INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO

MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Modelo de Controlo preditivo para comunidades de Energia

BRUNO DA SILVA ROSÁRIO

novembro de 2024

Model Predictive Control for Energy Communities

Bruno da Silva Rosário

Aluno nº 1220184

Dissertação para obtenção do Grau de
Mestre em Engenharia de Inteligência Artificial

Orientador: Luis Lino Ferreira, Professor Coordenador do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Co-orientador: Isabel Cecília Correia da Silva Praça Gomes Pereira, Professora Coordenadora do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Supervisor: Tiago Carlos Caló Fonseca, Investigador no Soft CPS Research Group do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Júri:

Presidente: Zita Maria Almeida do Vale, Professora Coordenadora Principal do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Arguente: Luís Filipe de Oliveira Gomes, Professor Adjunto do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Orientador: Luis Lino Ferreira, Professor Coordenador do Instituto Superior de Engenharia do Porto do Instituto Politécnico do Porto

Porto, 09 2024

Resumo

Nos objetivos da união europeia para 2050 está consagrado que a união deve de ser neutra em carbono, para tal a utilização de energias renováveis será praticamente obrigatória. A eficiência de produção e utilização de energia elétrica é um tema pertinente. Para esse objetivo métodos de análise e controlo de produção e utilização de energia estão atualmente a ser alvo de projetos de investigação, neste ramo a inteligência artificial mostra boas promessas para a resolução deste problema.

A solução apresentada é um sistema que conjuga Model Predictive Control com Reinforcement Learning, onde o Model Predictive Control faz o controlo de batterias, aquecimento de agua e arrafecimento do espaço para a otimização de um edifício e o Reinforcement Learning é utilizado para a melhoria do modelo tendo em conta os erros de previsão anteriores para melhorar as previsões futuras. A solução passa pela criação de um sistema modular onde as necessidades de cada edifício que fazem parte do sistema são avaliadas e geridas. A solução também ajuda a introduzir a produção própria de energia, a utilização de baterias estáticas presentes no edifício e por fim gerir as cargas elétricas que possam ser geridas como por exemplo uma bomba de calor, continuando a fornecer energia a cargas que não possam ser calendarizadas como por exemplo uma lâmpada.

Os diversos casos de uso que foram testados permitiram fazer a avaliação do melhor algoritmo, os casos de estudo relativos a comunidado o sistema demonstrou melhores resultados que os sistemas fornecidos pela plataforma CityLearn. Apesar de não ter uma performance melhor que todos os sistemas com que competiu o sistema apresentado não necessita de passar pelo dataset de 4 anos antes de começar a fazer as optimizações, o algoritmo correu em apenas 1 episodio onde aprendeu, treinou e tentou optimizar os inputs.

Palavras-chave: Comunidades de energia, Aprendizagem por reforço, Modelo Controlo Preditivo, Gestão energética

Abstract

The European Union's objectives for 2050 state that the Union must be carbon neutral, for which the use of renewable energy will be mandatory. The efficiency of production and use of electrical energy is a pertinent topic. To this end, methods of analysis and control of energy production and use are currently the target of research projects, in this field artificial intelligence shows good promise for solving this problem.

The solution presented is a system that combines Model Predictive Control with Reinforcement Learning, where Model Predictive Control dose the control of batteries, domestic hot water and cooling for the optimization of a building and Reinforcement Learning is used to improve the model by using the error of the previous predictions to better itself. The solution involves creating a modular system where the needs of each building that are part of the system are evaluated and managed. The solution also aims to help introduce its own energy production, the use of static batteries present in the building and finally manage electrical loads that can be managed, such as a heat pump, while continuing to supply energy to loads that cannot be scheduled, such as a light bulb.

There were multiple use cases that allowed the evaluation of the best algorithm, in the case studies relative to the energy community the system demonstrated better results than the systems present of the citylearn platform. Although it did not show the best performance of all systems, the system it lost to dose have a disadvantage that being it needs to first learn the entire dataset and only after dose it start to optimize, while the algorithm that is presented here only uses 1 episode where it learns, trains itself and tries to optimize the inputs.

Keywords: Energy communities, Reinforcement Learning, Model Predictive Control, Energy management

Index

List of figures

List of tables

Acronyms and Symbols

List of Acronyms

AI - Artificial Intelligence

MPC – Model Predictive Control

RL – Reinforcement Learning

MPC-RL - Model Predictive Control with Reinforcement Learning

DRL – Deep Reinforcement Learning

RBC – Rule-Based Control

HEMS - Home Energy Management System

ML- Machine learning

SOC – State of charge

LSTM - Long short-term memory

CNN-LSTM - convolutional neural network - Long short-term memory

DRL - Deep Reinforcement Learning

RBC - Rules Based Controller

MILP-MPC - mixed-integer linear programming - Model Predictive Control

EV - Electric Vehicle

HVAC - heating, ventilation, and air conditioning

V2G - Vehicle 2 grid

CIA - Confidentiality, integrity, and availability

KPI - Key Performance Indicators

CS - Case Study

1 Introduction

1.1 Contextualization

In the objectives of the European union for 2050 there is the goal of being carbon neutral, for that effect the usage of renewable energies to its maximum will be a necessity, the efficiency of production and utilization of electricity can facilitate this process [1]. Electricity is an integral part of our daily lives -we can't live without it - we use it to supply water, make traffic lights work, heat our houses, and cook our food.

Due to the need to improve reliability and resilience of the power grid, new ways to manage it are being implemented, empowered by the concept of smart grids [2]. Smart grids allow for better and more efficient integration of renewable energy by allowing decentralized energy production, among other characteristics. This t paradigm strongly benefits from the capability of being managed by a control system which can be centralized or decentralized [3].

This work has the main goal of testing the viability of using the Model Predictive Control (MPC) together with reinforcement learning for the control of assets on a smart grid/energy community whose foundation builds upon the work carried out in [4], where a multi-Agent reinforcement learning approach has been applied to the control of flexible assets in an energy community.

This work is integrated in the SoftCPS research group (Software Technologies for Cyber-Physical Systems) at the School of Engineering of the Polytechnic of Porto (ISEP), that concentrates on developing and implementing innovative solutions in Middleware for CPS, IoT and Edge Computing. It is also integrated in the OPEVA project (OPtimization of Electric Vehicle Autonomy) whose main objective is to explore the benefits that can be obtained from the interaction between the multiple actors involved electric vehicles, from its production and design to its operation in order to optimize the autonomy of electric vehicles in a modern, also considering sustainability and resource optimization [5].

1.1.1 Smart Energy Grid

Currently the energy grid consists mainly of a large set of centralized production facilities whose energy production gets transferred onto the consumer through high, medium, and low voltage power lines, this induces a significant number of losses, particularly on the transformers and on the transmission lines, which range between 4-6% [6].Smart grids are electricity networks that use technology to monitor and meet energy demands of the energy consumers and improve management of energy generation [2]. These grids are based on two-way communication, where information from the end user goes to the utility company, which uses that information to better optimize and adapt to changing electric demand.

A smart grid needs to improve reliability and resilience of the electricity grid, a disturbance on the energy grid can have a devastating effect, systems like traffic lights and communications can be disturbed when it is needed the most, on some harsher winters a failure of the energy grid can lead to people not being able to warm their houses, when a calamity happens the grid should be able to reroute the energy to either pass around the affected part or create a new route into the affected part. This, in part, will minimize or reduce the effects and time of recovery for outages. The energy can also be routed onto critical systems when the grid is coming back from a blackout thus improving efficiency and minimizing downtime of critical systems. Smart grids must manage distributed energy generation, to get all the possible benefits that they might bring. By decentralizing the grid this system will improve consumer control, by allowing to get information and manage interactions with the grids [3]. That might be day-to-day routine optimization or buying more efficient devices, although information needs to be private, secure, and untraceable basic data like how much energy a fridge consumes from a brand compared to another fridge from a different brand performs.

With the rising interest in smart grids, the rising interest in the concept of energy communities is also increasing leading to the change on energy idea.
An energy community is small agglomerate of energy producers, prosumers, and consumers usually on close physical proximity to each other where the energy is shared between all the users, this communities usually do try to use their own means of production like solar power for as much of the needs as possible, not relying as much on the national gird.

1.1.2 Previous work

As previously stated, the research starts in “A Multi-Agent Reinforcement Learning Approach to Integrate Flexible Assets into Energy Communities” where it was researched if a Reinforcement Learning approach could be used as an energy management tool, the research was based on energy communities and electric vehicles integration onto the grid, the system worked with a rules based controller in the beginning and later a full neural network with reinforcement learning.

It was found that this approach could reduce incoming electricity from the grid, electricity cost, emission reduction and daily peak load in par or better than other solutions. This solution would use photovoltaic panels, stationary batteries, electric vehicles and common building needs, where the electric vehicles could and would be used as another battery while still taking normal day to day car usage, the solution described in this document aims to do the same.

