculate both the current year's ramping and the prior year's ramping.

$$LF = \sum_{i=0}^{N} \left(\frac{E_i}{Max\{E_i\}}\right) \div (\mathrm{N}) \quad (58)$$

Whereas (E_i) denotes the net electricity consumption, and N is the mean value, as we are working with arrays rather than singular and scalar quantities. A Numpy array is used to represent the net electricity use. The final objective function is the quadratic objective function, as indicated in equation (59).

$$f(x) = \sum_{i=0}^{n} {E_i}^2 \quad (59)$$

Whereas (E_i) reflects non-negative net electricity consumption, the quadratic function is a polynomial function with one or two variables that is frequently employed in combination with linear functions in optimization tasks [143]. The quadratic function is not employed in this research, and it will not be used as an evaluation metric. The first step in computing the cost function is to estimate the RBC and the baseline cost. The real cost function is then calculated and divided by the normalization cost derived from RBC, both cost functions for the previous year and the first year will be calculated to provide insight into the agent's performance over time.

4 Case study and Results

4.1 Overview

This phase analyses the received output and assesses the performance and convergence of all agents. The results demonstrate the implementation of various types of RL algorithms for load shaping and DR on a group of nine buildings, five of which are residential. They also demonstrate the extent to which it is feasible to rely on RL optimization techniques, bearing in mind that the approaches used in this research are model-free approaches that do not require a model design process and the system dynamics are unknown. These simulations, which are run in a single episode, make use of a variety of different forms of data.

All data in the framework are collected and utilized as test data by [2], and they are pre-simulated using a variety of softwares, including EnergyPlus, MODELICA, and real-world data, but users may also use their own dataset. CityLearn obtains the simulated solar photovoltaic generation per kilowatt data from SAM [146], It is a free program used for research and PV modelling purposes, and then CityLearn multiplies the datasets by the output of a pre-simulated inverter to acquire the solar energy produced in kilowatt-hours. The weather data is classified according to US climate zones and includes the following cities and climate zones; however, keep in mind that all of the climate zones used in the environment are illustrated in table (3.1):

- 2A | Hot-Humid | New Orleans
- 3A | Warm-Humid | Atlanta
- 4A | Mixed-Humid | Nashville
- 5A | Cold-Humid | Chicago

All buildings have their own cooling and heating systems; all buildings have an air-to-water heat pump that is used for both cooling and heating; however, not all buildings have an electric heater for heating. These devices, together with other electrical devices such as non-shiftable loads, used to meet customer needs, and it consumes electricity from the main feeder. Keeping in mind that the installed photovoltaic systems offset only a portion of the electricity consumption, not the entire amount. Non-shiftable load profiles of residential buildings were obtained from Pecan Street in Austin, Texas [147], and then the author in [2] trained a probabilistic regression model using a neural network with a soft-max output layer on the data obtained from [147], in order to have multiple load profiles that could be injected into EnergyPlus and then used by CityLearn [2]. The environment used data from the solar row project [148] for the domestic hot water (DHW) load profiles, and the author generated both cooling and heating temperature set-points from the Restock project [149], which is a platform for modelling existing residential building stocks at national and local scales.

The purpose of this phase of the research is to determine if RL can increase performance and produce an appropriate load shape for use in DR. Keeping in mind that the agent has no prior knowledge of the system dynamics and no model to follow, the results will be divided into three parts: The first section is devoted to the single agent (centralized), and the second is the decentralized agent, and the third is the RBC, which serves as the reference answer on which the performance of other agents is compared. The results will begin with demonstrating the electricity needs using RBC and without the use of a control agent, followed by demonstrating the results utilizing MARLISA for decentralized agent, and SAC for both centralized and decentralized agents. Keeping in mind that all results are simulated in climate zone 5, which is classified as a cold-humid environment, all results will include three distinct scenarios: summer operation (10 days), winter operation (10 days), and overall operation for the whole period (4 years).

4.2 Load shape without using agent:

4.2.1 Results

This section illustrates typical electricity consumption; it includes a four-year simulation of two scenarios: electricity demand without the use of a photovoltaic system or storage control by an agent, and electricity demand with the use of a photovoltaic system but without the use of a storage control. The objective functions are listed in Table (4.1).

Table 4.1 – Objective function without using agent

Objective functions	Cost function	Cost function for the last year
Ramping	1.015	1.017
1-Load factor	1.138	1.128
Peak demand	1.09	1.124
Average daily peak	1.116	1.121

Net electricity consumption	0.987	0.987
Carbon emissions	1.000	1.002
Coordination score	1.090	1.098
Total	1.057	1.068
Simulation time (min)	2.864	2.864

4.2.2 Analysis

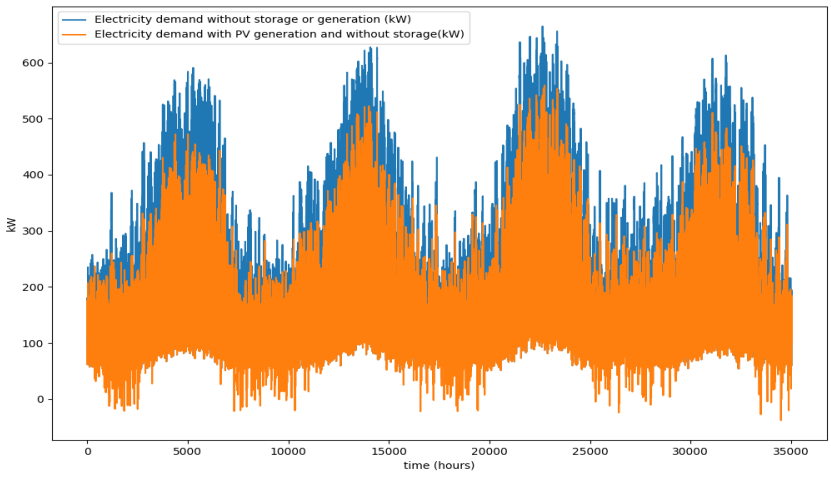

Figure 4.1 – Demand Without storage control

The demand simulations in Table (4.1) and Figure (4.1) demonstrate that while employing PV systems can offset some of the power need, reducing the demand further requires another way. The

demand peaked at roughly 650 kWh in the third year, and the PV system never surpassed the demand. For further information on building's electricity consumption, DHW demands, and cooling demand, please see Appendix C.

4.3 Baseline results using RBC

4.3.1 Results

In this section, a rule-based controller (RBC) was used to obtain baseline results, which will be compared to the results of other algorithms. The RBC was well tuned by the author in [133], and all objective functions are normalized by the (RBC), so if we implemented another algorithm and obtained a score in any evaluation metric equal to 0.88 (for example), this indicates that our agent performed better than the (RBC) by 12 percent [133], RBC objective function results are all equal to one, because it is normalized by itself (divided by itself), so there is no need to write it in this part.

4.3.2 Total operation

We ran a simulation over a four-year period for evaluation purposes, and the results are considered for two seasons: winter and summer. The reason for this is to assess the performance and agent adaptation to various scenarios and electricity demands, as well as different load profiles. The cooling and heating device of choice for almost all buildings is the heat pump, although some buildings have an electrical heater installed. However, in this chapter, we will use the building number five to evaluate the performance of the heat pump, Figure (4.2) shows the total simulation duration (4 years).

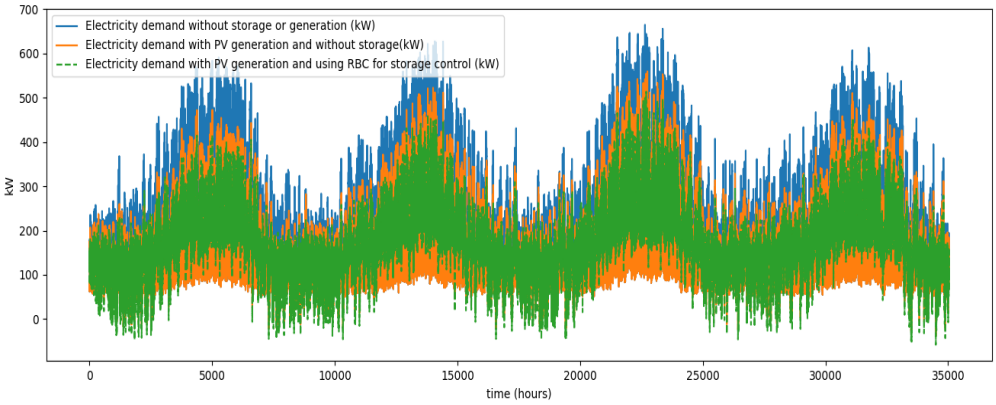

Figure 4.2 – Total simulation using RBC

4.3.3 Summer operation

4.3.3.1 First year

Different situations are considered. For the summer operation, and to allow for a more accurate evaluation of performance, we study two periods: the first-year summer and the last-year summer; this helps us comprehend the progression of the agent's learning process. The summer operation for the first year is depicted in figure (4.3).

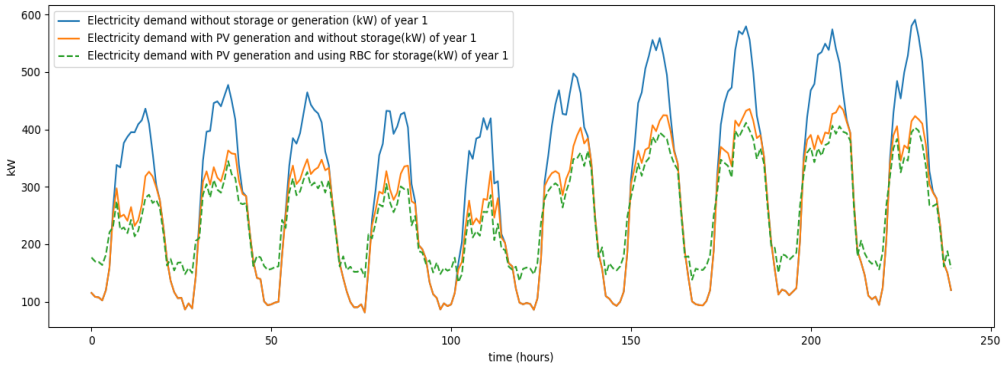

Figure 4.3 – Summer operation by the RBC (first year)

4.3.3.2 Last year

Figure (4.4) below depicts the summer simulation from last years. It's worth noting that the results are practically identical to the first year, as the agent is manually tuned and behaves consistently each day. When compared to the typical demand, the results indicate that demand peak decreased from over 600 kWh to almost 350 kWh in both the first and last years.

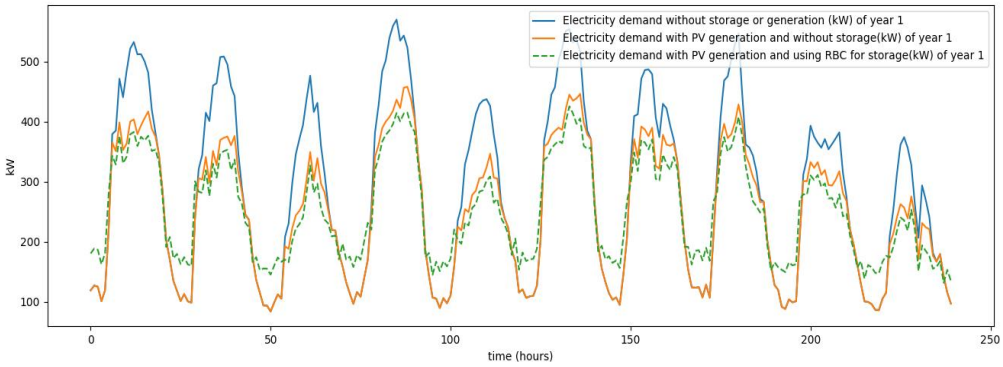

Figure 4.4 – Summer operation by the RBC (last year)

4.3.4 Winter operation

4.3.4.1 First year

As with the summer operation, we separated the winter operation into the first and final years for assessment reasons. The first winter operation is depicted in Figure (4.5), and the final year simulation is depicted in Figure (4.6). The typical consumption contains many peak demands at 225 kWh and 250 kWh the agent reduced the demand to almost half.

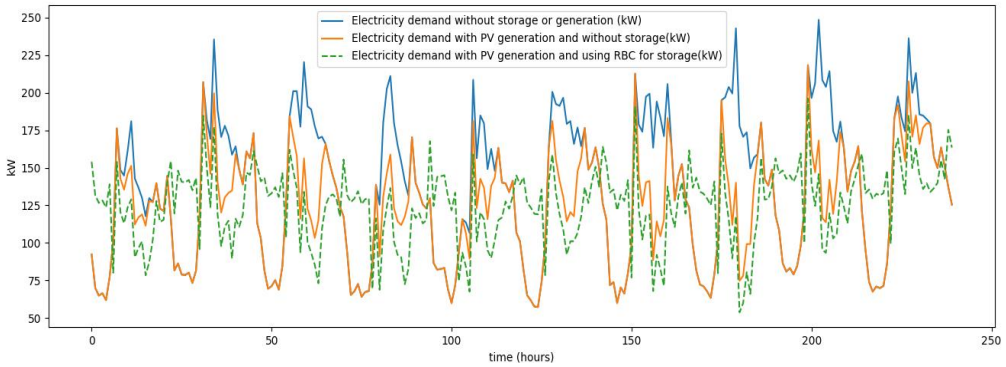

Figure 4.5 – Winter operation by the RBC (first year)

4.3.4.2 Last year

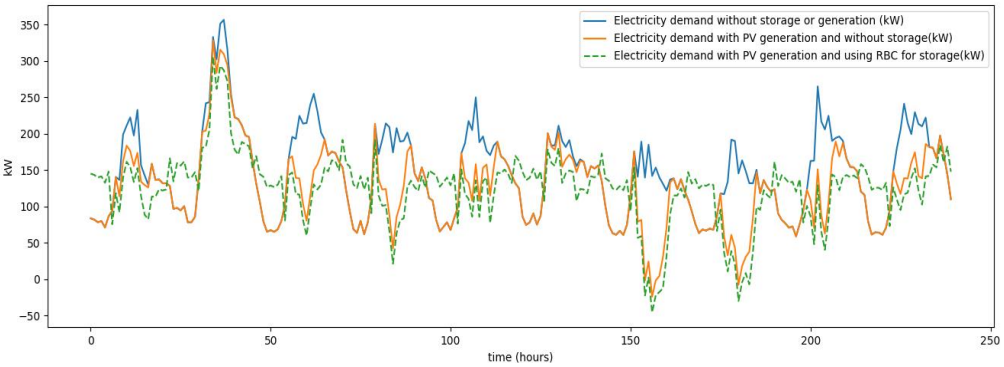

Figure 4.6 – Winter operation by the RBC (last year)

4.3.5 Cooling & Heating devices operation

Figure (4.7) illustrate the cooling demand in 4 days in winter, and how the heat pump is acting according to the cooling demand, the figure also shows energy balance which is equal to the difference between the energy transmitted from the storage system to the building and the energy from the supply device to the storage, and how much the cooling device is consuming energy. And lastly the COP of the heat pump.

4.3.6 Cooling & Heating devices in winter

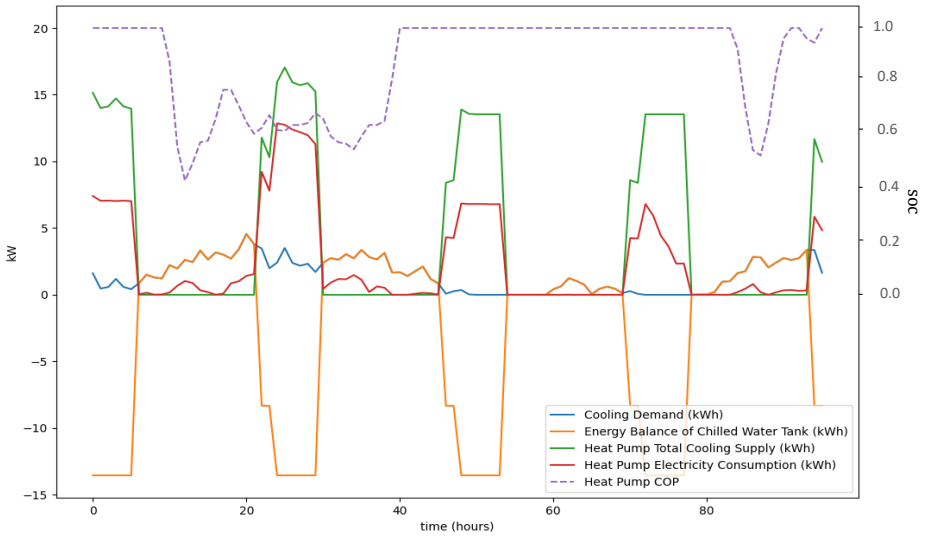

Figure 4.7 – Heat pump operations

Figure (4.8) and (4.9) shows how is the cooling and heating and the battery are behaving, in order to satisfy the building heating and cooling demands, keep in mind that both heating and cooling devices are the heat pump, although we have an option to select the electrical heater as heating device. The simulation shows 5 days of operation during winter.