Although this system does improve on the current state of the grid dose it with some problems firstly it cannot achieve an optimal control of the grid, and due to the way, it is designed where large parts of the system are centralized and deal with sensitive data it does have increased security concerns and needs to be much more secured than a decentralized system.

The Fig.1 present in Tiago Fonseca's work shows very well how the system works:

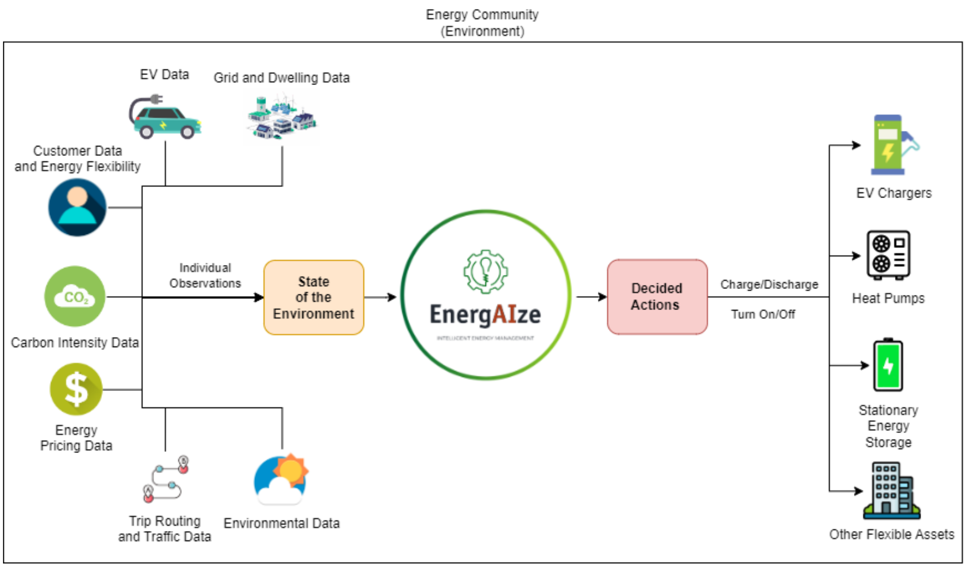


Fig 1- EnergAlze role within an Energy Community [4]

In Fig. 1 we can see that the algorithm takes the current state of the environment like electric vehicles data, environmental data, grid data the time, traffic data and the carbon intensity and energy pricing data of the electricity production. It then uses that information to control flexible loads on the system like electric vehicle chargers, heat pumps, other energy storage mediums and other flexible electric systems, the proposed solution is one where we relace the center of that image the Reinforcement Learning module with a Model Predictive Control with Reinforcement Learning module.

1.1.3 Why Model Predictive Control?

A MPC works by using a model to predict the future behavior of a system with the selected actions over a finite time window, based on this predictions of the future state of the system an optimal control inputs can be inferred using a defined objective or a set of constraints [7].

The System can do two sets of actions A or B. On this precise instant Action B is better than Action A but if we act A now our future is better than action B future, the System reaching the conclusion that action A is better on the long run the model will choose action A now.

As an example, we will use the following fictional scenario to convey the idea of how a MPC works, in this fictional scenario there is a shop that can only stock one product P1 and P2, we know that on the next day P1 gives a profit of 10€ and P2 gives a profit of 5€ but when this action is taken in consideration of multiple days the model predicts that P1 will give a profit of 50€ and P2 gives a profit of 60€, with this new information the store can choose to stock P2 and not P1 since for the same time it has more profit.

This system needs a prediction algorithm to “see into the future” so it can find the best possible usage of electricity for the users need.

1.1.4 The Problem Statement

With the rising numbers of electric vehicles, the energy grid needs to be reinforced and reworked onto a system that can support such demands, there is also a rise of the Ambiental conscious individual, where global warming and air pollution are worries, that has created a demand for clean energies. According to the European Environment agency currently 22,5% of the energy consumed in the European zone is renewable energy, the same agency notes that energy flexibility is a must have for the achievement of 42.5% of renewable energy share by 2030 [8].

A way to manage the energy grid will be necessary, managing self-production, electric vehicles, battery management, the customer needs, the community needs and the energy routing, all of this is needed for a future where we can rely solely on renewable energy, this is the problem that we will try to address, how to manage efficiently a power grid in the near future and all of the problems that brings, integrating the concept of smart grids by allowing small communities or individual buildings to go “smart” while maintaining functionality is paramount for the success of this system and as of writing there is no large-scale solution to this problem.

1.2 Proposed solution

This thesis will address the need to have energy management on buildings, it will combine energy generation, energy storage and energy usage. For that we will use a Model Predictive Control with Reinforcement Learning approach. The model predictive control will optimize the usage of the energy created, optimize the usage of batteries and electrical vehicles, while maintaining the requirements of the building, the user input on best achievable end goal is also considered where it being reducing carbon creation, saving on monthly electricity bill or being the most self-reliant as possible.

The reinforcement learning will adjust the parameters to create the best possible outcome to the needs of the building or the Energy Community and its owner. This system will aim to decentralize the energy grid while improving resilience and reducing carbon footprint by allowing people to manage energy sources and profit out of it.

With this thesis the user will be allowed to use their energy sources, energy storage like batteries to either profit or reduce their own carbon emissions. Using edge computing and a custom message protocol allows for efficient communication and information security, while allowing the system to work independently and at the user's location.

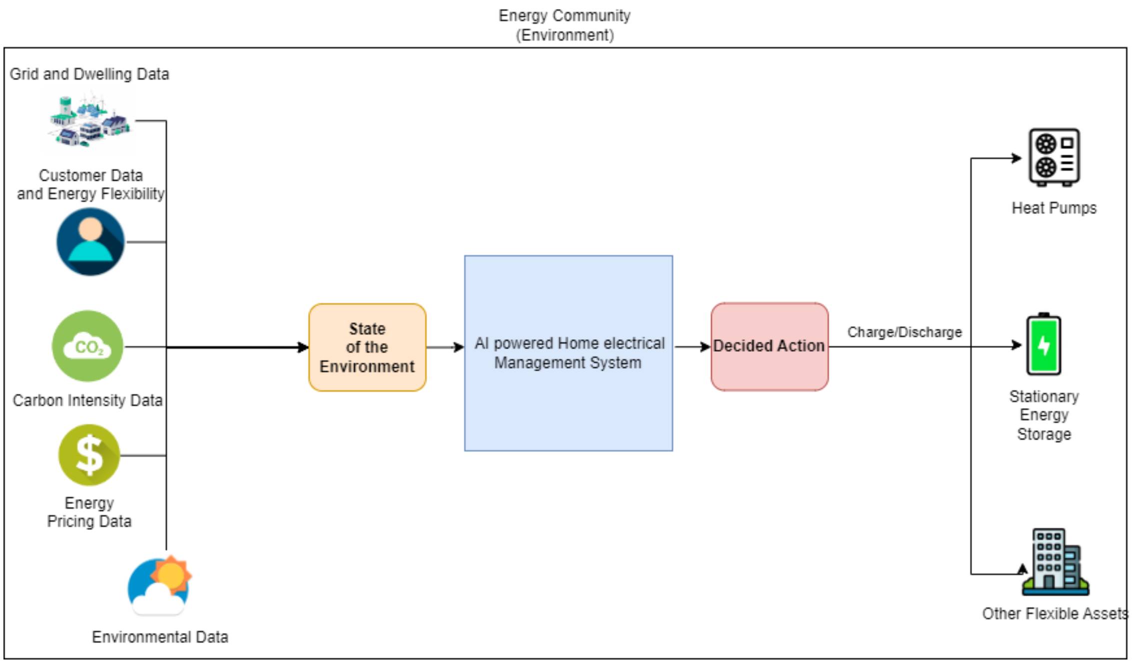


Fig 2 - System flow diagram

1.3 Thesis Objectives and Research Questions

To ensure the research aligns with the scope of this work, this section formally outlines a set of project objectives and Research Questions (RQ).

1.3.1 Objectives

Following the proposed solution in Section 1.2, the Main Objective (MO) of this thesis is defined as:

MO – Design, implement and evaluate a new energy flexibility scheduling framework based on MPC and RL for the integration of flexible assets into energy communities, considering user engagement, mixed personal and community objectives, scalability, and real-world applicability.

In a way to facilitate the achievement of the main objective, multiple Sub Objectives (SO) for this work were defined. They are described as follows:

SO1 - Investigate the current state-of-the-art on Renewable Energy Sources, Energy communities and predictive control systems.

SO2 - Investigate the current state-of-the-art of energy flexibility management methods of Model Predictive Control and evaluate their applicability.