4.3.6.1 Cooling device

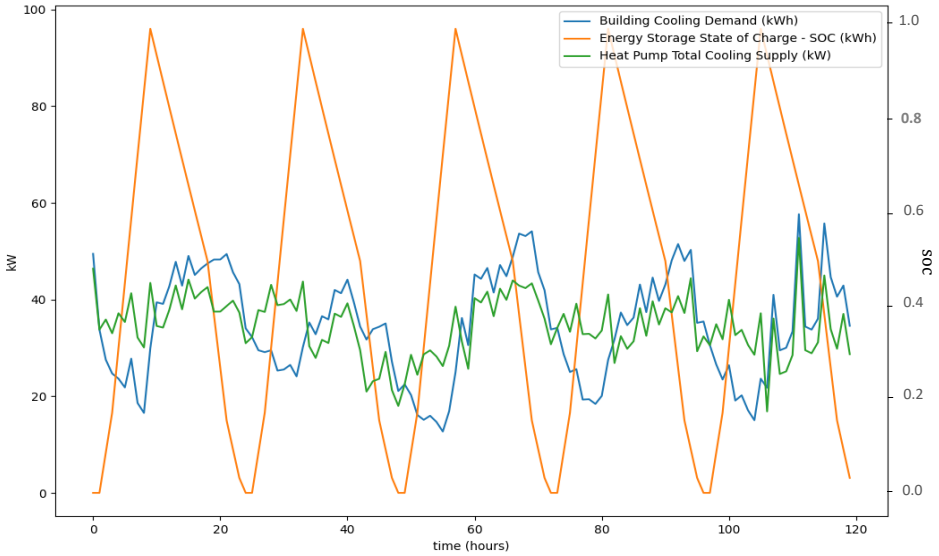

Figure 4.8 – Heat pump & Cooling demand

4.3.6.2 Heating device

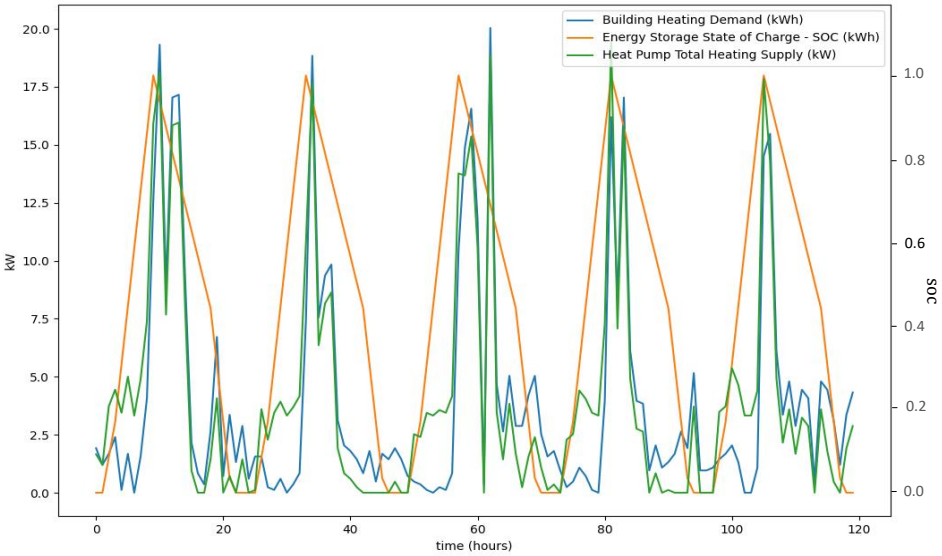

Figure 4.9 – Heat pump & Heating demand

4.3.7 Analysis

The RBC is our reference agent, that's why all the evaluation metrics results are ones, the simulation shows that using RBC for storage control, can be beneficial since it can lower the consumption by a big amount. Simulations shows that RBC always satisfied the building demands, and there was no point where the demand was lower than the ones obtained by the RBC (in both seasons of operation). Figure (4.3) and (4.4) shows that the agent reduced the demand from 600 kWh to less than 350 kWh, this is for both first and last years, and the reason is that the RBC have a consistence and similar performance. The basic concept of the RBC agent, that it stores the energy at night and then releases it at morning when the COP of the heat pump is higher, and this can be noticed by viewing figure (4.7), we can notice that if the COP of the heat pump is low the energy released by the heat pump is going to be low and vice versa. Figure (4.8) & (4.9) shows the heat pump operation while using RBC in winter, as noticed in winter the heat pump is not used much for cooling purposes, but it is used heavily for supplying heat energy.

4.4 SAC

4.4.1 Centralized agent

4.4.1.1 Results

A simulation procedure was done employing a single agent (centralized), utilizing a SAC algorithm implemented by stable baselines [150], the implementation was included also in the

intelligent-environment-lab repository [151], the simulation indicates a 4-years interval, the achieved results are displayed in table (4.2).

Table 4.2 – Objective function using SAC (centralized)

Objective functions	Cost function	Cost function for the last year
Ramping	1.017	1.109
1-Load factor	1.127	1.148
Peak demand	1.110	1.148
Average daily peak	1.151	1.155
Net electricity consumption	1.032	1.033
Carbon emissions	1.045	1.047
Coordination score	1.101	1.008
Total	1.080	1.088
Simulation time (min)	67.569	67.569

4.4.1.1.1 Total operation

A simulation was performed over a four-year period, comparing the electricity demands in three scenarios: the first, in blue, represents the electricity demand without the use of photovoltaic systems and without the use of a storage system controlled by a single agent; the second, in orange, represents the electricity demand with the use of photovoltaic systems in buildings and the addition of the generated electricity from the PV cells; and the third, in green, represents the electricity demand with the use of photovoltaic and storage systems controlled by a single control agent using SAC, The agent initially produced very high results; demand was about equal to 450 kWh in the first year, owing to the exploration phase, but demand fell after that. All three possibilities are depicted in Figure (4.10).

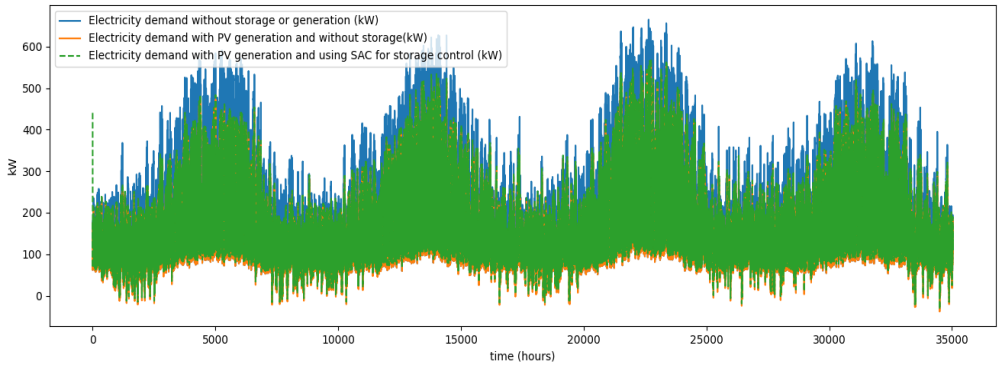

Figure 4.10 – Total simulation using SAC (Centralized)

4.4.1.1.2 Summer operation

Figure (4.11) (4.12) illustrates a simulation process for the summer only. In this case, the agent will adjust the control process based on weather data and electricity consumption. Because the cooling device (Heat Pump) demand will increase during the summer, the simulation is divided into the first and final years to evaluate the agent's learning process.

First year

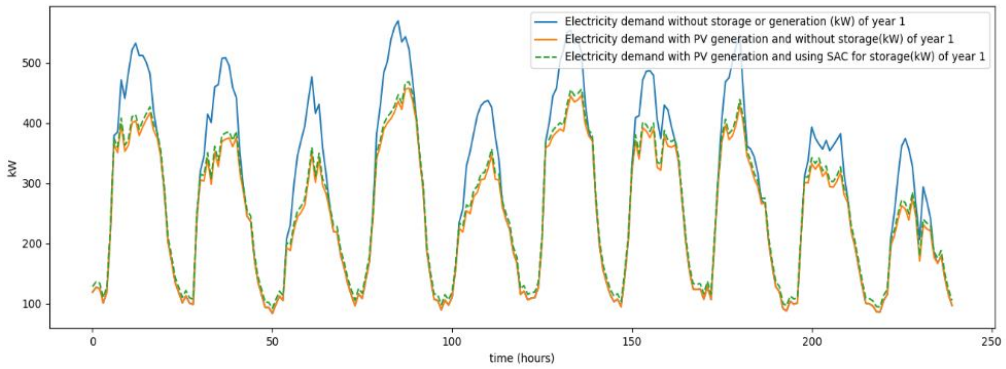

Figure 4.11 – Summer operation by the SAC (first year)

Last year

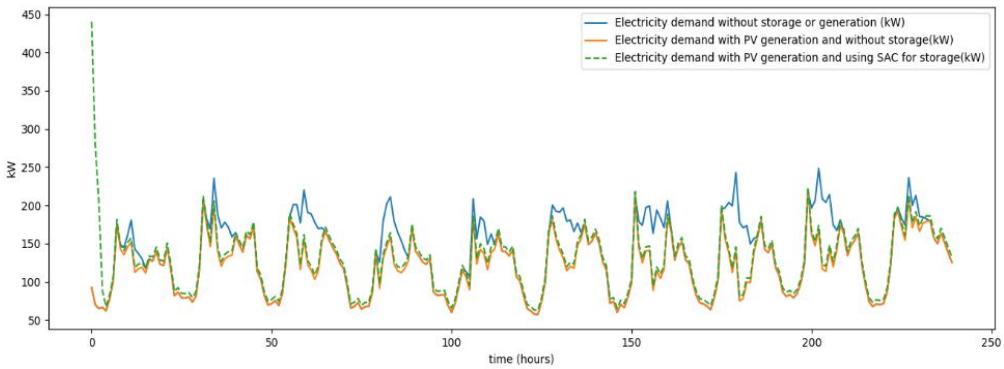

Figure 4.12 – Summer operation by the SAC (last year)

4.4.1.1.3 Winter operation

A simulation process only for 10 days of winter, the agent will change the control process to accommodate the winter load profiles, since the heating device (Heat pump) demand will be different than during the summer operation. The procedure is also divided into two years, the first of which is depicted in figure (4.13), and the second of which is depicted in figure (4.14).

First year

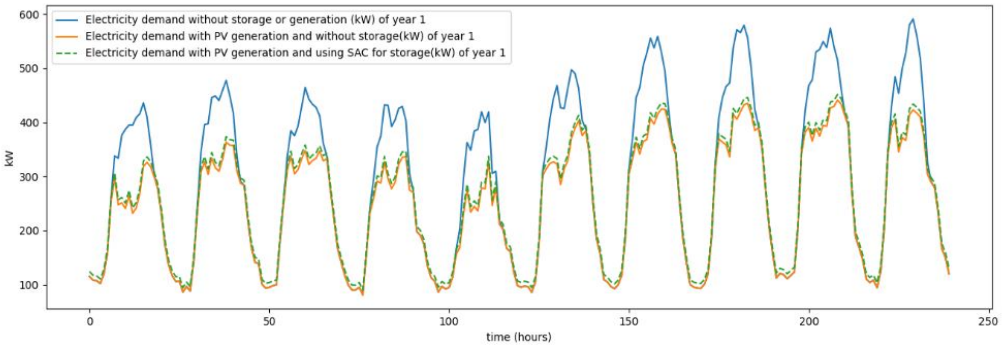

Figure 4.13 – Winter operation by the SAC (First year)

Last year

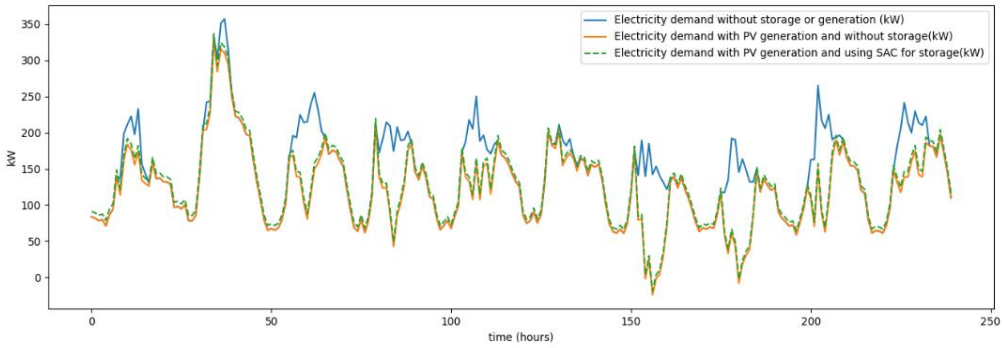

Figure 4.14 – Winter operation by the SAC (Last year)

4.4.1.1.4 Cooling & Heating device operation

Figures (4.15) and (4.16) illustrate how the cooling and heating devices, as well as the battery, behave in order to meet the building's heating and cooling demands, keeping in mind that both heating and cooling devices are heat pumps in this case. Additionally, because we are using building 5, a residential multi-family building, the simulation depicts operation during the winter, when heating demands are significantly greater than cooling demands.

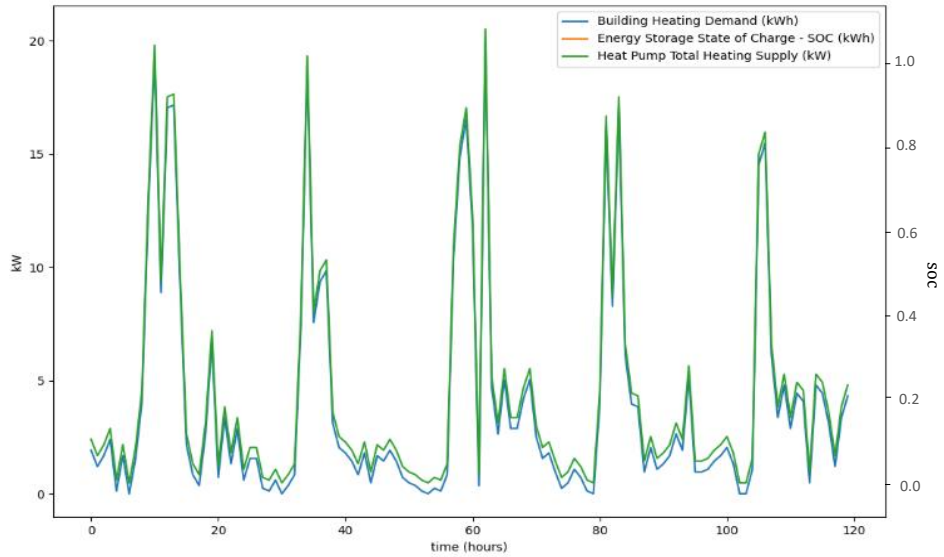

Figure 4.16 – Winter Heat pump operation as a heating device

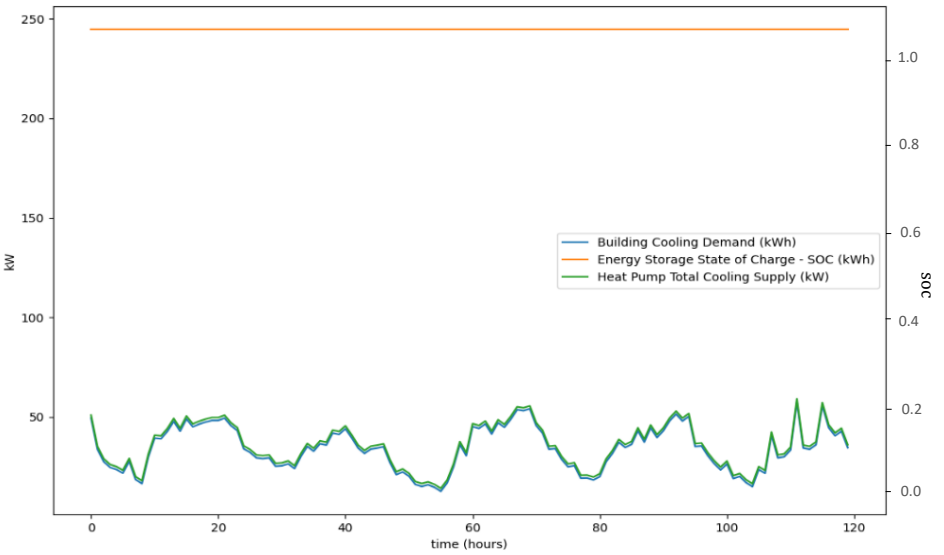

Figure 4.15 – Winter Heat pump operation as a cooling device

4.4.1.2 Analysis

When SAC is used as the centralized agent, the results indicate an increase in the ramping cost function when compared to the RBC agent, and even when compared to the first year of simulation, indicating a decline in performance in this area, as well as a decline in the (1-LF) cost function. Additionally, both average peak demand and peak demand show an increase when compared to the first year of operation, and an increase in the net consumption which resulted in an increase in the carbon emissions, Finally, the coordination score between the various buildings has increased, but keep in mind that all of the buildings are controlled by a single agent, and the reason for this is due to the algorithm utilized, which was created using stable baselines [150] , and for a good performance all the agent parameter, needs a tuning process, such as: learning rate, and discount factor ..etc, another reason can be that there is no battery system used in the process, while using single agent.

Figure (4.10) illustrates the total operation over a four-year period. When compared to a scenario in which there is no storage control agent and the PV system is used alone, the agent is barely meeting the building demands. This could be because the agent is not using or relying on a battery system, and this due to the size of the battery in the data sheet, it is difficult to use a battery that can satisfy all the nine buildings. That is why, in the original CityLearn implementation, we can observe that when the agent is centralized, the battery state is excluded from the state and action spaces. Finally, we can see the effect of the battery system's absence in figures (4.15) and (4.16), which depict the functioning of cooling and heating devices. As illustrated, the heat pump can barely meet the heating needs, and the agent's overall performance may be classified as poor performance.

4.4.2 Decentralized agent

4.4.2.1 Results

A simulation process was performed using SAC algorithm for a decentralized agent, the implementation of this agents is done by [2], the obtained objective function results are shown in table (4.3).

Table 4.3 – Objective function using SAC (decentralized)

Objective functions	Cost function	Cost function for the last year
Ramping	1.615	1.163

1-Load factor	1.102	1.055
Peak demand	1.160	1.133
Average daily peak	1.133	1.066
Net electricity consumption	1.015	1.007
Carbon emissions	1.022	1.015
Coordination score	1.255	1.104
Total	1.177	1.078
Simulation time (min)	67.570	67.570

4.4.2.1.1 Total operation

Figure (4.17) depicts a simulation over a four-year period, comparing the effect of using the SAC algorithm for decentralized agents to a scenario utilizing only photovoltaic systems. The regular electricity demand is depicted in blue, the electricity demand using the control agent is depicted in green, and the demand utilizing only photovoltaic systems is depicted in orange. The normal electricity consumption peaks at the third year of simulation at roughly 650 kWh, the demand with

SAC peaks at approximately 600 kWh, and the demand with a PV system peaks at approximately 500 kWh in the third year.

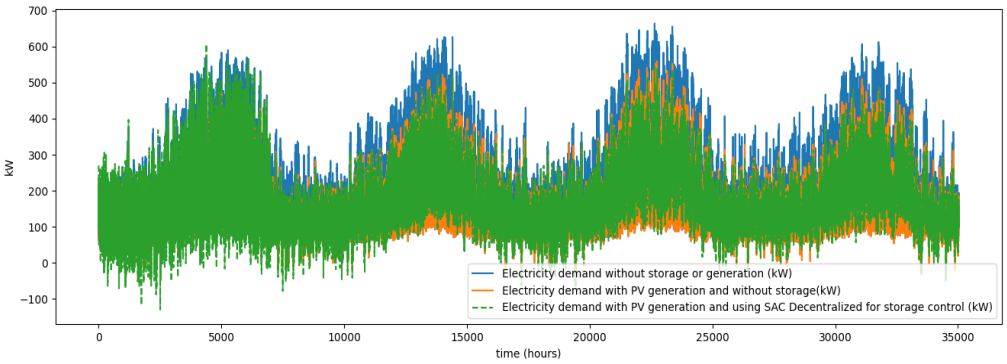

Figure 4.17 – Total simulation using SAC (Decentralized)

4.4.2.1.2 Summer operation

Summer simulations for ten days are depicted in figures (4.18) and (4.19) for the first and last years of simulation, respectively.

First year

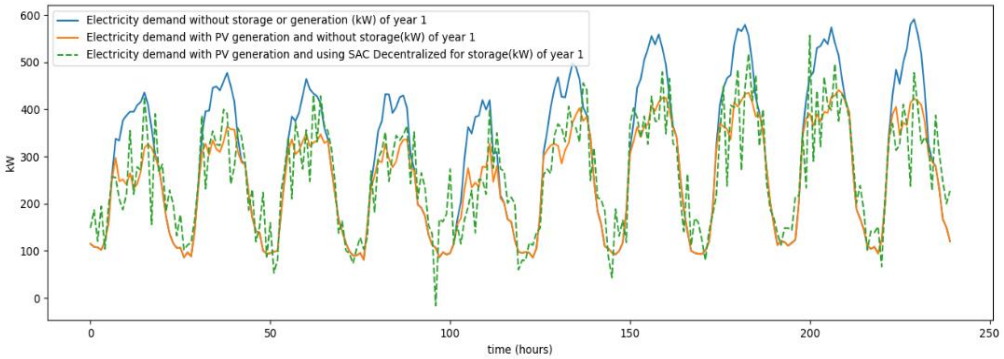

Figure 4.18– Summer operation by the SAC (first year)

Last year

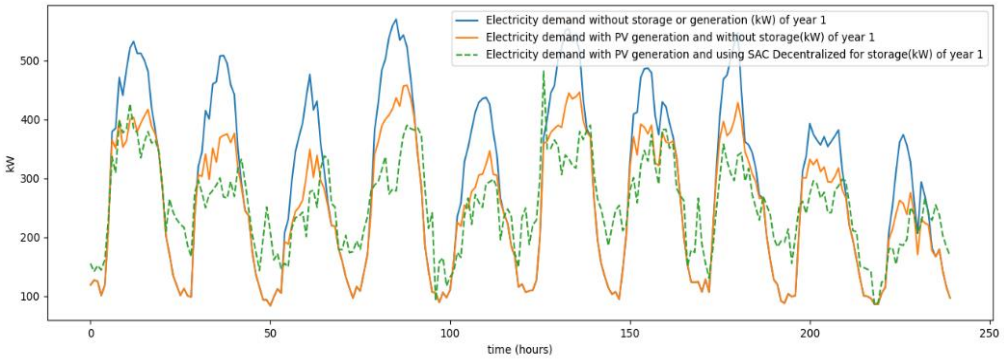

Figure 4.19 – Summer operation by the SAC (Last year)

4.4.2.1.3 Winter operation

This section depicts ten days of winter while using the SAC algorithm for a group of nine decentralized agents (Buildings), and then compares it to a case in which there is no storage control, but a photovoltaic system is installed. The simulation depicts the first year of using the algorithm and the last year, respectively, in figures (4.20) and (4.21).

First year

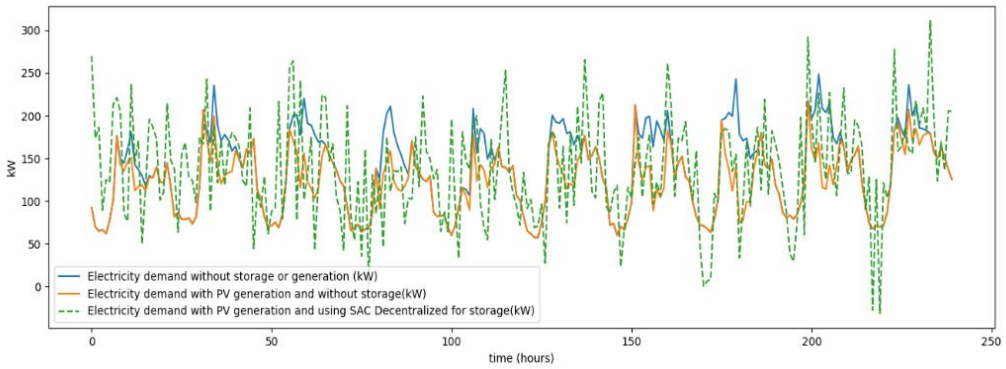

Figure 4.20 – Summer operation by the SAC (First year)

Last year

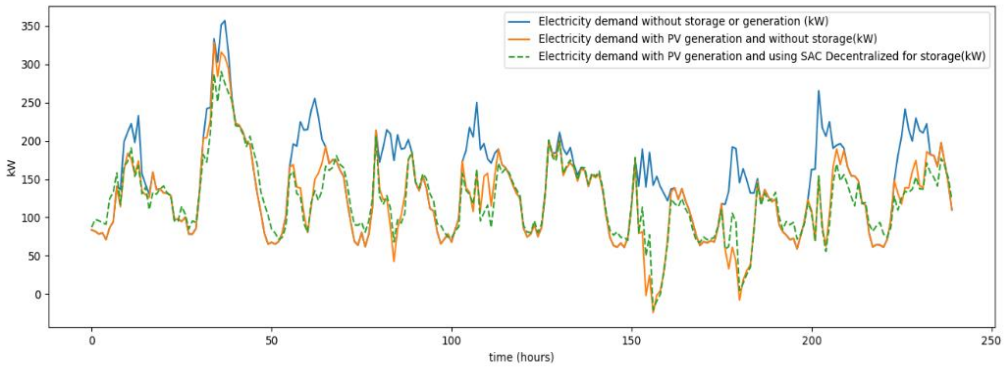

Figure 4.21 – Winter operation by the SAC (last year)

4.4.2.1.4 Cooling and Heating device operation

Figures (4.22) and (4.23) illustrate how the cooling and heating devices, as well as the battery, behave in order to meet the building's heating and cooling demands, while keeping in mind that both heating and cooling devices are heat pumps, although we do have the option of using an electrical heater for heating. Winter operations are depicted in the simulation.

Cooling device

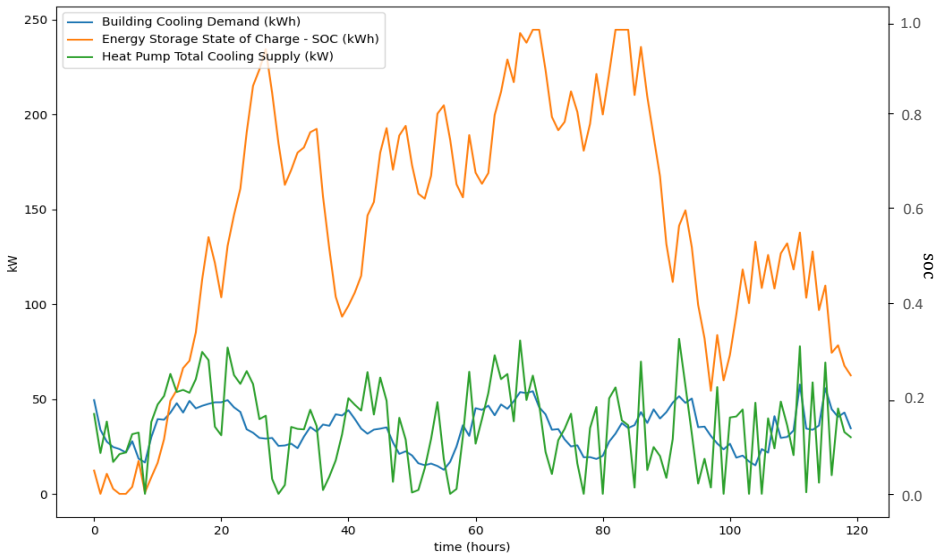

Figure 4.22 – Heat pump & cooling demand

Heating Device

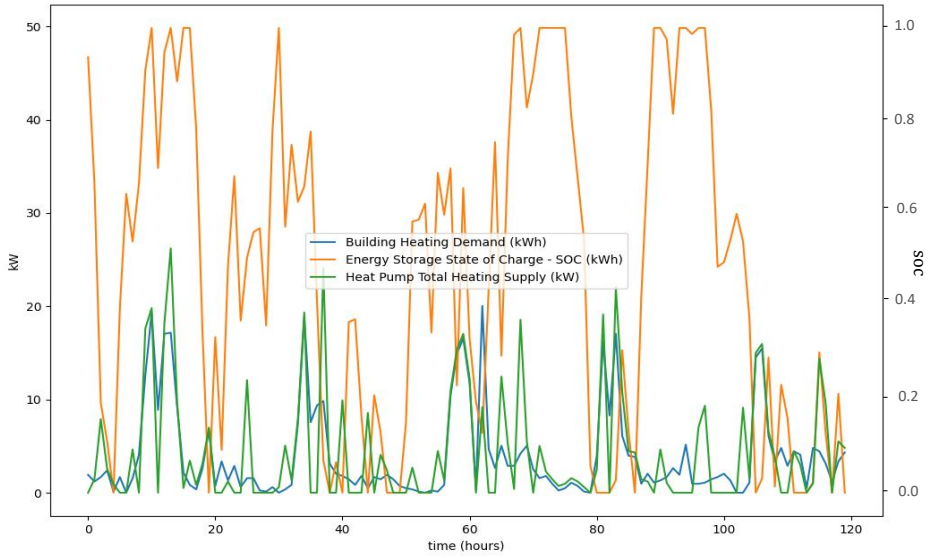

Figure 4.23 – Heat pump & Heating demand

4.4.2.2 Analysis

The results indicate a significant improvement in the ramping cost function, an increase in load factor (and a decrease in (1-LF)), an increase in peak demand throughout the year, and a noticeable increase in average peak demand. Additionally, the net electricity consumption and carbon emissions have improved, the total performance can be almost equal to the performance of the RBC, Although the RBC outperformed the total performance in several areas, the overall performance and

learning process from an agent with no prior knowledge of the environment and employing a primary reward may be classified as good performance.

Figure (4.17) depicts the total simulation over a four-year period. As can be seen, the SAC for decentralized agents performs significantly better than the SAC for centralized agents. This is due to the battery system; as mentioned previously, the single agent does not have a battery model, due to the high demand, and the need for a huge battery capacity to satisfy this amount of demand. In case of the decentralized agent, the figure depicts that the agent began the learning and optimization process with an upward trend on the first year, but then stabilized with the help of deep neural networks for the optimization process, it can be noticed that the agent achieved a result that is better than relying on a PV system alone.

This can be shown by examining both summer and winter operations; it can be seen that the agent first performed poorly, but subsequently improved significantly in terms of outcomes and learning process; Figures (4.20) and (4.21) summarize this. In terms of battery system performance and cooling and heating devices, which in the case of building 5 would be the heat pump, Figures (4.22) and (4.23) demonstrate that the heat pump performed very well in meeting the heat demand. Keep in mind, however, that in some instances, the heat pump delivered more than was required, and this is also true for the battery. The reason for this is that, because we have a group of nine buildings (agents), there will always be coordination between the various agents, as the SAC implementation provides correlation information between the various agents, so, any excessive amount of energy is going to be sent by the net to the other agents to be used.

4.5 MARLISA

4.5.1 Results

4.5.1.1 Safe Exploration

A simulation process was carried out employing decentralized agents, with all agents exchanging information and the safe exploration mode active. Which helps in action selection depending on the RBC agent. The result of the objective functions is shown in table (4.4).

Table 4.4 – Objective function using MARLISA (with safe exploration)

Objective functions	Cost function	Cost function for the last year
Ramping	1.372	1.316

1-Load factor	1.110	1.101
Peak demand	1.197	1.202
Average daily peak	1.107	1.098
Net electricity consumption	1.0	1.001
Carbon emissions	1.007	1.010
Coordination score	1.197	1.179
Total	1.132	1.130
Simulation time (min)	183.610	183.610

4.5.1.1.1 Total operation

A four-year interval simulation, it compares electricity demands for nine different agents (buildings) using the MARLISA in safe exploration mode for storage control in green colour, without storage control but with a photovoltaic system in orange, and finally without both storage control and photovoltaics in blue, which represents the regular demand. Figure (4.24).

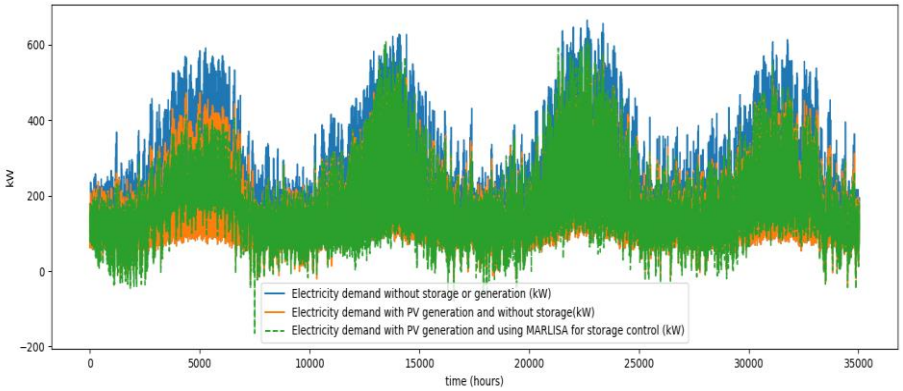

Figure 4.24 – Total simulation using MARLISA (safe exploration mode)

4.5.1.1.2 Summer operation

First year

Figure (4.25) depicts the summer operation of MARLISA in safe exploration mode for 240 hours (10 days) of summer simulation. It corresponds to the first year of simulation. The maximum peak in the figure is approximately 350 kWh and occurred on the second day. The agent provided a reduction to make this peak less than 250 kWh. The agent results always satisfied the demand during the interval except for day six, when the agent output was equal to the demand. During day seven, the agent output was equal to zero. Because the agent is considering the energy provided by the PV system, the performance can be described as good in general.

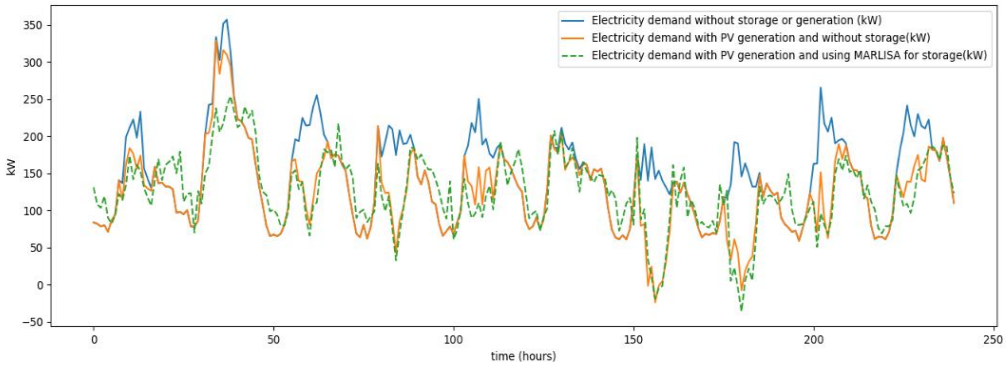

Figure 4.25 – Summer operation by MARLISA (first year)

4.5.1.1.2.1 Last year

Figure (4.26) shows the summer operation using MARLISA in safe exploration mode, during the course of 240 hours of simulation (10 days) of summer, and it is taking place in the last year of simulation.

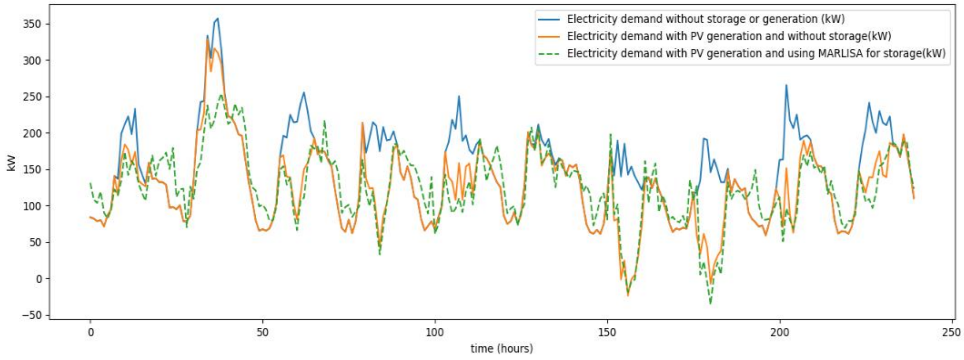

Figure 4.26 – Summer operation by MARLISA (last year)

4.5.1.1.3 Winter operation

First year

Figure (4.27) depicts a simulation of ten days of winter using MARLISA in safe exploration mode, during the first year of simulation.

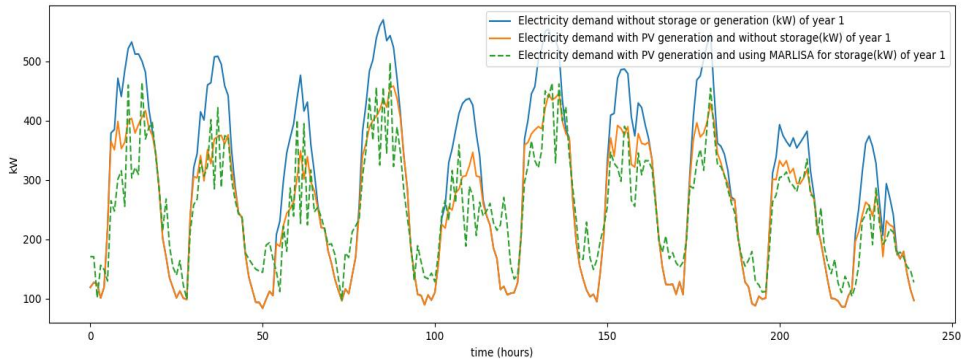

Figure 4.27 – Winter operation by MARLISA (first year)

Last year

A simulation process for ten days of winter utilizing MARLISA in safe exploration mode. The simulation displays the last year. In this situation, the agent will attempt to modify its learning because the consumption will be different than in the summer. As seen in the figure (4.28).