SO3 - Design a decentralized multi-agent Model Predictive Control approach that manages and optimizes flexible energy assets considering individual objectives.

SO4 - Evaluate, tune, and optimize a solution, use different optimization objectives and EC scenarios, evaluating its performance, scalability, and adaptability to real-world scenarios.

SO5 - Benchmark the proposed solution against other systems.

SO6 - Evaluate the tests with regards to environmental, financial and prosumer engagement.

1.3.2 Research Questions

To guide the research performed in the scope of this thesis and successfully accomplish the established objectives, the main research question to be investigated was carefully formulated as: "How can AI help to beneficially manage energy flexible assets, such as EV, inserted in Energy Communities?" The main question was divided into three narrower sub-questions:

RQ1 - What is the current usage of predictive control with reinforcement learning for energy management?

RQ2 - What usage has been found for model predictive control?

RQ3 - How to integrate user preferences into Model Predictive Control with Reinforcement Learning?

1.4 Contributions

The following section outlines the thesis contributions to both the scientific and societal spectrums.

Planed Scientific Contributions

This work was developed by Bruno Rosário as a master's thesis for an Artificial Intelligence course in School of Engineering of the Polytechnic of Porto (ISEP), it aims to develop and evaluate a system that uses Model predictive control with reinforcement learning for energy management.

The thesis contributions to the scientific field can be resumed as:

SctC1 - A review on the background concepts, benefits, motivations, and challenges of the intelligent management of flexible electrical assets.

SctC2 - A comprehensive literature review on the existing techniques for energy flexibility management.

SctC3 - The results of implementing and benchmarking, AI-HEMS, a decentralized multi agent model predictive control reinforcement learning algorithm for managing flexible assets.

1.5 Thesis Structure

Section 2 - Background and State of the Art, this chapter goes into the general background information about smart grids, Model Predictive Control and reinforcement learning, it also delves into what the current strategies of Predictive Control with Reinforcement Learning, the use cases of Model Predictive Control and how to integrate user preferences into Model Predictive Control with Reinforcement Learning.

Section 3 – Solution description – this chapter delves onto a more critical and higher-level analysis, where some design and future problems will be explained. This chapter will also delve into the data protection and security side of an implementation.

Section 4 – System Overview – in this section it and explanation of how the system proposed was implemented, it is architecture, and the general algorithms used in this system.

Section 5 - KPIs, Data set information and Test cases – this section explains the KPIs used for this project, how the simulation environment works and explains the dataset and the information contained in it.

Section 6 – Results – This section presents the results achieved by this system and comparing to other systems.

Section 7 – Conclusion - this section presents the conclusions that we were able to reach with this project, future work, improvements, and key findings.

2 Background and State of The Art

To understand the thesis thematic and to refine the proposed solution (presented at Section 1.2) this chapter aims to respond to the research questions presented in Section 1.3.2.

2.1 Background

Here it is explored some background concepts to better understand some critical ideas and systems. This ensures that the reader has all the tools to understand the major points of this thesis by explaining in higher detail what Reinforcement Learning, Model Predictive Control, energy communities are and introduces the citylearn framework.

2.1.1 Base Concepts

The following sections present the main concepts related with Reinforcement Learning, Model Predictive control, and energy communities.

2.1.1.1 Reinforcement Learning

Reinforcement Learning is a type of ML that evaluates the changes on the environment to create a better ML algorithm though the iterative nature of this method.

In this RL we have 3 distinct parts of the system the state space, action space and a reward signal.

- state space – the state space is all available information that exists and is useful for the system, this information can be measured or estimated [9]. An example of a state space on the scope of this thesis this would be the collection of sensor data and smart appliances.
- action space – The action space is all the decisions that the system can take [9]. An example of an action space on the scope of this thesis would be to charge, discharge, or do nothing with the static batteries, turn on flexible loads.
- reward signal – the reward signal is a value that measures if the action had a positive or a negative impact on the overall performance of the model [9]. An example of a reward signal on the scope of this thesis would be the error of the predicted value to the true value after executing the actions.

The generic operation of an RL algorithm is as follows:

1. gather the variables of state space.
2. the system decides an action to take by taking the variables of state space.

3. actions from the action space are selected and executed on the environment that the system is being run.
4. a reward is calculated by checking how the environment responded.
5. that reward is used to calibrate the system.
6. Restart the loop by going to the first step.

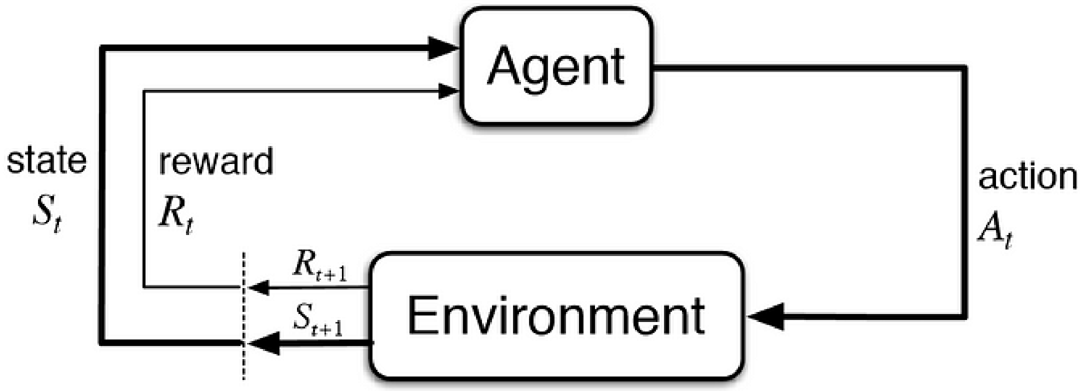


Fig 3 – simplified version of a reinforcement learning algorithm [10]

The generic operation of a reinforcement learning algorithm as seen in figure 3 is based on an agent interaction with an environment, receiving its new state and a reward, the Environment can be physical, simulated, or virtual in nature.

In certain cases, it is possible to add constraints and uncertainty into the model [9], constraints can be something like the power remaining of the battery cannot be lower than 0 and on uncertainties we can have the example of a blackouts, most blackouts cannot be predicted. A way to insert uncertainty onto a RL model is to penalize when the system fails to deliver an acceptable answer [9] on the case of this thesis the user using more energy due to using a oven and overshooting the available power budget is seen as a uncertainty, the reward on this case should heavily penalize the system.

RL can be either online or offline trained. On one hand, an online RL system is trained with either a simulation or where applicable, a physical space, where the system takes an action usually with uncertainty, that action is executed on the environment a reward is calculated and returned onto the system. On the other hand, offline RL systems where offline trained systems are trained using historic data and learns the policy though that data, this dose create a less responsive system [9].

2.1.1.2 Model Predictive control

A MPC is an algorithm which predicts the control of a system using a mathematical model. A MPC works on a time scale, where the model has a reference *X*, a set of possible input variables

Y, a set of measured outputs *O* and a cost function *C* where *X*, *Y* and *O* are vectors. Some MPC models will also consider disturbances but that will not be discussed here.

The MPC First will look towards the current reference X_0 and the current measured outputs O_0 where an optimal control of the input variables Y_0 is found by minimizing C, the types of algorithms to calculate *Y* by optimizing *C* will be discussed later in this document, using the calculated Y_0 control vector the system will try to predict the next system measured outputs O_1 where the same reference or a new set of references values X_1 is used to once again calculate new inputs Y_1 and the system restarts, until either a stop condition is reached usually 20% of the samples [11] or an equilibrium is reached where an equilibrium can be achieved and the optimal solution is executed.

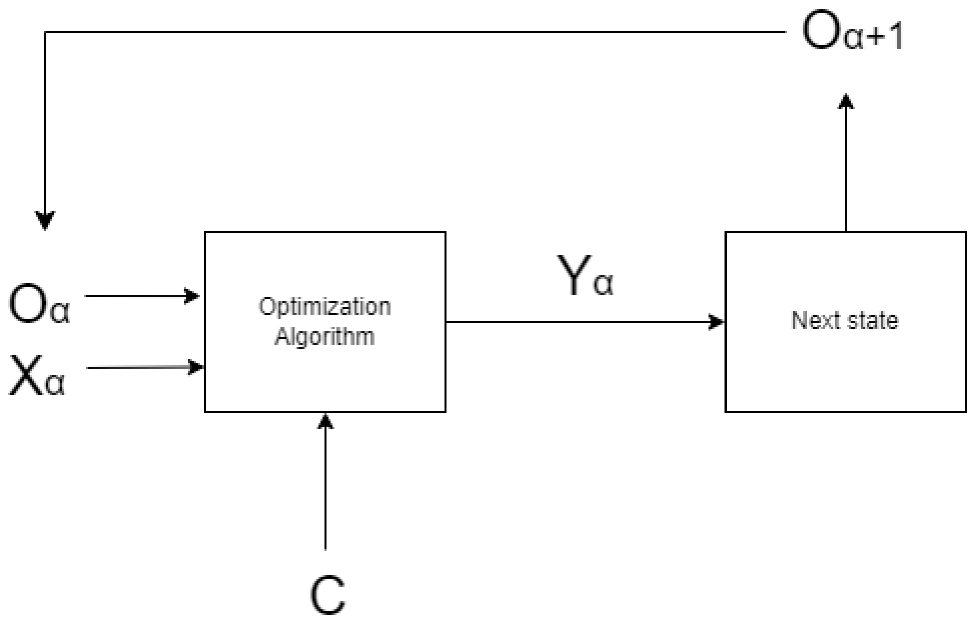


Fig 4 - MPC optimization loop

In the figure above X is a function or a set of parameters that indicate what is the reference it should target. E.g. a temperature set point or an EV range. O are the predicted outputs for the current iteration of the model. Y is a set of actions the system can take, can be a simple turn on or off or a more complex one like how many watts of cooling to use. C is a cost function to obtain the most favourable outcome for the system.

An MPC algorithm can have different classes like linear, nonlinear and hybrid.

- **Linear MPC** is a type of MPC where the objective function is linear or quadratic and the prediction model is linear [11].
- **Nonlinear MPC** is a type of MPC where the objective or the prediction model is non-linear [11].

- **Hybrid MPC** is a type of MPC where the prediction model is linear, but it employs other types of constraints like switching dynamics, binary or integer control variables, logic states or constraints [11].

There is also the Implicit MPC and Explicit MPC, an implicit MPC is a type of MPC where a solution is calculated at the run time by trying to achieve the best possible solution, this is done by an iterative method where the optimum solution set is recalculated until it reaches convergence [12].

An explicit MPC where the optimum solution is precalculated, to achieve this we separate the system in regions where the system starts by calculating a region and it is optimum function, this reduces the amount of processing power needed to achieve a good solution while reducing the region of values that is needed to be searched [11].

2.1.1.3 Energy communities

An energy community is an aggregate of buildings normally close geographic proximity usually with self-energy generation either by building or as a community space, can or not have means to store the energy like batteries and the buildings have means to communicate and manage the energy system as shown in figure 5.

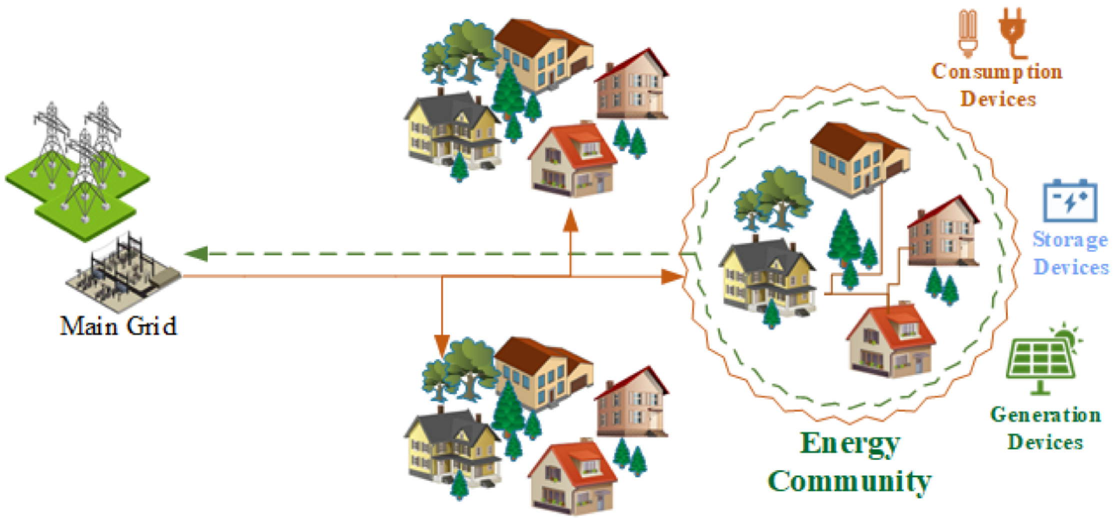


FIg 5 – Energy community scheme from [13]

These communities have three different types of members the producer members that only produce electricity, the consumer members that solely consume electricity and the producer and consumer member these last combine the energy production and usage of the other two by having means to produce and utilize energy like having a solar panel on their house.

Energy communities have a decentralized production system this helps increase its resilience, the decentralized nature of this idea means that renewable energy generation is favoured to more traditional and more pollution methods. The community shares electricity means whether that be by the sharing of excess production or the trade of electricity, this ideal is also very connected to smart grids a further explanation of smart grids can be read at 1.1.1.

2.1.1.4 CityLearn Framework

The CityLearn Framework is a project developed by Jose Vazquez-Canteli, Kingsley Nweye and Zoltan Nagy that aims to provide an environment to test solutions to the electricity management problem it supports the management of self-production means like photovoltaic panels, the use of static batteries, use of climate control between others, CityLearn handles the management of data and handles the calculation of other metrics like the amount of energy that a building uses, this means that anyone can create a dataset or an agent and publish it and other researches can use them [14].

CityLearn already provides some agents that can be used to evaluate against other systems and provides some datasets both of which will be used later to get the comparation of results and the environment where we are going to compare against other systems to access the performance of this system.

2.2 State of the Art

No one starts from scratch, there is a need to understand what has been previous done, problems that other projects had and the solutions they found, research question 1 is a general state of the art where there is the search of similar solutions and how they were implemented, while research question 2 and 3 go into more detail on a specific point like the integration of electric vehicles and how to implement user preferences onto a MPC-RL system, this gives a broad vision of the current state of the art and how to proceed with the development of this project.

2.2.1 Research Question 1 - What is the current usage of predictive control with reinforcement learning for energy management?

The systems that were found that used predictive control and reinforcement learning for electrical energy applications where different models like CNN-LSTM, MPC, RL, MPC-RL, MPC with Q-learning, online DRL training, offline DRL, RBC model, MILP-MPC models and DRL. Due to the differences in simulation environment, goals and use case it is not possible to compare one system to another at this phase thus each study was individually addressed.

A study comparing transformers and Convolutional Neural Networks and Long-Short Term Memory more known as CNN-LSTM with an application in Heating, Ventilation, and Air Conditioning systems where the neural network using transformers had a better performance than the CNN-LSTM and this solution saved almost 50% of energy compared to traditional methods [15].

Another study compared MPC, RL and MPC-RL models, this study used BOPTEST another simulation environment to evaluate the different systems. It was found that although MPC has a short fall by being too static RL can address this by adapting the model to the needed system [16].

This study analyses the difference between online DRL training, offline DRL, an RBC and a MPC based system. This study had as an objective the cooling of a room where it was found that the online DRL model had the worst performance, MPC had the best performance with the offline DRL model a close second, the RBC was worse by a small margin to both the offline DRL and the MPC [17].

Another study focused on a MILP-MPC system for an energy community with the use of a Q-Learning RL algorithm for energy planning, where a central energy system with capacity to store the energy was used, had V2H capabilities and solar panel energy generation. Although more focused on energy management it did find that aggregating an energy community did create the situation where more energy was sold, and less energy bought than a system that was more individualistic [18].

A study also tried to use an MPC-RL model on 3 households seeing an improvement of 17.5% in electricity costs while combining uncertain user demand, renewable power generation, managing peak power and market prices. This study one was based on a dataset of 3 households in Oslo, Norway thus making the simulation closer to reality [19].

A study about a home energy management system using MPC-RL model managed to satisfy thermal comfort and reduce costs by managing a battery and the house thermal inertia [20] [21].

The last study to look at used a DRL algorithm for hot water temperature control to optimize the usage of Photovoltaic panels, was able to keep the user's comfort where a temperature was specified at a specific time and the system needed to have that temperature at 99% of the setpoints of that temperature, it managed to have 16% of energy savings while improving Photovoltaic Usage for self-production and increased renewable energy consumption by 9.5%, when paired against a RBC algorithm [22].

It is possible to infer that an MPC-RL model can be used to control and manage energy communities, while taking the short comings out of just a MPC model, a paper also observed that a MPC system managed to beat an offline trained DRL, an Online trained DRL and an RBC.

After evaluating the solutions a MPC-RL based solution should offer good performance on energy management although most systems focus on small parts like temperature management there were others that already combine home energy production with this types of systems and those studies also have good performance, certain techniques like energy set points are going to be used later on this paper, we also saw that utilizing a RL with a MPC model makes to model able to have better performance and adaptability.

2.2.2 Research Question 2 - What usage has been found for model predictive control on Energy management with electric vehicles?