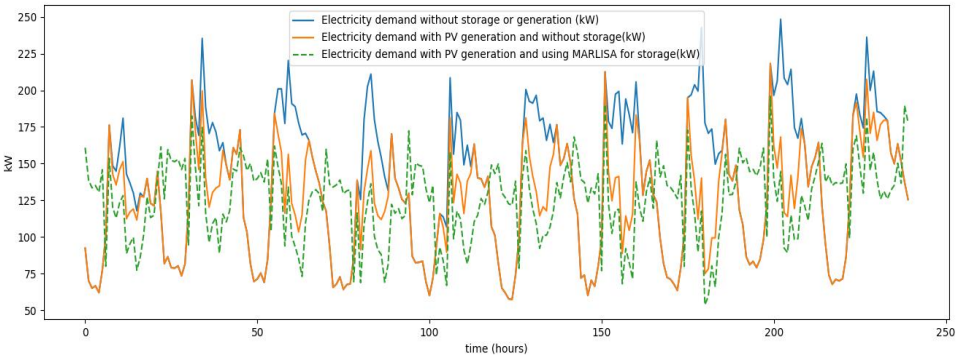

Figure 4.28 – Winter operation by MARLISA (last year)

4.5.1.1.4 Heating & cooling devices

The results in this section demonstrate the operation of the heat pump as a cooling and heating device over a five-day winter simulation period. It is used to evaluate the control algorithm's functionality. The simulation is limited to a single building, which is building number 5. Keeping in mind that building number 5 represents a residential building. The cooling demand is depicted in Figure (4.29), together with the battery's state of charge and the amount of heat supplied by the heat pump. The heating device, which is also a heat pump, is seen in Figure (4.30).

Cooling device

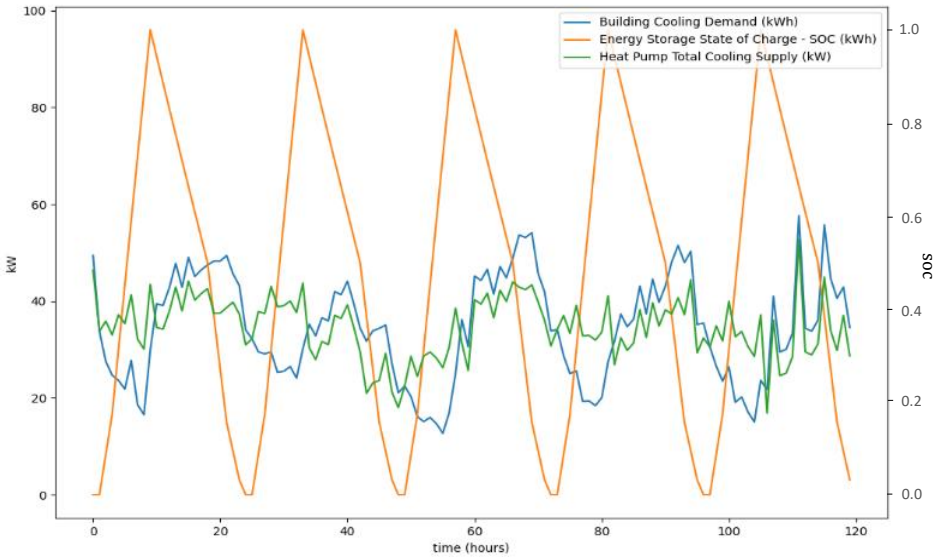

Figure 4.29 – Heat pump & Cooling demand

Heating device

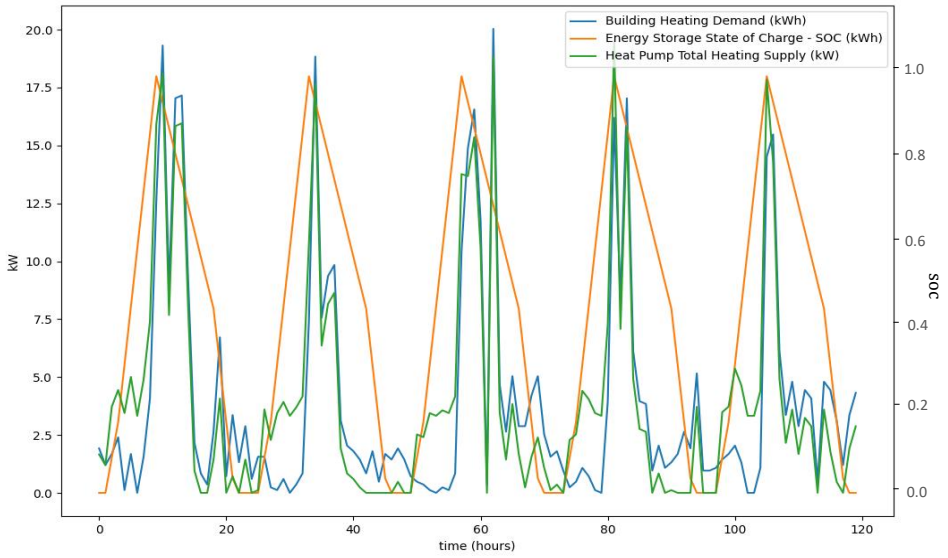

Figure 4.30 – Heat pump & heating demand

4.5.2 Analysis

When compared to the first year of operation, the results indicate an improvement in ramping and an increase in load factor (a decrease in 1-LF), as well as an increase in net electricity consumption and thus an increase in carbon emissions. Given that MARLISA is implemented using the SAC algorithm, the reason for this performance could be the reward function, which represents the default reward function provided by [133], the RBC performed much better.

4.5.2.1 Without safe exploration

In this part the safe exploration mode of MARLISA algorithm was deactivated, this means that the action selection is going to be performed by a random action selection method and not by the RBC agent (see equation (31)). All objective functions are obtained, and their results are illustrated in table (4.5).

Table 4.5 – Objective function using MARLISA (without safe exploration)

Objective functions	Cost function	Cost function for the last year
Ramping	1.465	1.089
1-Load factor	1.131	1.082

Peak demand	1.152	1.042
Average daily peak	1.138	1.064
Net electricity consumption	1.013	1.008
Carbon emissions	1.022	1.017
Coordination score	1.229	1.071
Total	1.158	1.054
Simulation time (min)	173.068	173.068

4.5.2.1.1 Total operation

Figure (4.31) depicts a simulation process for a four-year interval, the agent is MARLISA, and the simulation depicts three distinct scenarios: total operation without using a storage control method in blue, total operation using only a photovoltaic panel in orange, and total operation using photovoltaic systems and an agent for controlling energy storages, which in this case is MARLISA, in green.

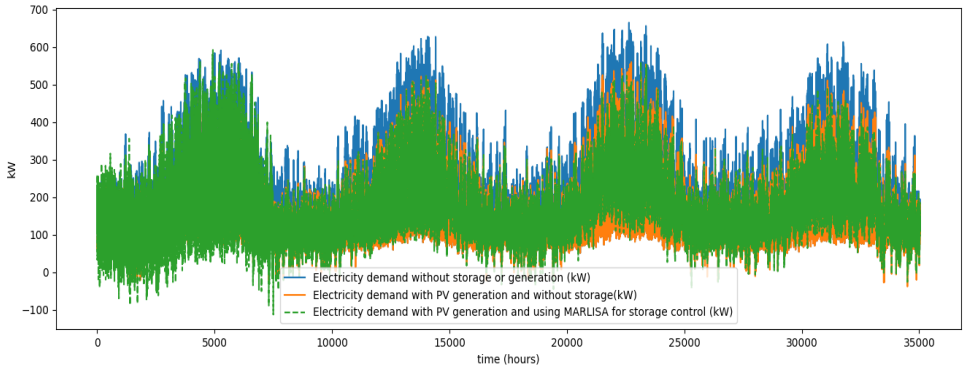

Figure 4.31 – Total simulation using MARLISA (without safe exploration)

Summer operation

First year

This section depicts ten days (240 hours) of summer operation during the first year, with the purpose of evaluating the agent's capacity to alter the control process. As seen in the image (4.32), the maximum peak for the typical consumption equal to 600 kWh in the last day of the interval.

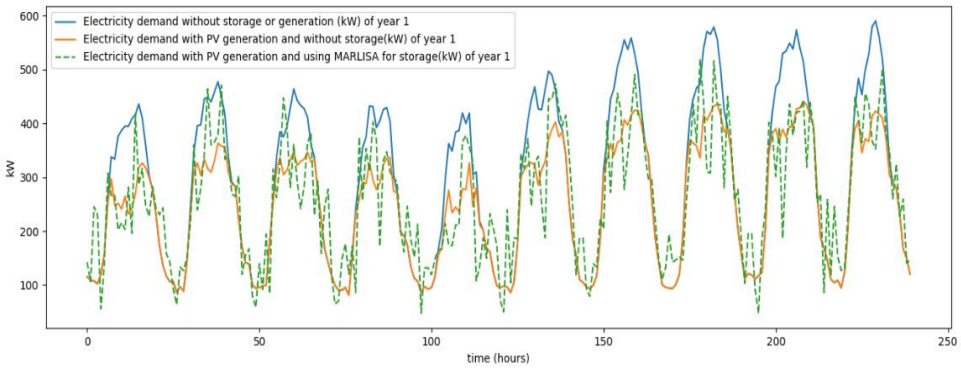

Figure 4.32 – Summer operation by the MARLISA (first year)

Last year

This section depicts ten days (240 hours) of summer operation with the goal of evaluating the agent's capacity to alter the control process. As seen in the image (4.33), the figure represents summer operation for the last year, and it can be noticed that the typical peak demand equal to 350 kWh. Further explanation will be provided in analysis section.

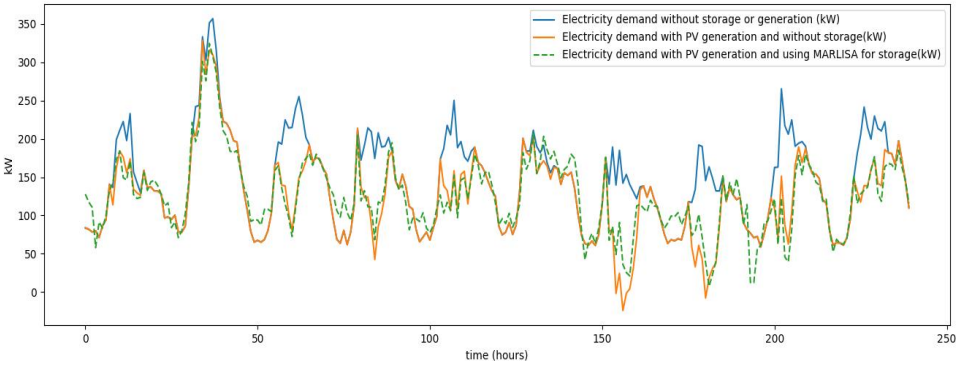

Figure 4.33 – Summer operation by the MARLISA (Last year)

Winter operation

A simulation for 5 days of winter is depicted in figure (4.34), the simulation shows the load shape while using MARLISA for in the first year of operation, and the last year in figure (4.35).

First year

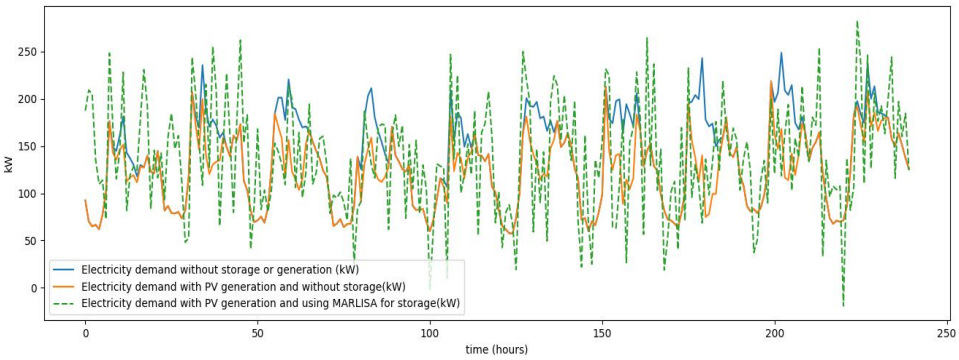

Figure 4.34 – Winter operation by the MARLISA (first year)

Last year

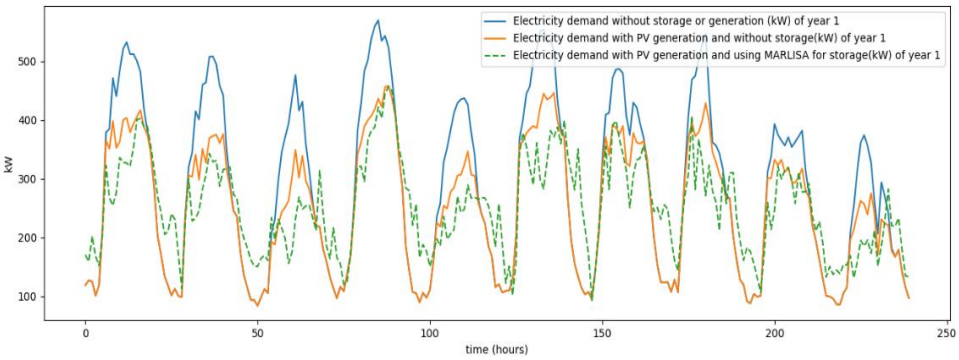

Figure 4.35 – Winter operation by the MARLISA (last year)

4.5.2.1.2 Heating & Cooling devices

The results in this section demonstrate the operation of the heat pump as a cooling and heating device over a five-day winter simulation period, which is used to evaluate the control algorithm's functionality. The simulation is limited to a single building, which is building number 5, keeping in mind that building number 5 represents a residential building. The cooling requirement is depicted in Figure (4.36), together with the battery's state of charge and the quantity of heat supplied by the heat pump. The heating device, which is also a heat pump, is seen in Figure (4.37).

Cooling device

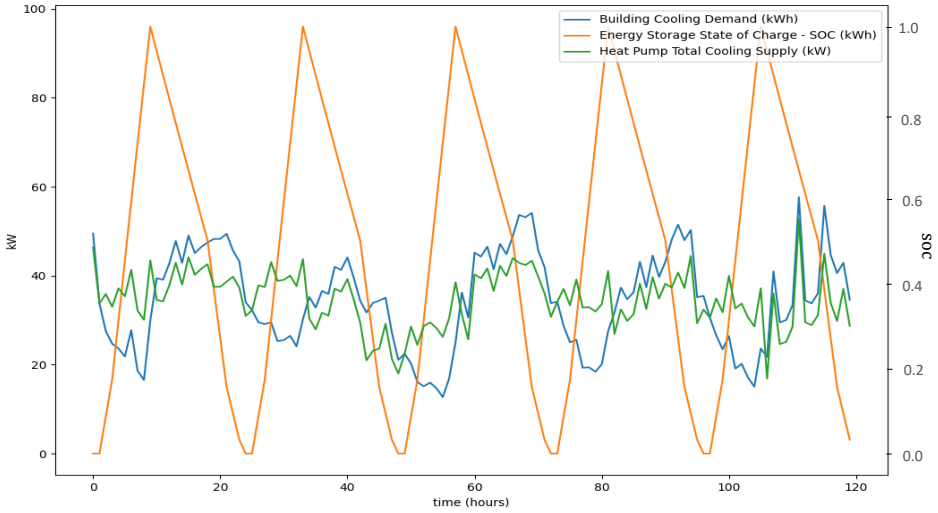

Figure 4.36 – Heat pump & cooling demand

Heating device

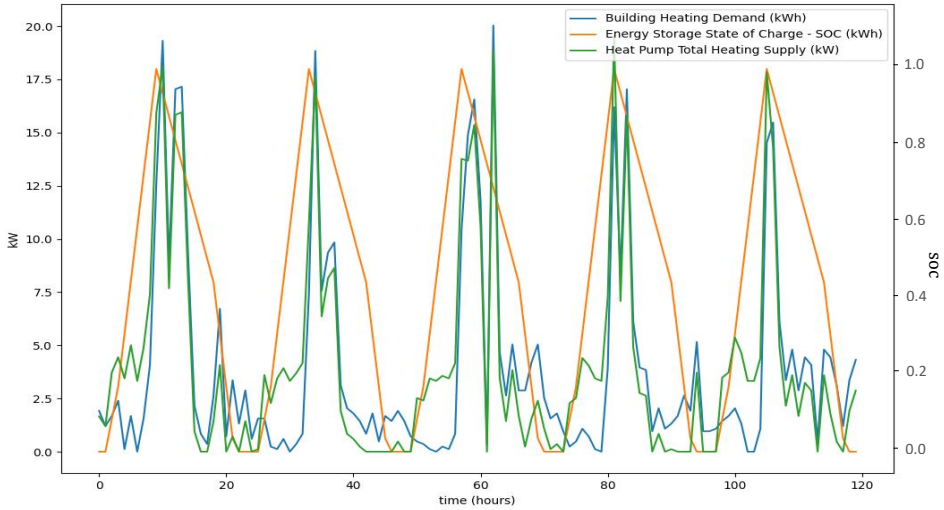

Figure 4.37 – Heat pump & Heating demand

4.5.2.1.3 Analysis

As illustrated, the results obtained with the MARLISA algorithm are divided into two groups: those obtained with and without the safe exploration mode. This is because, when attempting to select the appropriate action during the simulation, we either let the RBC choose the action for us or we choose the action using another method that multiplies a sample from action space by a coefficient defined to be equal to 0.5. As seen in the tables and simulations, the agent that does not employ the safe exploration approach outperforms the agent that does. In fact, the results indicate that the cost functions derived from the agent that uses the normal action selection method perform approximately identically to the RBC.

We can observe an increase in learning since agents (in both modes) are improving their performance from the last year. The simulations demonstrate that utilizing the MARLISA agent in safe exploration mode initially consumed less electricity than using merely a photovoltaic system. The second year saw a dramatic spike in electricity consumption, which is normal as the agent is still looking for its optimal zone, but the subsequent years saw a rapid drop and a large decrease in electricity consumption, as seen in figure (4.31), there was a steep decline and a significant decrease in the electricity consumption compared to using only PV panels. According to the summer operation figure (4.32) and (4.33), the agent was successful in reducing peak demand from 350 kWh to 300 kWh at 45 hours in the last year and performed exceptionally well when compared to the first year's performance. It is noticeable that the agent's performance is stable, in contrast to the first year, when it was unstable and inconsistence. The reason for the agent's poor performance during the first year could be because it is still seeking for the optimal zone in which to conduct multiple exploratory trials.

During the winter operation, the agent was unstable and went through an exploration period. Eventually, the agent achieved a larger demand than typical consumption, but when compared to the first year's performance, it is clear that the agent performed considerably better and eventually reduced demand to an optimal level. We can see that the agent began with a high consumption level in comparison to depending only on photovoltaic systems, but subsequently had a massive decline in subsequent years. In terms of coordination between the multiple agents, utilizing MARLISA without using the safe exploration mode performs far better than the other option. Nonetheless, it is critical to note that both modes result in the agent converging to the optimal solution.