A MPC system can be used in multiple ways using electric vehicles from thew objective of this thesis building energy management, traffic energy management, EV charging management, wheel power management and battery management. In this research it any vehicle that is not fully electric is not considered, there is a need to also investigate other vehicle types like hybrids, fuel cell electric vehicles and range extended electric vehicles. These types of vehicles are not taken into consideration due to the nature of this thesis.

Tacking first the traffic energy management a speed management system was tested with the regenerative braking energy used to run the car AC [23] another system was proposed where constraint stochastic model predictive control was used to plan the speed of the vehicle and improve powertrain energy management [24] another research paper proposed a system to manage battery usage by choosing what motor to enable for the most efficient energy usage [25], a similar system was also proposed where only 2 engines exists that system managed to improve speed tracking accuracy by 58.93%, the high efficiency range of powertrain by 40.93% and electric consumption was reduced by 9,29%, [26] with a last research being done on improving energy consumption and the life of mechanical parts of the car on a two motor electric car [27], a system was also proposed where a user would input the needed charge for the predicted usage of the vehicles and the system would coordinate that user input with other users to manage the EV chargers [28].

The second use case EV charging management where all followed the same system the charging station has access to renewable energy and the grid, one incorporated batteries on the system to maximize the renewable energy usage [29], another uses a predictive system to buy the needed energy from the grid to supplement the renewable energy generation [30] and lastly a system to incorporate Photovoltaic generation prediction and when vehicles connect and disconnect [31], another approach can be to predict the car usage to optimize charging times [32]. A system was also proposed for four-wheel-independent-driving electric vehicles management system [33].

A traffic energy management is a great tool that EVs can exploit, the usage of vehicle-to-vehicle communication and vehicle-to-infrastructure communication can be used to optimize the speed and braking of a vehicle this will improve driving safety, comfort, and energy economy of EVs, another avenue for EVs are supercapacitors/battery EVs on this type of EVs both types of energy storage medium have different use cases where a predictive system can help manage both systems, this did result on the decrease of battery degradation [34] another way to see this is to incorporate a similar tool to the vehicle-to-vehicle with sensors and use a system to first optimize electricity cost and ensure safety and second to minimize the battery degradation and power loss [35]. Lastly the use case of the proposed system of this thesis integrating EVs into Household energy management where a EV can be used to reduce electricity bills and EV battery degradation costs while maintaining amenities like thermal demands [36], used as a storage medium inside a micro gird [37] while reducing peak load on the grid [38], managing multiple EVs to increase or decrease the charging time [39], energy management withing a smart community to better allocate resources from the grid [40], another use case is one where the speed of charging changes with the HVAC needs of the building [41]. Still within this group we find the Household electric systems where a photovoltaic Source is used with heat pumps, a thermal energy storage and the electric vehicle being used as a battery to supply energy into the house are taken into account to optimize cost and emission optimizations [42], another system even proposes that the energy stored at the EV be used to sell to the grid [43].

A MPC system can be used for many types of system, the types found that are not important to this thesis were the traffic related management and the battery management systems, where we find the systems capable of optimizing the EVs during use and in the case of one optimize each individual wheel, a system like the ones described above could be a great way to reduce energy consumption even more thus easing the transition into fully renewable sources. A way to better and more efficiently manage charging with multiple EVs is also a good avenue of research this one more focused towards large charging systems and less the household one, on the household front we see the prevalence of the usage of EVs as a way to use more efficiently the photovoltaic power generation, store energy for later usage without forgetting that it is still a vehicle and should be used as such being the power storage more a nice bonus and less the primary goal.

2.2.3 Research Question 3 - How to integrate user preferences into MPC-RL models?

The usage of reinforcement learning implies that a way to appraise how good or bad a solution, and due to the nature of this system a way for the user to manage and better suit the system to their own needs. For that a solution that integrates Reinforcement Learning with user preferences must be integrated.

In this research it was found 6 such examples where two have hard constraints and six with soft constraints.

Checking the ones with hard constraints first we have a system where the user tells the type of ride that wants where it can either share the car or not [44], the second where the user tells the system the waypoints it needs to arrive to and the order [45] , this systems although valid and with proved use cases, on the proposed system of this thesis these restrictions should be integrated within the MPC part of the system and not the RL.

The soft constraints that were found where defined by a range or a more subjective user input apart from one. Most systems focused on thermal energy management with 3 different approaches the first with discriminated temperatures at set intervals where the system should hit those values a certain percentage of the setpoints [46] , the second where a temperature range is set, and the system has to have the temperature between that range at specific intervals [47] ,the third and last has both systems where an acceptable temperature range is set with a preferred temperature where between that range the rewards of the RL model are less than outside the specified range, if the system hits the target temperature it receives maximum rewards, the rewards is reduced within the range with a large reduction outside of that range [22] . The other soft constraint system is going to be analysed differently, the system has the following available user inputs, Safety, Speed changes, length of trajectory, waypoint deviation and planned ETA, each input can be set with a range of natural numbers between 1 and 10, the ranges selected are then normalised before being used [48].

The hard constraints even if the user input can change them are better suited to be used inside the MPC, looking at the soft constraints, we have 2 different systems, the system where a preference like temperature, is continuous in range and time, the preference needs to be met. The other system is more generalized and dependent constraints, like in our case, the constraints self-reliance, battery usage, EVs in the V2G capacity, cost and CO_2 footprint where a system like [48] is interesting to research where the user can set the multiple preferences to a value within a range and the normalized values used as the input and not the direct user input, one such case could be cost and CO_2 footprint where if the cost of energy is negative and the user inputted both as maximum priority can create problems that are mitigated by normalizing the results.

3 AI Powered Home Electrical Management System

This chapter explains and proposes how the system should be integrated, this chapter is an explanation on the full system with more development should be done it answers some problems like how to better manage and what a central system should do, it delves into the security concerns and answers it, giving a full overview of the planning phase of this system. These types of systems are usually called Home electrical Management System, abbreviated to HEMS, these systems can also integrate an artificial intelligence component, so the name AI powered Home electrical Management System abbreviated do AI-HEMS gives the perfect idea of what this system does and how it does it.

3.1 Project overview

The system is designed to work solo or in tandem with other systems with that in mind a fully modular system helps to implement these two criteria.

The current implementation works solo on each building thus the priority system was not implemented, this is a requirement for future implementation of a central agent.

A layered priority-based policy is introduced to better accommodate the community aspect of this proposed system, this system allows us to firstly abstract from the individual parts of the system, two improve and ease user input systems and ease of use and thirdly to abstract the data and improve safety when transporting data outside of the user’s house.

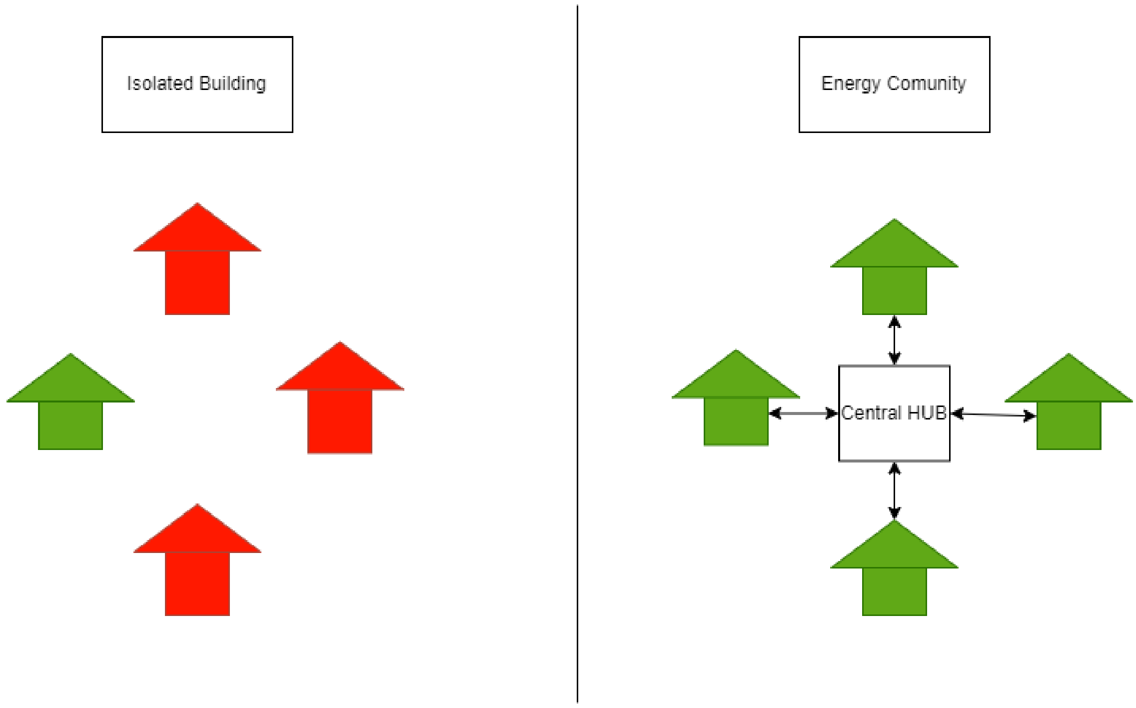


Fig 6 - Diagram of types of scenarios

A modular system brings inherent benefits to this problem, a way to ease the introduction of this system and the natural growth of it. A user can install the system on his building and the system would work without any other similar system near as shown on Fig 6, of course the benefits would not be as good as if the entire community had this system but having a way of starting this system on a community with only one building needing it, then later as other would join the rest of the system like the central manager can then be installed, for such things to happen the system must guarantee to improve energy billing even when working without a central management system.

The layer priority system helps to simplify the priority implementation, and a user's ease of use by simplifying the priority system for that there is 6 priorities listed below.

Priority 1. **Critical Public Infrastructure** – the critical public infrastructure like water distribution, emergency communication lines, hospitals, etc. must be maintained supplied, this part of the system the user cannot change the priority level of this level.

Priority 2. **The system** – the system must consider and must keep it self-operational on this part of the system the user cannot change the priority level of this level.

Priority 3. **Important building systems** – on houses this would be a fridge/ cooking appliances and lights, on a commercial building it can be lights and the security system, on this part of the system the user can and should change the items on this priority level to better suit it is needs.

Priority 4. **Essentials** – this is important systems that have a smaller importance than the more critical mentioned on priority 3, this would be washing machines, hot water, and more critical amenities, on this part of the system the user can and should change the items on this priority level to better suit it is needs.

Priority 5. **Comfort and wellbeing** – here we have everything that does not fit the categories above, like a tv or a computer.

Priority 6. **Batteries** – The batteries should be the last thing to consume energy, it should consume only when the other priorities have been met.

The case for shiftable priorities we can take the example of batteries, where a user might want to always have a small percentage of the batteries to use in case of a blackout for that the batteries should not be on priority 6 but a higher priority, this is a user preference and something to consider, so under certain situations some systems might increase or decrease in priority.

Due to the way the data is sent to the higher system by the priority system, it ensures that critical information is hidden, it also allows to latter create a system to analyse how much of a routine with that information can be inferred and penalise the system accordingly.

The idea of predicting for t+2 should also be investigated, because it allows the system to have more time to find the best solution and, reduce the impact of updating the system, since the system already predicted the t+1 and knows what to send at that iteration so it has 1h to do what it needs to do, this also puts much less strain on the speed of the central system due to having more time to process this can be achieved by reducing the time a system has to process from the theoretical 1h to a lesser time.

3.1.1 System Architecture

The system architecture was made so the model and optimization algorithm were easily changed this was achieved by the way the CityLearn agent was made. The CityLearn agent that is exposed to other applications already gives a degree of abstraction by sending the information in a standardized way, this means that in case the agent is or is not centralized, a flag set at the configuration of CityLearn, the way the agent is programmed is always the same, but since early on the development of the system, the idea of it being decentralized was always at the centre, this allowed for another abstraction layer that being the building object, this building object handles all of the information related to the building it saves the model that represents it, creates the inputs and gets the current information that was provided and saves the actions to take, reducing the amount of processes on the Citylearn agent those being updating the buildings, start the optimization algorithm, set the Citylearn action variable and call for the next time step. This means that all the needed information relative to a building is already separated and each building is processed the same way with the same algorithm, there is no need to handle the building logic inside the optimization algorithm. There is also a data retrieval in the agent but that was not made modular.

The optimization algorithm just needs to ask the building to create the input and give the actions that it needs to evaluate, the building will then process the inputs and calculate the objective function to then send again to the optimization algorithm thus easing the implementation of the algorithm.

Both the model and optimization algorithm have standardized ways to implement both having an init where the main configurations are saved and for the optimization algorithm just another function is needed to start the optimization, while the model has 2 other functions the predict that runs the neural network and the fit that handles the logic in training the neural network.

The prediction of each element should itself be isolated from other predictions for each time step, since each prediction is only dependent of our own actions from each time step, the prediction is of an outside element like the outdoor temperature or in the case of hours and months a static prediction that only relies on previous information, this system allows for the mismatch of different sources like the meteorological service for the outside temperature prediction or prediction models for other predictions that are more individualistic like the solar generation of each house.

The system was modeled after the citylearn implementation of an agent, where the standard implementation of some functions was implemented. Those functions manage standard parts of the agent like the initialization of the object setting the action map, it was also overwritten the predict function to implement what was described on this thesis, the predict function itself gets called every time step and it is also responsible for advancing the time, the loop of this function is simple if the objects for each building were not yet created it initializes them, giving a list of set parameters to ignore like the month, day type and the hour, if the buildings were already initialized that mean that we are at the first hour and thus we can start aggregating the data to later get our information, after that all building observations are updated and a function to fill the individual actions of a building is called, then the standard way of setting actions and go into the next time step is called. The function to fill the individual actions of a building works by calling the optimization algorithm and saving the actions and the predicted values, the logic for the rules-based controller that works for the first 2000h in the case of the case studies 5 to 7 is also implemented here. The optimization function works as a simple implementation, since all the settings/ input of the model are saved to the building object, that has a predict function that handles all the logic of taking the set of actions and getting the value of the fitness function, by simply calling the model predict and saving its outputs at each step of the action test object.

3.1.2 Other Approaches

Other approaches can be adding a more complex central system that uses more information, this causes the problem that a user's routine can be more easily inferred, this is a large security concern and can drive away potential users, there is a large focus of anonymity in processed data and giving more information to a central system and not an edge system like this one was designed can create such vulnerabilities and legal data protection considerations.

At some point the possibility of predicting more abstract values like the user's mood could be researched and implemented, but the methods and or data can prove tricky to adjust on a per user case. This also goes against a design philosophy of this system that all predictions are based on hard data like the temperature of a house, it is something that can be effectively measured and quantified, and that the user has an instinctive understand of, like it is indoor temperature preference or his energy usage vs the current price, this removes the user error in self-evaluation there is no prediction if the user feels more comfortable or not, this in the future can just me a way to measure how good the solution is, replacing the objective function with something less intuitive or adding the prediction of the user comfortability with the future predictions can be a value in the objective function in conjunction with the other values like the utilization of self-generated power.

The use of a large system that englobes all of the energy community or even multiple energy communities is also a route that can be taken although it reduces the users ability to control the system and optimize for its use case, this approach should be able to better optimize for a given parameter and common goal, also something that goes against this systems design philosophy of letting the user preferences and final objective easily changed, even if those setting are not better for the user.

There is a case to be done for having more of an all-in-one algorithm where the output is only the actions needed to be taken, this system again should be better and more optimized but is this system being optimized for only one parameter or can the user choose, and what impact on the system dose the user changing its idea have, dose the system need to be trained again to find a better solution for that new objective function, common logic says that the old data can be used with the new objective function but how much time dose the system need to achieve a stable point, the proposed system just takes the current user's objective function and due to being based on hard data and an optimization function, it can change at the next algorithm iteration without the user having a time with worse performance.

3.2 Data protection, Safety and Ethic concerns

3.2.1 Data protection

The nature of this data implies that it is possible to understand the user's routines and current status, this information is very dangerous, a bad party can use routines to know when a user is outside the house or that the lights work at weird times of the night this latter one can be an indicator of a user's health problems, an insurance company if had this data could increase or not insure a user, this data needs to be heavily protected , for that the distributed system and encryption with the current data system that does not send direct user behaviors through the internet, also a state of the art authentication system needs to be used to ensure the information is being sent to the correct computer and prevent man in the middle attacks.

3.2.2 Data Safety

Data Safety can be explained by the CIA triad where CIA stands for Confidentiality, integrity, and availability. On the system part Confidentiality can be assured by user data not being accessible outside of a building and encrypted within the system on the case that old user data is of use for improvement, on the case that old user data is no longer of use it should be deleted, to ensure that only authorized systems can communicate within themselves state-of-the-art authentication systems and encryption should be used, none can be selected by the author due to the constant evolution on authentication systems and encryption.

As for integrity the information should not be able to be tampered with, like the authentication and encryption problem there is constant evolution on this field so the author chooses to refrain from any requirement the only stipulation is that like the authentication and encryption this system should be state-of-the-art and adequate to this system.

Last but not least availability, due to the nature of this system, it focus on always being online with at least down time as possible this is ensured by always having the preference on making sure it has energy for himself before any other system is considered, availability is also the communication of data whether that be by a fiber optic cable, wi-fi or any other communication medium, on this further assessments and real world tests need to be executed to achieve a safe and reliable communication medium.

3.2.3 Ethical considerations

As of writing the author does not have any ethical considerations or concerns that he is aware of.