5 Conclusion

In recent years, the energy sector has undergone a drastic change, with a significant push toward greener sources of energy, increased adoption of distributed energy sources, and electrification. However, because the current structure of the market and its investment templates are oriented toward fossil fuels. This has created uncertainty in investment, resulting in a lack of energy efficiency and sustainability. As well as environmental impacts associated with conventional fossil fuels generation units. As well as the intermittent nature of RES. A new supply method has emerged that focuses on limiting system imbalances through the use of demand-side measures. This method of operation is referred to as demand side management (DSM). One of the DSM programs is referred to as demand response (DR), which is viewed as a tool for increasing the system's flexibility and optimizing customer energy consumption. For this reason, DR makes use of a new market entity called a demand response aggregator (DRA). It is responsible for analysing and managing participants' load profiles in order to negotiate with other market participants in an organized market, to sell, purchase, and acquire their services.

The purpose of this work is to present an optimization procedure for load profiles for a small urban community. The community consists of nine buildings, five of which are residential, and the remaining are small commercials. This will help in the process of DR aggregation and will improve energy efficiency. A RL-based framework was chosen as a tool for load shaping and management. RL is a significant branch of machine learning. It is based on an intelligent agent interacting with an environment through an action. This interaction results in a reward from the environment. The objective is to increase the total number of rewards over time. Which is accomplished through numerous trial and error procedures. This will result in an agent that knows which action to do in order to acquire the most rewards. It has gained a lot of interest in recent years, especially when combined with deep neural networks, due to its extremely high accuracy and precision.

The framework is divided into two major components: the environment, called CityLearn, and the agent (or controller), which is the RL algorithm that will be used to control and shape the load. In this study, there are two types of agents: a decentralized agent, and a single agent which is also known as centralized agent. We used three different algorithms in this work. The first is called multi-agent reinforcement learning with iterative sequential action selection (abbreviated MARLISA), which is an algorithm that can be used for both decentralized and centralized agents. However, the original implementation of MARLISA requires an alternative reward function implementation or a modification in the action selection method. Because the default implementation of MARLISA does not permit the usage of a centralized agent without modification. The study concentrated only on employing a decentralized agent while using MARLISA. The second method

is a direct implementation of SAC algorithm, which is a deep RL algorithm, for both centralized and decentralized agents. Keeping in mind that the centralized agent implementation is based on an implementation from [150] [2]. Third is an RBC that represents an agent that the author [133] built and tuned to represent the baseline results for comparative reasons.

The results indicate that utilizing a reinforcement learning agent to control storage systems can be beneficial. All agents satisfied the electricity demand, and it is seen that the agent often begins with really poor results in the first year, but gradually converges to optimal results. Although the MARLISA algorithm from [133] produced the greatest results when used as a decentralized agent, no algorithm outperformed the baseline results in terms of overall performance. This is because, as previously stated, an alternative reward function needs to be developed, but ultimately, electricity consumption is reduced through control actions on energy storages. Demonstrating that RL can be depended upon for optimization problems and load shaping for DR. For adapting and reproducing these results various debugging procedures where performed, because it is critical to understand each and every element in the environment and agent in order to use it in the main file.

Although the work is based on RL, it makes use of regression methods from the ML branch of supervised learning to provide a good estimate of the results and the PCA algorithm from the unsupervised learning branch of ML to compress and reduce the dimensions of the data. It is critical to mention that using the centralized agent for SAC requires modifications to the environment code, by including the battery state and action in the observation and action space, which are excluded by default for the single agent. But even after including it, it won't be utilized eventually. Finally, we used a GPU resource provided by the Laboratory of Advance Computing (LAC) at the university of Coimbra, which is basically a platform known as NAVIGATOR+ to perform simulation procedures using CityLearn. The platform uses Bash command language and SLURM which is a queue management system, for more information about the simulation tool, please see Appendix D.

5.1 Future work

This part contains my suggestions, in order to use this research as a reference and a baseline for future developments, I suggest:

- A pricing system to simulate the dynamic pricing procedure, although a similar work was developed by [152] , but it would be nice to have different options of pricing system, to simulate the prices of the energy, from dynamic to static prices, and to find a way to simulate DR event days to see how the agent is going to behave, this can help to evaluate in which system the agent can perform well, and it can help the control agents to adapt to different pricing signals.

- Using different types of loads, because in this framework, all appliances and devices are limited to home appliances, using an EV would be a good choice, and to change the battery efficiency from curve type to a singular and scalar value.
- Redo the work with the electrical heater as the main heating device and compare results.
- Apply modification to the action selection method for MARLISA to be used for single agent, by finding a way to make the method accepts a single reward value rather than a list of 9 values, and this is what the CityLearn challenge is aiming for.
- Try to develop a reward function that takes the buildings correlation into account. Building's correlation instance is known is "Building_info", and it contains different information about the correlation factor between buildings, and it was illustrated in the methodology section.

Bibliography

[1] M. A. Umbark, S. K. Alghoul and E. I. Dekam, “Energy Consumption in Residential Buildings: Comparison between Three Different Building Styles,” Ideas Spread, 2020.

[2] J. R. Vázquez-Canteli, Z. Nagy, S. Dey and G. Henze, “CityLearn: Standardizing Research in Multi-Agent Reinforcement Learning for Demand Response and Urban Energy Management,” Cornell University, 2020.

[3] “Architecture2030,” Aug., 2019. Accessed on: Jan. 17, 2022 [Online]. Available: https://architecture2030.org/why-thebuildingsector/#:~:text=Buildings%20generate%20nearly%2040%25%20of,for%20an%20additional%2011%25%20annually.

[4] A. Conteh, M. E. Lotfy, K. M. Kipngetich, T. Senjyu, P. Mandal and S. Chakraborty, “An economic analysis of demand side management considering interruptible load and renewable energy integration: A case study of Freetown Sierra Leone,” MDPI, 2019.

[5] S. T. Tzeiranaki, P. Bertoldi, F. Diluiso, L. Castellazzi, M. Economidou, N. Labanca, T. R. Serrenho and P. Zangheri, “Analysis of the EU residential energy consumption: Trends and determinants,” MDPI AG, 2019.

[6] G. Trenev and P. Bertoldi, “Energy Efficiency in Domestic Appliances and Lighting,” in Proceedings of the 7 th International Conference EEDAL'2013, Joint research center, 2013.

[7] European commission, "Stepping up Europe’s 2030 climate ambition. The 2030 Climate target plan". Brussels, 2020. Accessed on: Oct. 20, 2021. [Online]. Available: https://eur - lex.europa.eu/legal- content/EN/TXT/PDF/?uri=CELEX :52020DC 0562&from=EN

[8] L. Paulo and S. Federico, “Demand Response for residential buildings: Case studies and DR program design by the RESPOND project,” in Conference: 4th Annual APEEN ConferenceAt: Covilhã - Portugal, 2019.

[9] S. Bahrami and A. Sheikhi, “From Demand Response in Smart Grid Toward Integrated Demand Response in Smart Energy Hub,” Institute of Electrical and Electronics Engineers Inc., 2016.

[10] D. York and M. Kushler, “Exploring the Relationship Between Demand Response and Energy Efficiency: A Review of Experience and Discussion of Key Issues,” American Council for an Energy Efficient Economy, 2005.

[11] M. S. Peter, “Energy Economics,” 1st ed Routledge, 2017.

[12] “DeepMind AI Reduces Google Data Centre Cooling Bill by 40%,” Google DeepM ind, Jul. 20, 2016. Accessed on: Jan. 20, 2022. [Online].Available: https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40.

[13] W. Ko, H. Vettikalladi, S. H. Song and H. J. Choi, “Implementation of a demand-side management solution for South Korea's demand response program,” MDPI AG, 2020.

[14] A. F. Meyabadi and M. H. Deihimi, “A review of demand-side management: Reconsidering theoretical framework,” Elsevier Ltd, 2017.

[15] I. Apolinário, N. Felizardo, G. A. Leite, P. Oliveira, A. Trindade and P. Verdelho, “Criteria for the Assessment of Demand Side Management Measures in the Context of Electricity Sector Regulation,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2006.

[16] D. Research and J. Ponoćko, “Springer Theses Recognizing Outstanding Ph Data Analytics-Based Demand Profiling and Advanced Demand Side Management for Flexible Operation of Sustainable Power Networks” Springer, 2020.

[17] C. W. Gellings, “Evolving practice of demand-side management,” Springer Heidelberg, 2017.

[18] “Overview of Greenhouse gases,” Environment Protection Agency government (EPA), Accessed on: Oct. 20, 2021 [Online]. https://www.epa.gov/ ghgemissions/ov erview-greenhouse-gases.

[19] “International emissions,” Center for Climate and Energy Solutions. Accessed on: Jan. 20, 2022. [Online]. Available: https://www.c2es.org/content/international-emissions/.

[20] J. Aghaei and M. I. Alizadeh, “Demand response in smart electricity grids equipped with renewable energy sources: A review,” Renewable and Sustainable Energy Reviews, Elsevier, vol. 18(C), pages 64-72. 2013.

[21] Ferran Torrent-Fontbona, “Optimization methods meet the smart grid,” Ph.D. dissertation, Universitat de Girona, Girona, 2014.

[22] P. Warren, “Electricity demand-side management: Best practice programmes for the UK,” WITPress, 2013.

[23] S. Nojavan and K. Zare, “Demand response application in smart grids: Concepts and planning issues - volume 1, ” Springer International Publishing, 2020, pp. 1-282.,2020.

[24] J. Lee, S. Yoo, J. Kim, D. Song and H. Jeong, “Improvements to the customer baseline load (CBL) using standard energy consumption considering energy efficiency and demand response,” Elsevier Ltd, 2018.

[25] “Energy efficiency and demand side management,” International energy agency (IEA), Accessed on: Oct. 1, 2021. [Online]. Available: https://www.iea.org/policies/578-energy-efficiency-and-demand-side-management-eedsm-programme.

[26] H. J. Jabir, J. Teh, D. Ishak and H. Abunima, “Impact of demand-side management on the reliability of generation systems,” MDPI AG, 2018.

[27] H. Saboori, M. Mohammadi and R. Taghe, “Virtual power plant (VPP), definition, concept, components and types,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2011.

[28] G. Karmiris and T. Tengnér, “Peak shaving control method for energy storage,” Sandia National Laboratories, 2014.

[29] M. H. Albadi and E. F. El-Saadany, “A summary of demand response in electricity markets,” Electric Power Systems Research, ScienceDirect, 2008.

[30] “Benefits of Demand Response in electricity markets and recommendations,” U.S. Department of Energy, 2006.

[31] International Energy Agency. and Organisation for Economic Co-operation and Development., “The Power to choose : Demand response in liberalised electricity markets,” OECD/IEA, 2003, p. 151.

[32] Y. T. Tan and D. Kirschen, “Classification of control for Demand Side Participation,” University of Manchester, 2007.

[33] “Application Note: Load Curtailment / Demand Response,” Obvius. Accessed on: Oct. 6, 2021. [Online]. Available: http://www.obvius.com/sites/obvius.com/files/AN_Demand_response.pdf

[34] P. Tarasak, C. C. Chai, Y. S. Kwok and S. W. Oh, “Demand bidding program and its application in hotel energy management,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2014.

[35] Ernest N. Morial Convention Center, “Smart solutions for a changing world” IEEE PES Transmission & Distribution Conference & Exposition, Apri I, 19 to April, 22, 2010, New Orleans, Louisiana.,2010.

[36] A. S. M. Khan, R. A. Verzijlbergh, O. C. Sakinci and L. J. De Vries, “How do demand response and electrical energy storage affect (the need for) a capacity market?,” Elsevier Ltd, 2018.

[37] K. Oureilidis, K. N. Malamaki, K. Gallos, A. Tsitsimelis, C. Dikaiakos, S. Gkavanoudis, M. Cvetkovic, J. M. Mauricio, J. M. M. Ortega, J. L. M. Ramos, G. Papaioannou and D, “Ancillary services market design in distribution networks: Review and identification of barriers,” MDPI AG, 2020.

[38] V. Subramanian, T. K. Das, C. Kwon and A. Gosavi, “A Data-Driven Methodology for Dynamic Pricing and Demand Response in Electric Power Networks” Electric Power Systems Research, Institute of Electrical and Electronics Engineers Inc. (IEEE), 2019.

[39] “Understanding Time of use rates,” Energysage. Accessed on: Jan. 6, 2022. [Online]. Available: https://news.energysage.com/understanding-time-of-use-rates/.

[40] Z. Wang and F. Li, “Critical peak pricing tariff design for mass consumers in Great Britain,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2011.

[41] J. Yusuf, A. S. Hasan and S. Ula, “Impacts Analysis Field Implementation of Plug-in Electric Vehicles Participation in Demand Response and Critical Peak Pricing for Commercial Buildings,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2021.

[42] A. Faruqui, D. Harris and R. Hledik, “Unlocking the €53 billion savings from smart meters in the EU: How increasing the adoption of dynamic tariffs could make or break the EU's smart grid investment,” Elsevier Ltd, 2010.

[43] A. Faruqui, A. Hajos, R. M. Hledik and S. A. Newell, “Fostering economic demand response in the Midwest ISO,” Elsevier Ltd, 2010.

[44] A. Star, L. Kotewa and M. Isaacson, “Real-Time Pricing Is the Real Deal: An Analysis of the Energy Impacts of Residential Real-Time Pricing,” ACEEE Summer Study on Energy Efficiency in Buildings, 2006.

[45] J. Zethmayr, D. Kolata “The costs and benefits of real-time pricing: An empirical investigation into consumer bills using hourly energy data and prices,” The Electricity Journal, 2017

[46] A. Faruqui and S. George, “Quantifying customer response to dynamic pricing,” Elsevier Inc. 2005.

[47] A. Faruqui, R. Hledik and J. Tsoukalis, “The power of dynamic pricing,” The Electricity Journal, ScienceDirect, 2009.

[48] P. H. Stephen and E. T. Mansur, “Is Real-time pricing green?,” National bureau of economic research, Cambridge, 2007.

[49] J. Lazar, L. Schwartz and R. Allen, “Pricing Do's and Don'ts: Designing Retail Rates As if Efficiency Counts Principal authors,” The Regulatory Assistance Project (RAP), 2011.

[50] North American Reliability Corporation (NERC), “Polar Vortex Review,” (NERC), Atalanta, GA, USA, 2014. Accessed on: Oct. 6, 2021. [Online]. Available: https://www.nerc.com/pa/rrm/January%202014%20Polar%20Vortex%20Review/Polar_Vortex_Review_29_Sept_2014_Final.pdf

[51] H. Zareipour, K. Bhattacharya and C. A. Cañizares, “Electricity market price volatility: The case of Ontario,” Energy Policy, Elsevier, 2007.

[52] P. A. De and E. G. Y. Rt Mentofen, “Demand Response and Advanced Metering 2020 Assessment of Staa Report Federal Energy Regulatory Commission December,” Federal Energy Regulatory Commission, 2020.

[53] M. H. Albadi, . Mohammed, E. F. El-Saadany, “Demand Response in Electricity Market: An Overview” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2007.

[54] P. V. Vassilopoulos “Models for the Identification of Market Power in Wholesale Electricity Markets,” University Paris IX – Dauphine, 2003.

[55] “Customer Engagement,” Michigan government, Michigan Public Service Commission. Accessed on: Jan. 15, 2022 [Online]. Available: https:// www.michigan .gov/mpsc/0,9535,7-395-93307_93312_93593_95590_95594---,00.html.

[56] J. A. Carr, J. Carlos Balda and H. A. Mantooth, “A Survey of Systems to Integrate Distributed Energy Resources and Energy Storage on the Utility Grid” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2008.

[57] N. Oconnell, P. Pinson, H. Madsen and M. Omalley, “Benefits and challenges of electrical demand response: A critical review,” Elsevier Ltd, 2014.

[58] T. Mount and P. Leader, “Coupling Wind Generation with Controllable Load and Storage: A Time-Series Application of the SuperOPF Final Project Report” The Power Systems Engineering Research Center (PSERC), 2012.

[59] C. De Jonghe, B. F. Hobbs and R. Belmans, “Optimal generation mix with short-term demand response and wind penetration,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2012.

[60] B. M. Sanandaji, T. L. Vincent and K. Poolla, “Ramping Rate Flexibility of Residential HVAC Loads” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2015.

[61] B. J. Kirby, “Load response fundamentally matches power system reliability requirements,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2007.

[62] I. Hiskens and D. Callaway, “Achieving Controllability of Plug-in Electric Vehicles,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2011.

[63] C. L. Su and D. Kirschen, “Quantifying the effect of demand response on electricity markets,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2009.

[64] U. S. Department of energy, “Benefits of Demand Response in electricity markets and recommendations for achieving them, ” Report to the US Congress pursuant to section 1252 of the energy policy act of 2005, 2005.

[65] S. Burger, J. P. Chaves-Ávila, C. Batlle and I. J. Pérez-Arriaga, “A review of the value of aggregators in electricity systems,” Elsevier Ltd, 2017.

[66] The European Parliament “Common rules for the internal market for electricity and amending Directive 2012/ 27/ EU,” Official Journal of the European Union, 2019.

[67] A. Forouli, E. A. Bakirtzis, G. Papazoglou, K. Oureilidis, V. Gkountis, L. Candido, E. D. Ferrer and P. Biskas, “Assessment of demand side flexibility in european electricity markets: A country level review,” MDPI AG, 2021.

[68] J. Martin, “Distributed vs. centralized electricity generation: are we witnessing a change of paradigm?” HEC Paris, 2009.

[69] R. Bray and B. Woodman, “Barriers to Independent Aggregators in Europe, ” University of Exeter, 2019.

[70] A. Forouli, E. A. Bakirtzis, G. Papazoglou, K. Oureilidis, V. Gkountis, L. Candido, E. D. Ferrer and P. Biskas, “Assessment of demand side flexibility in european electricity markets: A country level review,” MDPI AG, 2021.

[71] X. Lu, K. Li, H. Xu, F. Wang, Z. Zhou and Y. Zhang, “Fundamentals and business model for resource aggregator of demand response in electricity markets,” Elsevier Ltd, 2020.

[72] Smart Grid Task Force (SGTF), “Regulatory Recommendations for the Deployment of Flexibility,” SGTF-EG3 Report, 2015.

[73] European Commission, “Directive of the European Parliament and the council,” EU Commission, Brussels, 2016.