4 System Overview

In the previous chapter it was described where this system fits in a generalized way, the system should also be able to be individually used, this guarantees that this system can find more costumers due to reducing the initial set up, but when this system is present in multiple houses a centralized system might be needed to further optimize. But since even if there is a centralized system, each individual building is still optimized by itself the system presented only optimizes itself. In this chapter we are going to delve into the implementation of the system.

4.1 System implementation

This system has the objective to manage an individual building the information that the system receives comes from the CityLearn framework, that provides a set of individual set of observations per each building as shown in FIG 7. In turn the system manages individual assets like a static battery a cold storage and a domestic hot water, further assets are possible to be managed but due to the limitations of the CityLearn framework this avenue of optimization cannot be researched here.

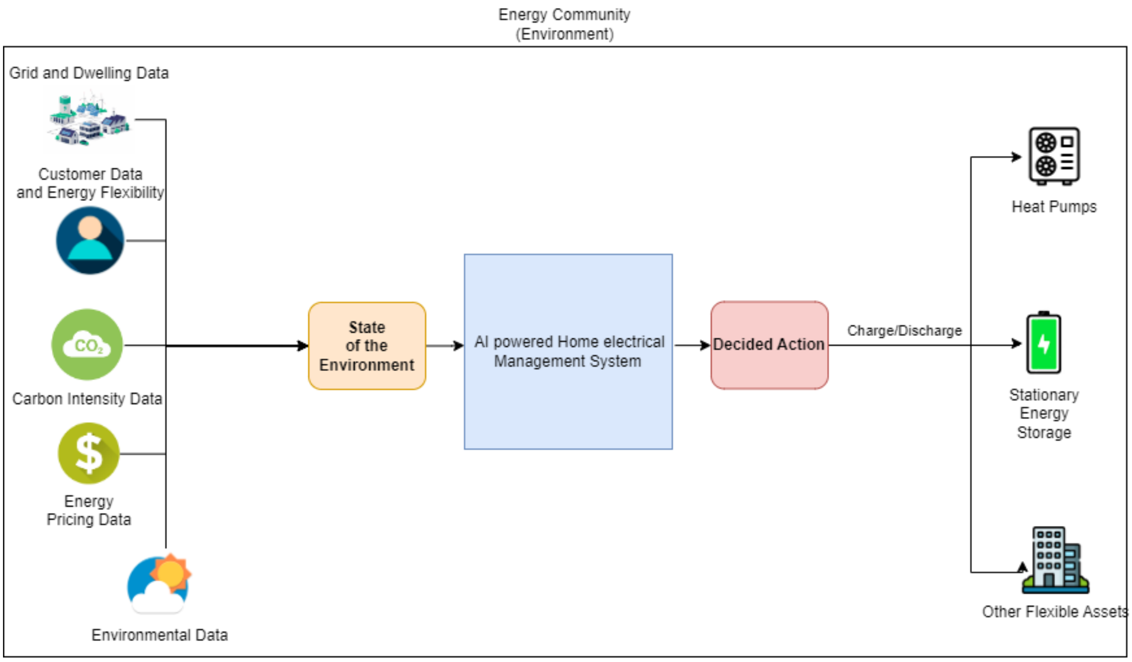


Fig 7 – individual building Diagram

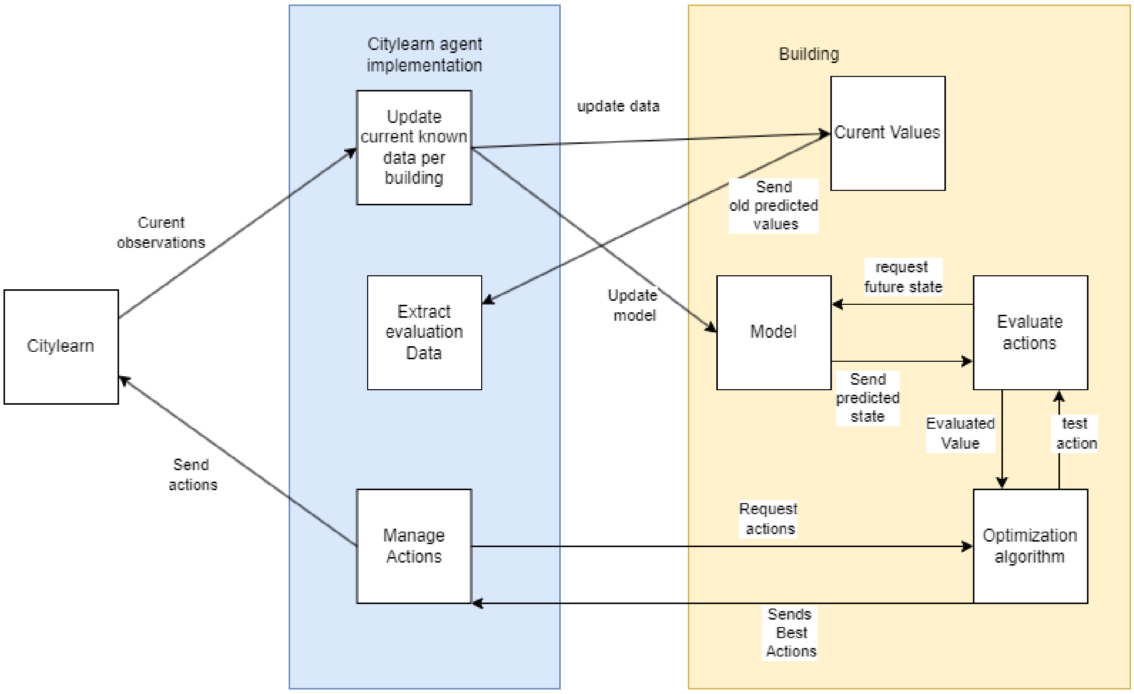


Fig 8 - System architecture diagram

As shown in fig 7 the system architecture starts with citylearn sending the current known observations to the agent implementation, the agent then updates the current known data of each building, and the information relative to the model evaluation is extracted, then it is requested to each building the set of actions most optimal for them, this in turn starts the main building loop of where the optimization algorithm requests to evaluate a set of actions that are then requested to the model the predicted effects though each time step in this paper the system will predict 3 hours into the future, the optimization algorithm receives a number relative to how "good" the solution is, that in this paper calculated through the following equation:

$$\sum_{n=1}^{3}(1/4)(P1(n-1)+P1(n))$$

Where P1 is the net electricity consumption prediction given by the model and n is the time step.

The set of actions of each building is then sent to citylearn and the cycle repeats. After this introduction now we present a more in depth investigate the system.

The main part of the system consists in creating and managing the environment, first files pertaining to a previous run are eliminated and the needed directories are created, a file that contains the next parameter to test is loaded, the first line is removed and the system starts to create the CityLearn environment the environment contains the chosen dataset that in this case is the citylearn_challenge_2021, we are not going to delve into the dataset here further information about the dataset can be found in section 5.1.2, and the central agent flag is set to false this flag if set to true aggregates all the buildings into one list as the flag is set to false the building observations are separated per each building, and in case there is the need to finish earlier the simulation end time is set with the desirable value, this was used in the case studies 1 to 4 where more information can be found in section 5.3. After setting up the citylearn environment an agent object is created and is run. After the agent finalizes the simulation, the information contained in the run files are summarized according to the KPIs needed see section 5.2 and some graphs are generated using matplotlib's pyplot, for the case study 5 the KPis pertaining to the CityLearn framework are also saved. During the tests of the environment this would loop again to get the next test value.

The customized agent class that was implemented for this thesis has 2 large functions the predict that is called by the CityLearn framework and the fill actions, this latter exists to ease multithreading implementation. The predict function first starts by checking if there is already a building list if there is none one is created, and each building object is filled with basic information, the building information is later in this chapter, if the building list is already created than it runs the building function that allows to fit the model. The data required to retrieve at each time step is saved to a directory in individual files to later be summarized, after each building observations are updated, and the fill actions object is called for each building, after each building has finalized getting their actions the action variable is set with the new values and the CityLearn function to start the next time step is called. The only difference between the case studies 1 to 4 and 5 to 7 is that since it only needed one building, although all buildings were created, only one was being monitored with the others having random action variables.

Continuing in the agent the fill actions function as the name suggests handles in filling the actions of each building individually, it creates an object of the optimization function and sets the main variables like the building that will optimize, the action names, the iterations and the particle number, after it creates the array that contains the upper and lower bound of each action, after it calls the optimization function that returns the actions and the predicted values, the predicted values are saved on a building variable to latter be used to extract the data needed in the case studies, if the current step is under 2000 it calls the simple RBC code provided by CityLearn, after the 2000th time step the values found by the optimization function is set in the actions array. This concludes the logic that the agent implementation has.

The building class is a layer of abstraction, it saves information relative to each building like the name, the action names, the observation names and created the model. The only function of note is the fit function that runs the logic to be able and fits the model.