[74] Eurelectric, “European Commission’s legislative proposal on common rules for the internal market in electricity,” A EURELECTRIC position paper, 2017.

[75] P. Baker, “Benefiting Customers While Compensating Suppliers: Getting Supplier Compensation Right,” The Regulatory Assistance Project, 2016.

[76] I. e. w. E. First, “Participation of Demand Response in French wholesale electricity market,” RTE, the French Transmission System Operator, 2014.

[77] H. Haeri, K. Horkitz, H. Lee, J. Wang, T. Hardman, H. Ratcliffe, M. Izawa, J. Brant, J. Eckstein, N. Preston, L. Garth “Assessment of Barriers to Demand Response in the Northwest's Public Power Sector,” The Cadmus Group LLC, 2018.

[78] N. Habib, Hands-on Q-learning with Python : practical Q-learning with OpenAI Gym, Keras, and TensorFlow, Packt Publishing, 2019.

[79] "Expectation Values and the Bellman Equation," P. Tabor. Accessed on: Jan. 15, 2022 [Online]. Available: https://www.udemy.com/course/actor-critic-methods-from-paper-to-code-with-pytorch/.

[80] J. R. Vázquez-Canteli and Z. Nagy, "Reinforcement learning for demand response: A review of algorithms and modeling techniques," Elsevier Ltd, 2019.

[81] T. M. Hansen, E. K. Chong, S. Suryanarayanan, A. A. Maciejewski and H. J. Siegel, "A partially observable markov decision process approach to residential home energy management," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2018.

[82] F. Doshi-Velez, "The Infinite Partially Observable Markov Decision Process" Conference on Neural Information Processing Systems, 2009.

[83] J. Fürnkranz, T. Scheffer and M. Spiliopoulou, "Lecture Notes in Artificial Intelligence 4212 Subseries of Lecture Notes in Computer Science," Springer Book series, 2006.

[84] V. Pong, S. Gu, M. Dalal and S. Levine, "Temporal Difference Models: Model-Free Deep RL for Model-Based Control," Cornell University, 2018.

[85] F. Yi, W. Fu and H. Liang, "Association for Information Systems AIS Electronic Library (AISeL) Model-based reinforcement learning: A survey" Association for Information Systems AIS Electronic Library (AISeL), 2019.

[86] T. M. Moerland, J. Broekens and C. M. Jonker, "Learning Multimodal Transition Dynamics for Model-Based Reinforcement Learning" Cornell University, 2017.

[87] M. Parvania, M. Fotuhi-Firuzabad and M. Shahidehpour, "Optimal demand response aggregation in wholesale electricity markets," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2013.

[88] S. Mhanna, G. Verbič and A. C. Chapman, "A Faithful Distributed Mechanism for Demand Response Aggregation," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2016.

[89] W. L. Schram, T. Alskaif, I. Lampropoulos, S. Henein and W. G. Van Sark, "On the Trade-Off between Environmental and Economic Objectives in Community Energy Storage Operational Optimization," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2020.

[90] A. T. Al-Awami, N. A. Amleh and A. M. Muqbel, "Optimal demand response bidding and pricing mechanism with fuzzy optimization: Application for a virtual power plant," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2017.

[91] D. T. Nguyen and L. B. Le, "Risk-constrained profit maximization for microgrid aggregators with demand response," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2015.

[92] H. Longbo, W. Jean and R. Kannan, "Optimal Demand Response with Energy Storage Management," IEEE, 2012.

[93] D. T. nguyen, M. Negnevitsky and M. De Groot, "Walrasian market clearing for demand response exchange," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2012.

[94] M. Gonzalez Vaya and G. Andersson, "Optimal Bidding Strategy of a Plug-In Electric Vehicle Aggregator in Day-Ahead Electricity Markets under Uncertainty," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2015.

[95] R. Henriquez, G. Wenzel, D. E. Olivares and M. Negrete-Pincetic, "Participation of demand response aggregators in electricity markets: Optimal portfolio management," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2018.

[96] A. Taşcikaraoğlu, N. G. Paterakis, O. Erdinç and J. P. Catalão, "Combining the Flexibility From Shared Energy Storage Systems and DLC-Based Demand Response of HVAC Units

for Distribution System Operation Enhancement," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2019.

[97] W. Liu, Q. Wu, F. Wen and J. Ostergaard, "Day-ahead congestion management in distribution systems through household demand response and distribution congestion prices," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2014.

[98] R. García-Bertrand, "Sale prices setting tool for retailers," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2013.

[99] L. Zheng and L. Cai, "A distributed demand response control strategy using lyapunov optimization," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2014.

[100] Y. Shimomura, Y. Nemoto, F. Akasaka, R. Chiba and K. Kimita, "A method for designing customer-oriented demand response aggregation service," Elsevier Inc., 2014.

[101] G. Huangjie, J. Evan S, A. Rosemary E and A. G. Frye, "Demand Response of HVACs in Large Residential Communities Based on Experimental Developments," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2020.

[102] W. Ran, L. Yifan, W. Ping and N. Dusit, "Design of a V2G Aggregator to Optimize PHEV Charging and Frequency Regulation Control," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2013.

[103] Z. Chen, L. Wu and Y. Fu, "Real-time price-based demand response management for residential appliances via stochastic optimization and robust optimization," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2012.

[104] P. Samadi, H. Mohsenian-Rad, V. W. Wong and R. Schober, "Real-time pricing for demand response based on stochastic approximation," Institute of Electrical and Electronics Engineers Inc. (IEEE), 2014.

[105] N. Good, E. Karangelos, A. Navarro-Espinosa and P. Mancarella, “Optimization under Uncertainty of Thermal Storage-Based Flexible Demand Response with Quantification of Residential Users' Discomfort,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2015.

[106] H. Dagdougui, A. Ouammi and L. A. Dessaint, “Peak Load Reduction in a Smart Building Integrating Microgrid and V2B-Based Demand Response Scheme,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2019.

[107] A. Safdarian, M. Fotuhi-Firuzabad and M. Lehtonen, “A distributed algorithm for managing residential demand response in smart grids,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2014.

[108] W. Shi, N. Li, X. Xie, C. C. Chu and R. Gadh, “Optimal residential demand response in distribution networks,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2014.

[109] A. J. Conejo, J. M. Morales and L. Baringo, “Real-time demand response model,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2010.

[110] B. S. K. Patnam and N. M. Pindoriya, “Demand response in consumer-Centric electricity market: Mathematical models and optimization problems,” Elsevier Ltd, 2021.

[111] A. T. P. Imthias, M. S. Danish, A. A.-A. Essam and N. Malik, “A Simulated Annealing Algorithm for Demand,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2011.

[112] S. Nan, M. Zhou and G. Li, “Optimal residential community demand response scheduling in smart grid,” Elsevier Ltd, 2018.

[113] T. Terlouw, T. AlSkaif, C. Bauer and W. van Sark, “Multi-objective optimization of energy arbitrage in community energy storage systems using different battery technologies,” Elsevier Ltd, 2019.

[114] M. Rahmani-Andebili, A. Abdollahi and M. P. Moghaddam, “An Investigation of Implementing Emergency Demand Response Program (EDRP) in Unit Commitment Problem,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2011.

[115] L. P. Qian, Y. J. A. Zhang, J. Huang and Y. Wu, “Demand response management via real-time electricity price control in smart grids,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2013.

[116] D. O'neill, M. Levorato, A. Goldsmith and U. Mitra, “Residential Demand Response Using Reinforcement Learning” Stanford University, 2010.

[117] “Market and Consumer Energy Communities,” European commission. Accessed on: Jan. 15, 2022. [Online]. Available: https://ec.europa.eu/energy/topics/markets-and-consumers/energy-communities_en.

[118] E. O'Shaughnessy, J. Heeter, J. Gattaciecca, J. Sauer, K. Trumbull and E. Chen, “Empowered communities: The rise of community choice aggregation in the United States,” Elsevier Ltd, 2019.

[119] G. Michaud, “Deploying solar energy with community choice aggregation: A carbon fee model,” Elsevier Inc., 2018.

[120] S. F. Kennedy and B. Rosen, “The rise of community choice aggregation and its implications for California’s energy transition: A preliminary assessment,” SAGE Publications Inc., 2021.

[121] G. B. Huitema, A. v. d. Veen, V. Georgiadou, M. Vavallo and M. A. García, “Demand-Response Optimization in Buildings and Energy Communities, a Case in Value Stacking,” MDPI, 2020.

[122] V. Z. Gjorgievski, S. Cundeva and G. E. Georghiou, “Social arrangements, technical designs and impacts of energy communities: A review,” Elsevier Ltd, 2021.

[123] A. Abdulaal, R. Moghaddass and S. Asfour, “Two-stage discrete-continuous multi-objective load optimization: An industrial consumer utility approach to demand response,” Elsevier Ltd, 2017.

[124] T. Kerdphol, Y. Qudaih and Y. Mitani, “Optimum battery energy storage system using PSO considering dynamic demand response for microgrids,” Elsevier Ltd, 2016.

[125] A. A. Aarts, J. E. Anderson, C. J. Anderson, P. R. Attridge, A. Attwood, J. Axt, M. Babel, Š. Bahník, E. Baranski, M. Barnett-Cowan, E. Bartmess, J. Beer, R. Bell and H. Bentley, “Estimating the reproducibility of psychological science,” PubMed, 2015.

[126] L. Buşoniu, B. De Schutter and R. Babuška, “Decentralized reinforcement learning control of a robotic manipulator,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2006.

[127] M. Moradi, “A centralized reinforcement learning method for multi-agent job scheduling in Grid” Cornell University, 2016.

[128] M. C. Baechler, T. L. Gilbride, P. C. Cole, M. G. Hefty and K. Ruiz, “Guide to Determining Climate Regions by County,” Pacific Northwest National Laboratory, 2015.

[129] “Solar radiation Basics,” Solar Energy Technologies Office. Accessed on: Jan. 15, 2022. [Online]. Available: https://www.energy.gov/eere/solar/solar-radiation-basics.

[131] I. S. Ertesvåg, “Uncertainties in heat-pump coefficient of performance (COP) and exergy efficiency based on standardized testing,” Elsevier Ltd, 2011.

[132] “Pearson Product-Moment Correlation,” Statistics Laerd. [Online]. Accessed on: Jan. 15, 2022. Available: https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php.

[133] J. R. Vazquez-Canteli, G. Henze and Z. Nagy, “MARLISA: Multi-Agent Reinforcement Learning with Iterative Sequential Action Selection for Load Shaping of Grid-Interactive Connected Buildings,” Association for Computing Machinery, Inc, 2020.

[134] T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” Cornell University, 2018.

[135] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel and S. Levine, “Soft Actor-Critic Algorithms and Applications,” Cornell University, 2019.

[136] W. Fedus, P. Ramachandran, R. Agarwal, Y. Bengio, H. Larochelle, M. Rowland and W. Dabney, “Revisiting Fundamentals of Experience Replay,” Cornell, University, 2020.

[137] “Soft Actor Critic slides, ” O. Sigaud and T. Pierrot. Accessed on: Jan. 15. Available: http://pages.isir.upmc.fr/~sigaud/teach/sac.pdf.

[138] J. Xu, X. Sun, Z. Zhang, G. Zhao and J. Lin, “Understanding and Improving Layer Normalization,” Cornell University, 2019.

[139] J. Daniel and J. H. Martin, “Speech and Language Processing,” Stanford University, 2021.

[140] D. P. Kingma, T. Salimans and M. Welling, “Variational Dropout and the Local Reparameterization Trick” Cornell University, 2015.

[141] A. R. Jha, Mastering PyTorch, Birmingham-Mumbai, 2021.

[142] V. Könönen, “Asymmetric multiagent reinforcement learning,” Institute of Electrical and Electronics Engineers Inc. (IEEE), 2003.

[143] Y. V. Makarov, D. J. Hill and I. A. Hiskens, “Properties of quadratic equations and their application to power system analysis” Elsevier Ltd, 2000.

[144] “The System Advisor Model (SAM),” National Renewable Energy Laboratory (NREL). Accessed on: Jan. 16 [Online]. Available: https://sam.nrel.gov/.

[145] “Pecan Street,” Pecan Street Inc. Accessed on: Jan. 16. [Online]. Available: https://www.pecanstreet.org/dataport/.

[146] R. Hendron, “Building America Research Benchmark Definition: Updated December 19, 2008,” National Renewable Energy Laboratory (NREL), 2008.

[147] “Restock project,” National Renewable Energy Laboratory (NREL). Accessed on: Jan. 18. [Online]. Available: https://github.com/NREL/resstock.

[148] “Stable Baselines's Soft Actor Critic,” Stable Baselines. Accessed on: Jan. 16. [Online]. Available: https://stable-baselines.readthedocs.io/en/master/modules/sac.html.

[149] “CityLearn Github repository,” Intelligent Environments Laboratory. Accessed on: Jan. 16. [Online] Available: https://github.com/intelligent-environments-lab/CityLearn.

[150] F. Tolovski, “Advancing Renewable electricity Consumption with Reinforcement Learning” International Conference on Learning Representations, 2020.

[151] S. Maggiore and M. Benini, “Evaluation of the effects of a tariff change on the Italian residential customers subject to a mandatory time-of-use tariff” European Council for an Energy Efficient Economy, 2013.

[152] P. Rodilla and C. Batlle, “Electricity demand response tools: current status and outstanding issues of Energy Markets,” The European Energy Institute 2008.

[153] D. Schlegel, “Deep Machine Learning on GPUs,” Institute of Computer Engineering at Heidelberg University (ZITI), 2015.

[154] “The Laboratory for Advanced Computing at University of Coimbra (LAC),” University of Coimbra. Accessed on: Jan. 17. [Online]. Available: https://www.uc.pt/lca/ClusterResources/Navigator/general_info.

[155] “Slurm Quickstart,” Slurm. Accessed on: Jan. 18. [Online]. Available: https://slurm.schedmd.com/quickstart.html.

[156] Y. Rebours and D. Kirschen, “What is spinning reserve?,” University of Manchester, 2005.

[157] “Spinning Reserve,” Energy storage. Accessed on: Jan. 17. [Online]. Available: https://energystorage.org/spinning-reserve/.

[158] D. Pudjianto, C. Ramsay and G. Strbac, “Virtual power plant and system integration of distributed energy resources,” Institute of Electrical and Electronics Engineers Inc. (IEEE) ,2007.

[159] M. Ivas, M. Telenerg doo and S. cesta, “Probablistic risk assesment of island operation of grid connected multi-inverter power plant,” Taylor & Francis Online, 2014.

[160] “EnergyPlus,” EnergyPlus Inc. Accessed on: Jan. 17. [Online]. Available: https://energyplus.net.

[161] “Building models,” The U.S. Department of Energy (DOE). Accessed on: Jan. 17. [Online]. Available: https://www.energycodes.gov/prototype-building-models.

Glossary

Term	Definition
Bi-directional	Functioning in two directions
Buffer	A portion of a program's memory set aside for storing the data now being processed
Convergence of Algorithm	When as the iterations proceed the output gets closer and closer to a specific value (optimal value)
Deterministic	Descriptor for an argument or approach that simplifies causation to one or two components functioning directly or nearly so to cause outcomes
Entropy	Unpredictability and a lack of order; a descent into stochasticity and randomness
Gradient	A differential operator applied to a three-dimensional vector-valued function to yield a vector whose three components are the partial derivatives of the function with respect to its three variables
Intermittent	Uneven in nature; not continuous or steady
Neural network	A neural network is a sequence of algorithms that strives to detect underlying correlations in a piece of data through a method that mimics the way the human brain processes
Optimization	The act of making the best or the most effective of one's circumstances or resources
Polynomial	Multi-algebraic statement, specifically the sum of numerous parts with various powers of the same variable or variables
Scalar value	A single item, as opposed to a composite or collection
Triggering	Activating and causing a particular action, process, or situation

Appendices

Appendix A

Time of use rate (TOU) example in Europe

Using Italy as an example, the rate period is divided into two parts: from 8 a.m. to 19 a.m. represents the first-rate period (on-peak), and the remainder of the day represents the second-rate period (this is only on working days). Since 2010, this TOU rate has served as the default rate for residential customers [153]**.** However, TOU rate period can span multiple days. An example of this is the Tempo tariff in France, which has six distinct price levels based on the type of day. The day can be blue, which corresponds to the normal demand price (most days are blue), white, which corresponds to higher prices and increased demand for electricity. Finally, red, which corresponds to critical events days, which contain the highest prices. Each day is divided into two intervals: on-peak and off-peak, with on-peak referring to the period between 6:00 a.m. and 10:00 p.m. (6:00 a.m. to 22:00 p.m.) and off-peak referring to the rest of the day [154].

Appendix B

The figure below (14.1), illustrate CityLearn environment structure, and the main component, and the methods used for each section.

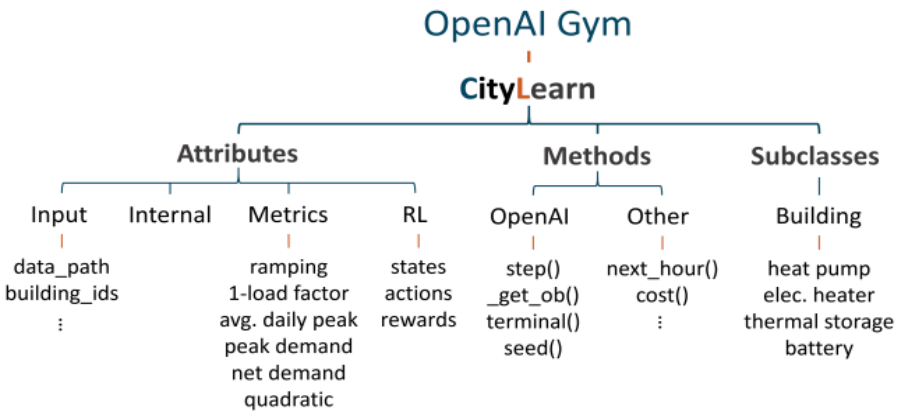

Figure – CityLearn component

Appendix C

This section will provide more information about the building electricity consumption and cooling and heating demands.

DWH demands

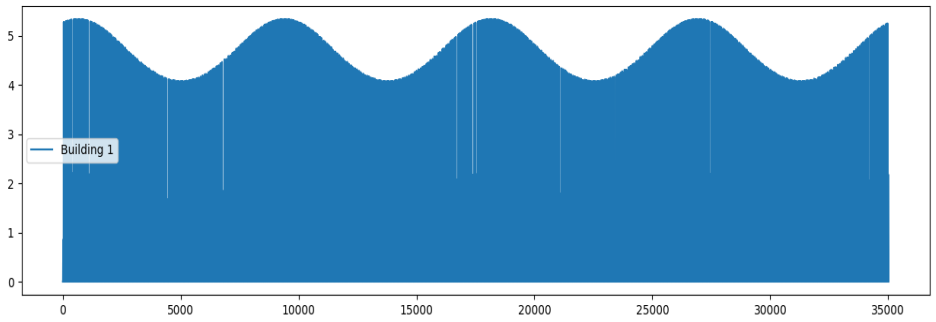

Figure Appendix C.1 – DHW demand for building 1

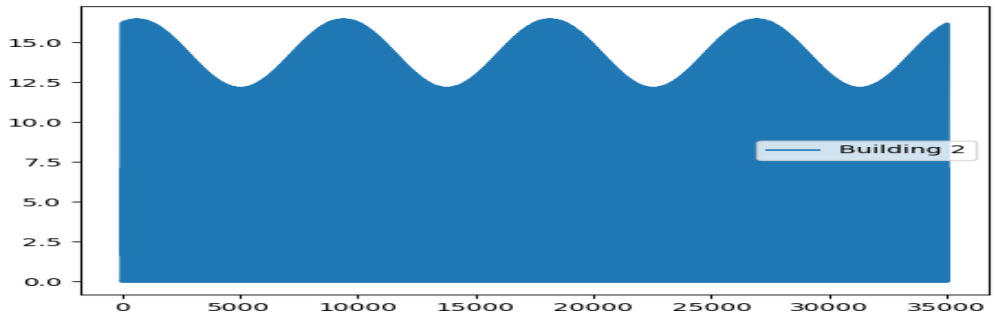

Figure Appendix C.2 – DHW demand for building 2

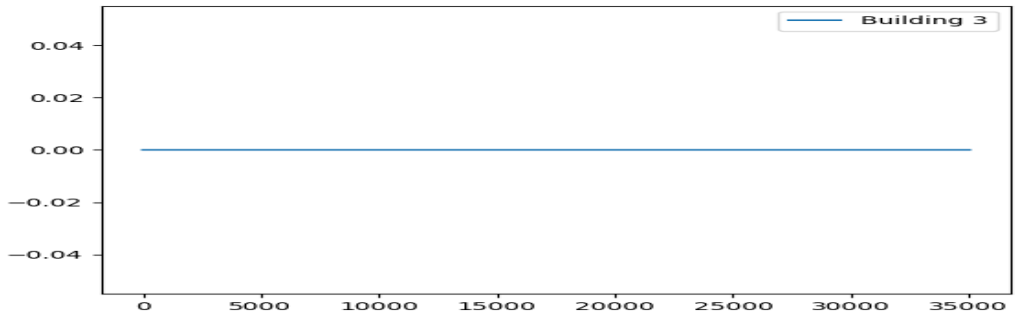

Figure Appendix C.3 – DHW demand for building 3

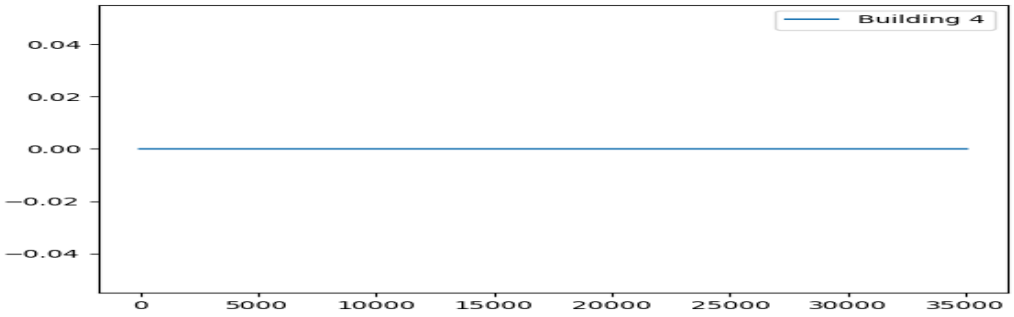

Figure Appendix C.4 – DHW demand for building 4

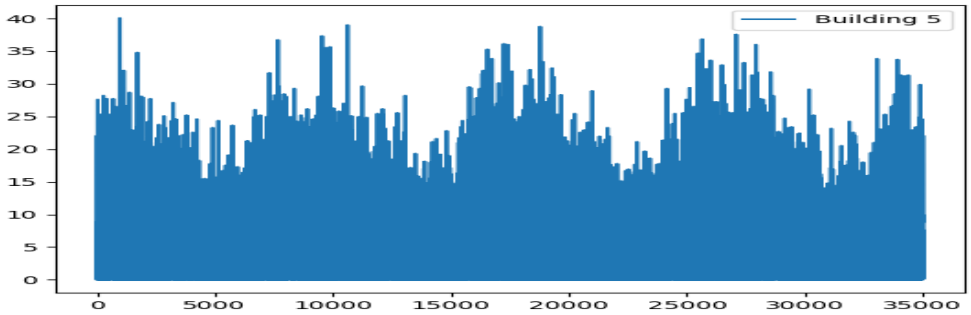

Figure Appendix C.5 – DHW demand for building 5

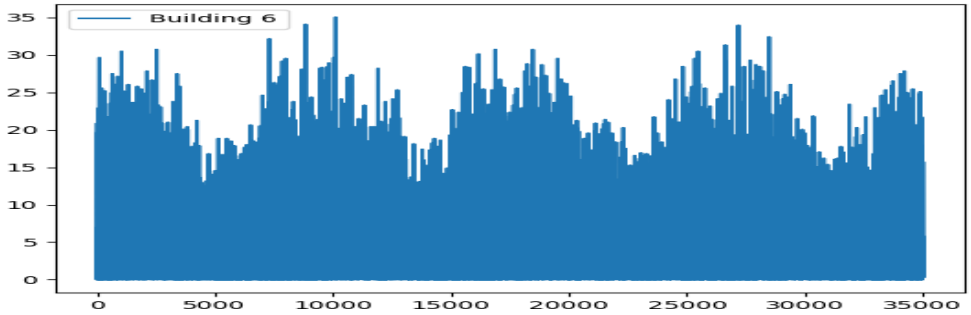

Figure Appendix C.6 – DHW demand for building 6

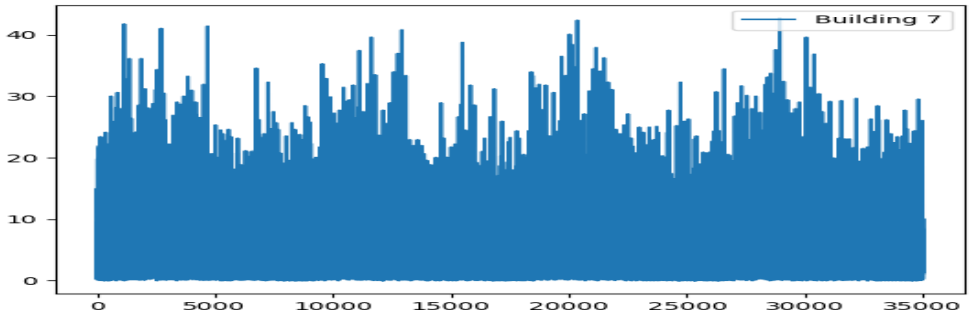

Figure Appendix C.7 – DHW demand for building 7

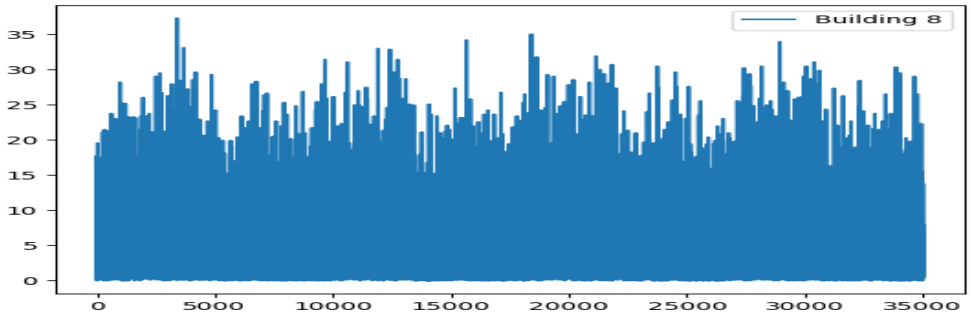

Figure Appendix C.8 – DHW demand for building 8

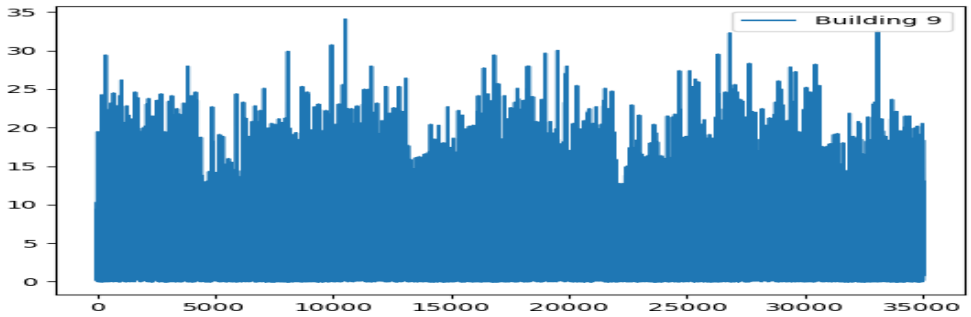

Figure Appendix C.9 – DHW demand for building 9

Cooling demand

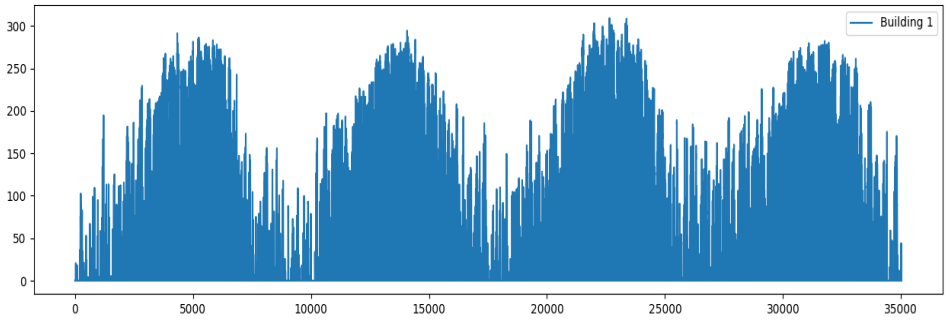

Figure Appendix C.10 – Cooling demand for building 1

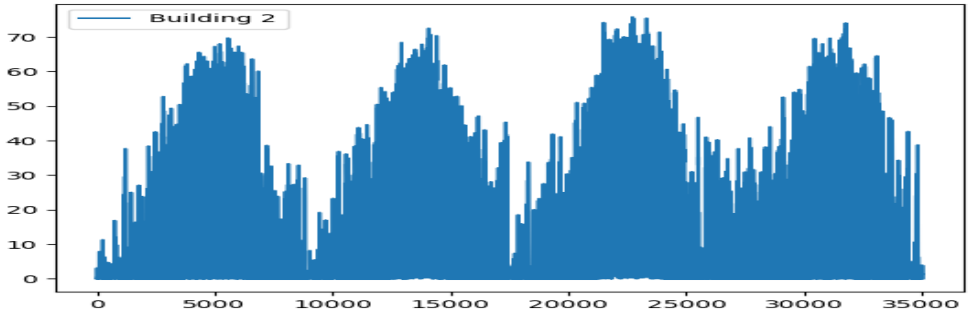

Figure Appendix C.11 – Cooling demand for building 2

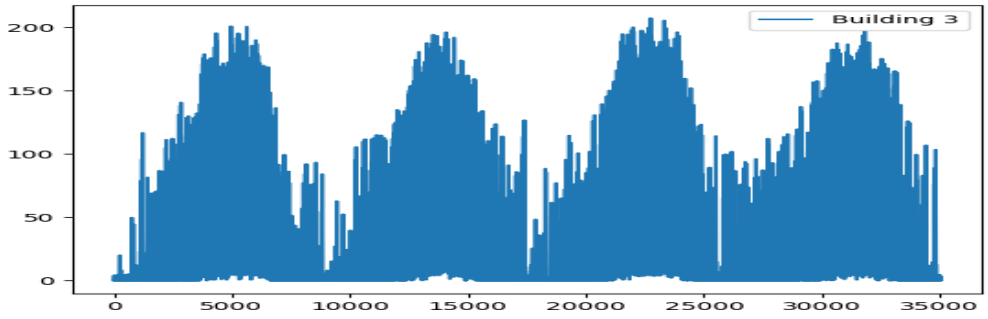

Figure Appendix C.12 – Cooling demand for building 3

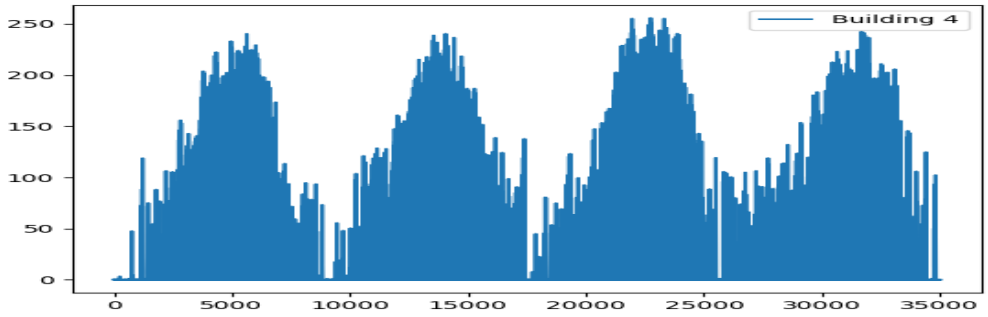

Figure Appendix C.13 – Cooling demand for building 4

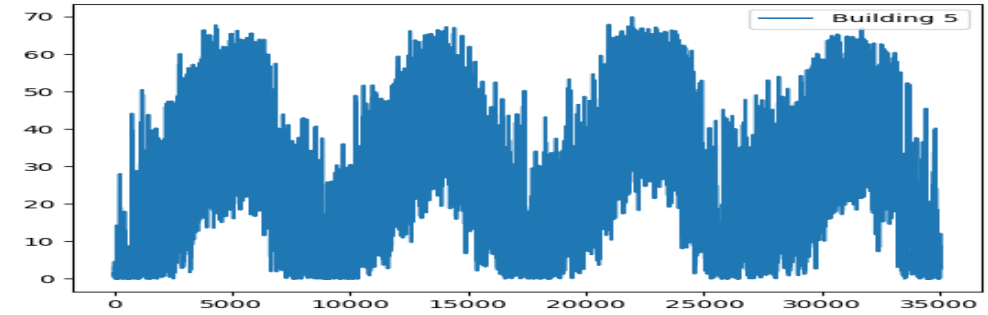

Figure Appendix C.14 – Cooling demand for building 5

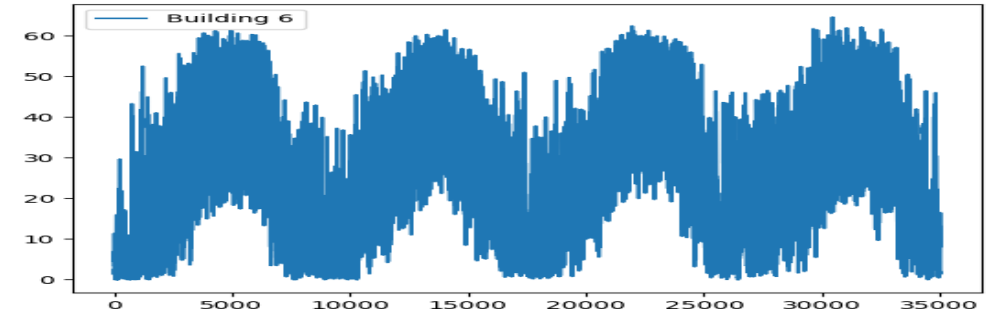

Figure Appendix C.15 – Cooling demand for building 6

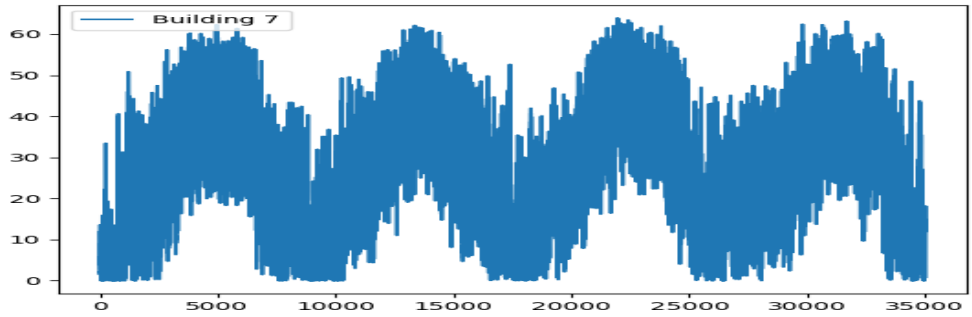

Figure Appendix C.16 – Cooling demand for building 7

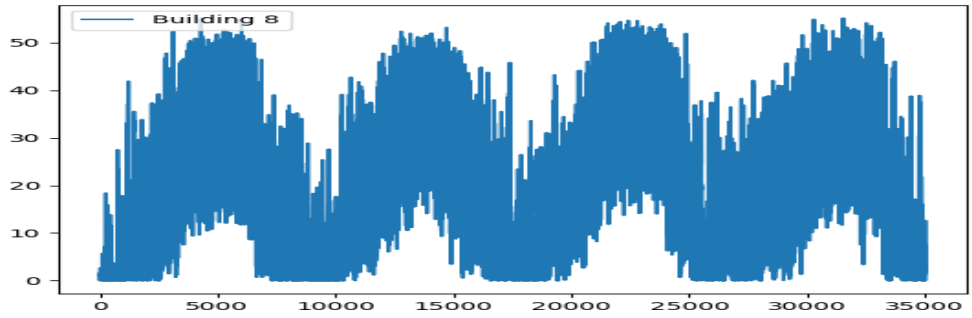

Figure Appendix C.17 – Cooling demand for building 8

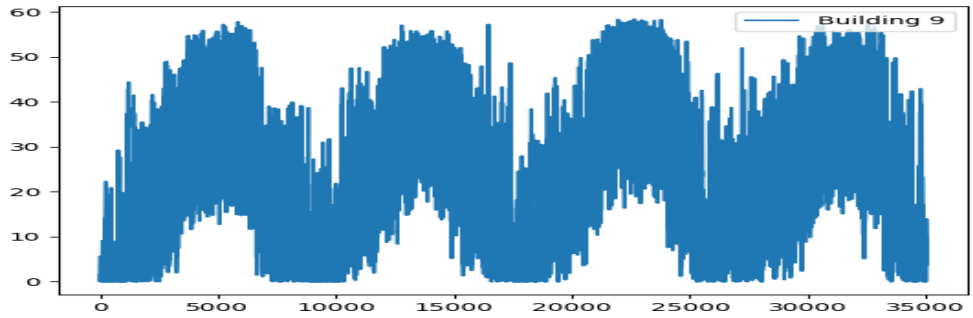

Figure Appendix C.18 – Cooling demand for building 9

Building's electrical equipment's demands

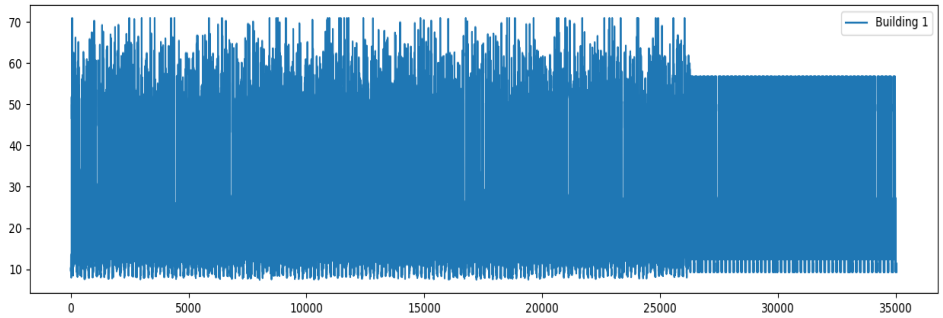

Figure Appendix C.19 – Electrical equipment demand for building 1

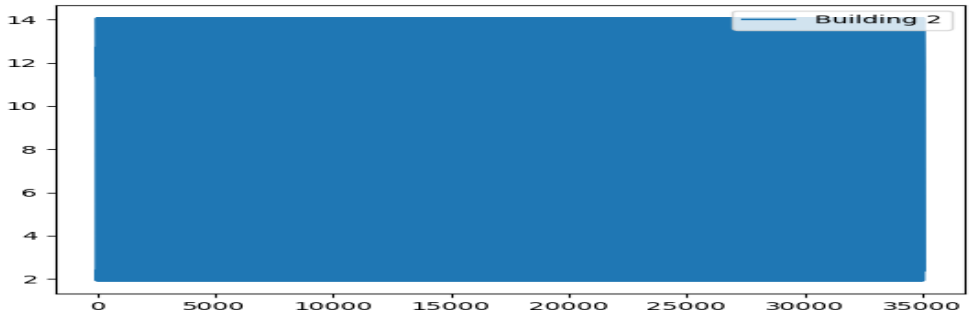

Figure Appendix C.20 – Electrical equipment demand for building 2

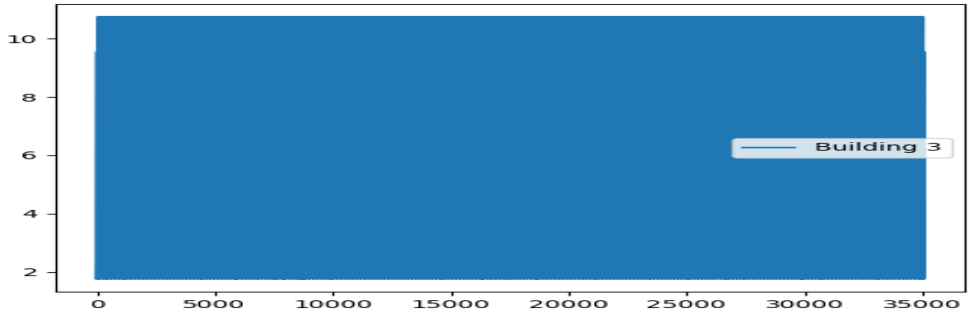

Figure Appendix C.21 – Electrical equipment demand for building 3

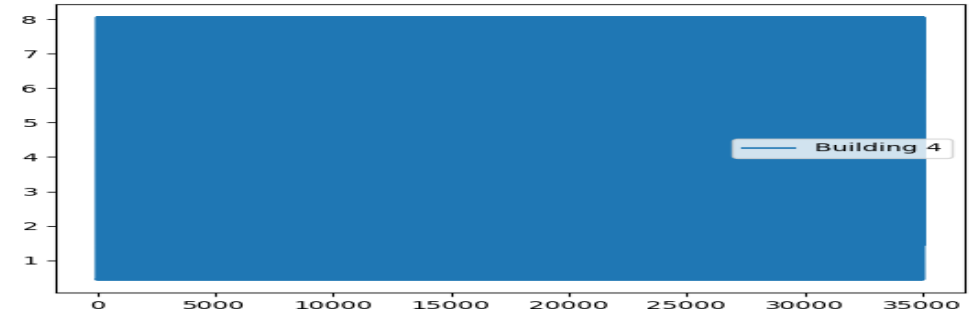

Figure Appendix C.22 – Electrical equipment demand for building 4

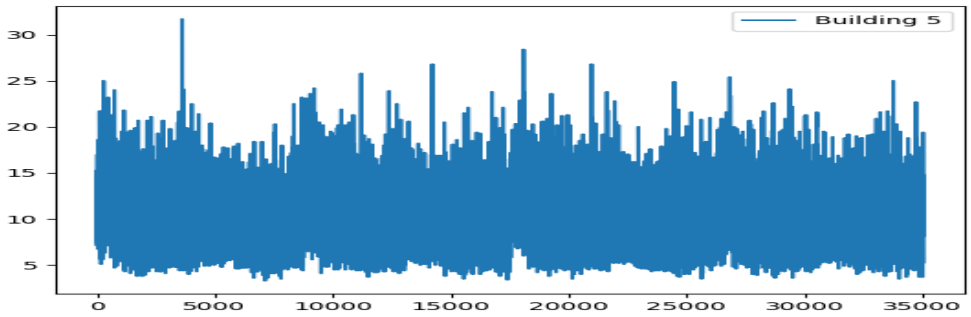

Figure Appendix C.23 – Electrical equipment demand for building 5

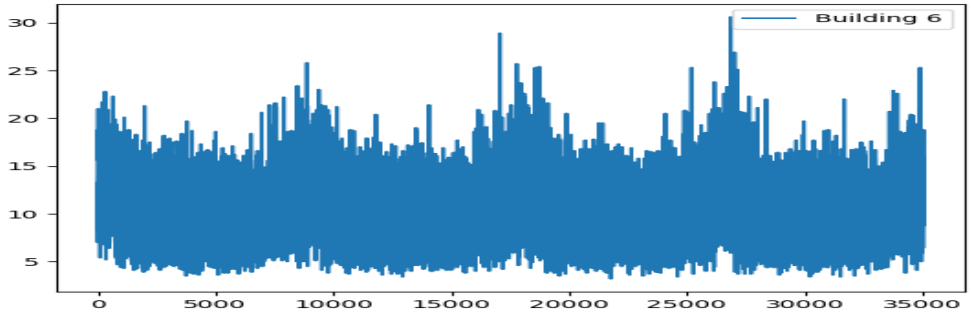

Figure Appendix C.24 – Electrical equipment demand for building 6

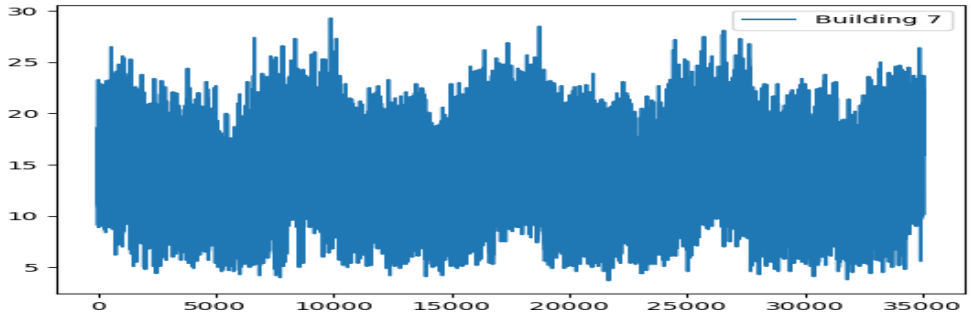

Figure Appendix C.25 – Electrical equipment demand for building 7

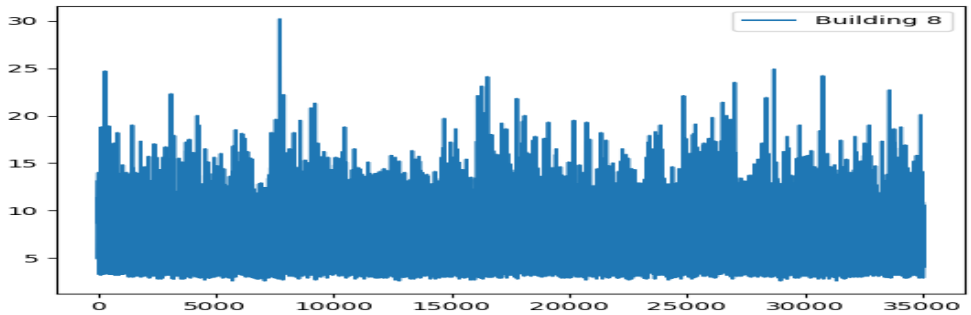

Figure Appendix C.26 – Electrical equipment demand for building 8

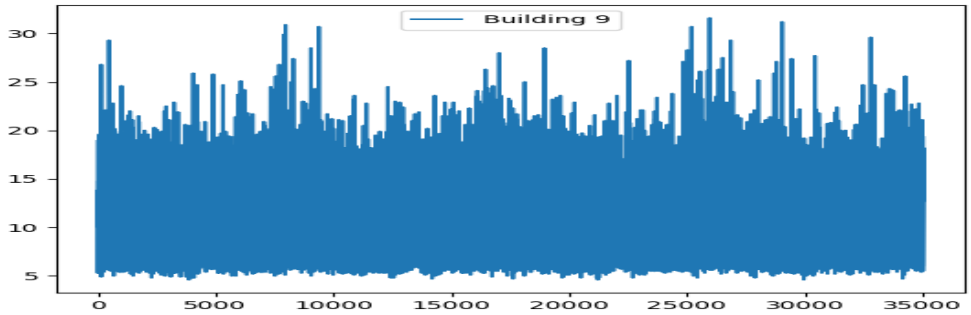

Figure Appendix C.27 – Electrical equipment demand for building 9

Appendix D

Simulation platform

We entered the era of big data, in this era, we need a valid method to process our data in an efficient way, GPUs represents the best way to make progress in this field according to the recent researches [155], GPUs were originally developed to accelerate the graphic computations, they have a high ability in speeding up the computational power, that's why the DL computations are heavily relied on it, considering all the new GPUs are designed specifically to meet DL computation needs. this research, uses a framework based on deep RL methods, one of the obstacles during the development of this research is the high prices of graphic cards, due to the high demands for it, especially in the last couple of years, the reason behind this, is the global trend of mining the cryptocurrency and gaming.

The other issue is that PyTorch, TensorFlow libraries, and even Gym library, relies heavily on CUDA, which is defined as a platform used for parallel computing, it is created by NVIDIA, and it can only work and support graphic cards that is produced by NVIDIA, another solution was to use a cloud computing service such as: Google collab, and Kaggle ...etc. But due to the limited quotas per user (approximately 36 hours of GPU available in Kaggle for free, and 25 hours in Google collab), but keeping in mind that free cloud computing service is not always available, sometimes there are no available GPU service at all, and the user must wait for a period of time, and it depends on the time you are performing the simulation process on.

That's why depending on cloud computing services is not feasible, eventually we used a hypercomputing resources provided by The Laboratory for Advanced Computing at University of Coimbra (LAC), the platform is known as Navigator+, Navigator+ is a computational infrastructure for research activities, its computers use the CentOS distribution of the Linux OS, its commands are using "bash" shell [156]. for queue management system, Navigator+ uses Simple Linux Utility for Resource Management (SLURM), which is an open source, and scalable system, used for cluster management and job scheduling for Linux systems [157]. To submit a job in SLURM, one needs to set the environment, and there are two ways for achieving it, by GUI where the users can load Anaconda-Navigator module, which considered a GUI Python distribution for package management simplicity. It is included inside the Navigator+ platform, with many other modules, these tools are enabled by the Linux module system in the Navigator+ to suit every user needs.

For managing resources and using SLURM, we need to master the most important command of it, but in this research, I am only focusing on the job submission part, and I will give a quick review of other commands. Using the manual pages, all SLURM commands and API functions can be found in [157], the most relevant commands can be found in table (14.1).

Table Appendix D – Important commands for SLURM

Command name	Description
sacct	Is used to report accounting information about current or finished jobs or job steps
salloc	Is used when we want to allocate resources for our job
sbatch	It is used for job submission, normally the script contains many srun inside of it to do parallel jobs, and it will be executed later
scancel	Is used for job cancellation, the state of the job doesn't matter, scancel can be used for a running job or for a pending job
scontrol	Is an administrative command, it can be used for state modification, and to view states, and to report a detailed information about nodes, but this command can only be executed by the user root
sinfo	It is used for reporting purposes of partitions and nodes states
sprio	It is used when we want to check what is affecting our job priorities. Although in most cases it is a resources issue
squeue	It is a way of reporting the running jobs and showing the priority orders, you can type squeue -u username to check your jobs queue.
srun	It is a way of submitting job steps, srun contains many options like the minimum and maximum node count, or if you want to specify a certain feature of the node ...etc
sview	Is a GUI option to show the different information, managed by SLURM
Purge	It removes all the modules, which considered the first step before loading other modules
module avail	To view all the available module with different versions.
CUDA ML	Load CUDA module for GPU computation (in case you are using CUDA)

For job submission, common mode is to submit a script that contains job steps, for a later execution, first we have to define the number of nodes needed for the job, nodes are defined as objects that have features assigned to them, the users should specify which feature is required for his/her job, the second line of our submission script is a request for a GPU resources, by defining an option known as Generic Resource Scheduling (GRES), the generic resource scheduling cannot be allocated by default, it must be specified in a job step inside the script. Additionally, there are built-in features, that can be enabled for choosing a specific type of GRES, such as: Graphics Processing Units (GPUs) and CUDA.

Then we need to specify the number of tasks in our job, in our case the script only contains one task, specifying the number of tasks can be helpful in case we have more than one task, within the same batch script. Then we can attach a time limit for the job, and define how much memory is needed for the job, in both Google collab and Kaggle platforms, the simulation period of MARLISA decentralized agent is approximately 3 hours, in case of Navigator+ it is more than this because we don't allocate many resources for the job, and the reason behind that, is because in SLURM there is a priority order for users, if a user allocated much resources, this is going to downgrade his/her priority, and vice versa. Lastly, we named the "STDOUT" and "STDERR" files, where the output and the error files are going to be written, before submitting the job, we must load all needed modules and packages for the job, Additionally, we can load Anaconda, if the user wants to use anaconda environment, to setup all the libraries and the dependencies.

Job submission example script

This part presents a SLURM script example for job submission using NAVIGATOR+, all job steps are explained in the table below the script, table appendix D.

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --gres=gpu:1
#SBATCH --ntasks-per-node=1
#SBATCH --time=05:00:00
#SBATCH --mem=3000M
#SBATCH -A
#SBATCH -p gpu
#SBATCH --cpus-per-task=4

#SBATCH -o hostname.out # File to which STDOUT will be written
#SBATCH -e hostname.err # File to which STDERR will be written
#SBATCH --mail-type=END # Type of email notification-
BEGIN,END,FAIL,ALL
#SBATCH --mail-user= TalhaappendixD@my.ipleiria.pt # Email to which
notifications, Write your email here

# Prints the working directory, name of the assigned node, and
# date/time at the top of the output log.
pwd; hostname; date

module purge
export PATH=/veracruz/home/a/atalha/env386/:$PATH
module load Python/3.8.6-GCCcore-10.2.0
module load CUDA/11.1.1-GCC-10.2.0
python3 example_marlisa.py
/usr/bin/nvidia-smi
Uptime
```

Table Appendix D – explaining Job script's steps

Job step	Description
`#!/bin/bash`	Tells the system what type of interpreters to run.
`#SBATCH --nodes=1`	Define the number of nodes
`#SBATCH --gres=gpu:1`	GPU resource request
`#SBATCH --ntasks-per-node=1`	Number of tasks performed by the user
`#SBATCH --time=05:00:00`	How many hours/minutes your job is going to take, but it is important to know that if you specified a smaller number of hours than what is needed for your job, your file won't execute and show you an output, so it is better to set the time to a high value, in our case the simulation takes almost 3 hours that's why time is set to 5 hours
`#SBATCH --mem=3000M`	Amount of memory needed to perform your job, but it is important to mention that in SLURM consuming to much of resources will affect your position in the priority list
`#SBATCH -A`	The name of the account provided by the LCA
`#SBATCH -p gpu`	Type of the resource needed, it is better to first check and confirm which resource is available (idle) by typing: sinfo
`#SBATCH --cpus-per-task=4`	Number of CPU needed, you can estimate it by performing your simulation in an online cloud GPU/CPU service first before using Navigator+ such as: Kaggle, Google collab, ..etc. to observe your job performance in real time, and give you a better estimation about your job requirements
`#SBATCH -o hostname.out`	Output file, which will be exported after the job is finished to the directory you submitted the job from

`#SBATCH -e hostname.err`	Error file, which will be exported after the job is finished to the directory you submitted the job from
`#SBATCH --mail-type=END`	Email notification when the job is finished, but it is possible to change the type of the notification to `BEGIN,END,FAIL,ALL`
`#SBATCH --mail-user= TalhaappendixD@my.ipleiria.pt`	Your email account
`pwd; hostname; date`	To print your current directory and the hostname and the date in the output file
`module purge`	It removes all the modules, which considered the first step before loading other modules
`module load Python/3.8.6-GCCcore-10.2.0`	Loading Python module, to do so, first print module avail to check all the available modules in the system and then copy the name of the required module, and paste it in your script
`module load CUDA/11.1.1-GCC-10.2.0`	Loading the CUDA module, using the same method
`python3 example_marlisa.py`	Your python file