The MPC needs an optimization algorithm, that here is its own class that has the optimize and calculate objective functions, the optimize function has the logic pertaining to the optimization algorithm that in this case is the particle swarm algorithm that is further explored in 4.3.2.1. The calculate objective function as the name suggests calculates how good a solution is by calculating it following the objective function described above, it does this by calling the predict function of the model inside the building, now this only gives the prediction of the impacts on one set of actions, this impact is then translated into a new input of the neural network, that now can run again with another set of actions this gives the prediction to T+2, this cycle of getting the prediction, creating the input and getting the prediction for the next time step is iterated for each set of actions, after getting all of the predictions the value of how optimal this set of actions is, is calculated and returned.

With the main agent, building and optimization classes explained that only leaves the model class to explain the behavior and implementation. The model class has the init like the others where all of the information relative to the model is done, in the case of the models that used the neural networks this meant that it was here that the neural network was created, it had a input layer followed by the definition of all middle layers, and ended with the output layer, since this function receives the layer information the number of units in a layer was set here as the learning rate of the network was also set, this object is then stored to be used latter. In the case of the ensemble methods the same happened here, with different parameters this being what ensemble methods to use like the number of estimators or the learning rate in case it has one , in the case of the regressions the same happened but to a list that has all of the objects generated this allows later to use a simple average to try to achieve a better result. For the Case study 5,6 and 7 this was switched to a hard coded model, it used the same methods just not receiving the information from the main system or a file.

For the neural network implementation, the fit and predict functions handled the information it received and set it in a way that it can be inputted into the networks, this was mostly handling arrays and changing the odd value to float, after that it would call the fit or predict function of the TensorFlow object.

The ensemble implementation differs from the neural network since there is more processing, the input data processing is still present now more simplified, but each function had differences, the predict function had the ability to get multiple predictions from multiple regressors and do a simple average, this not being something that makes sense in the previous neural network implementation, the fit behaves similarly to the predicts where it can fit multiple regressors at the same time, the fit function deviates from the neural network implementation by only fitting every 2000 iterations, this was done since most regression models cannot be partially fitted and neural networks can get new data added without a large computational loss, another point to make the regressors fit every 2000h is that opposed to the neural networks the regression models had to be fitted form scratch this was computationally heavy and takes some time even with the training multithreading implemented where multiple models could be trained at the same time it still took too long to be able to get the end results needed for this thesis in a timely manner.

4.2 Methods used

A model Predictive control needs multiple systems, there is a need for a model algorithm to evaluate the impact of the actions into the future and an optimization algorithm. These models can be created by using previous knowledge of the environment to create a mathematical model or like in this case use machine learning techniques to get the same, the impacts of the actions on the future of the environment where they are enacted. As there is a need for a model, a way to optimize the search of the best action or set of actions to take is also needed where for this system the particle swarm optimization was chosen.

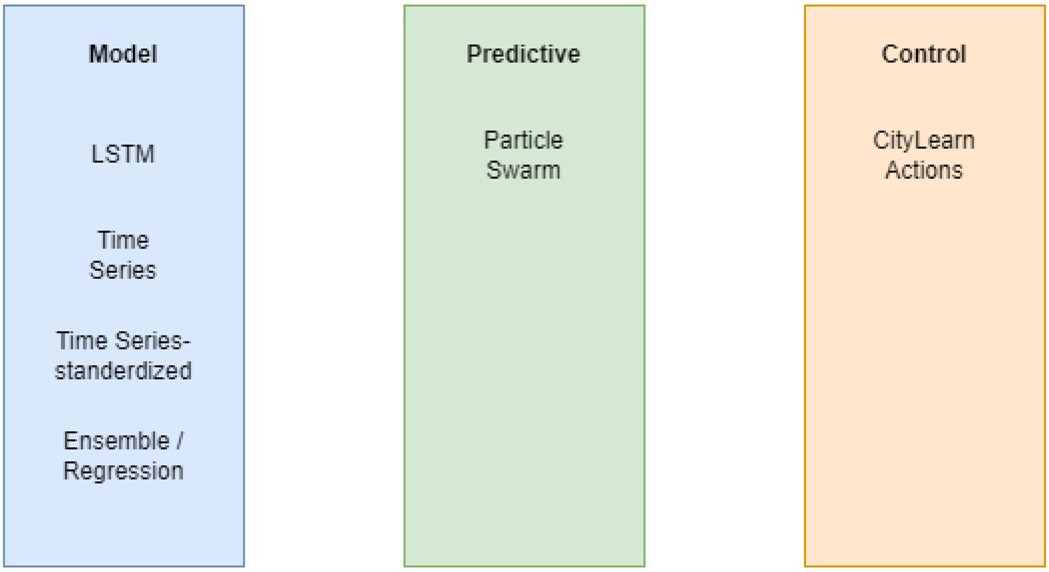


Fig 9 – Different types of methods used relative to where they act.

As it is possible to see in the above picture there were different needs for different parts of the MPC implementation it was tested 4 different models to be used as the MPC model, one optimization system to predict what is the best path to take, there was no need to implement a control system since CityLearn already has one implemented on every agent.

4.2.1 Model algorithms

In this section the different algorithms used as the model in this implementation of a house management MPC are described, the LSTM and timeseries implementations the TensorFlow framework was used to ease the development, while for the ensemble and regression models the framework scikit-learn was used, and additional custom-made logic to manage this last model was also made.

4.2.1.1 LSTM

This first implementation was based on a single input and multiple output model

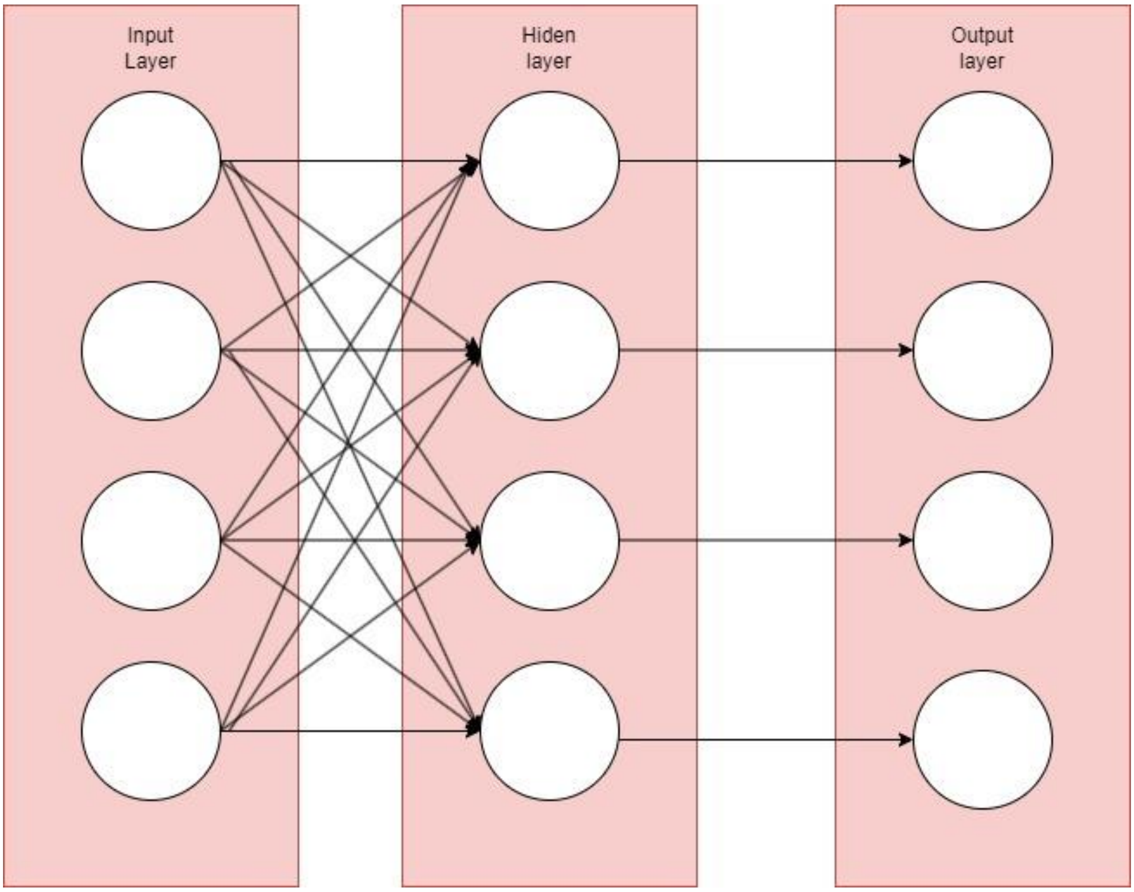


Fig 10 – A example of a multiple input multiple output neural network logic used here.

Although the above image shows how the LSTM network was designed where only the input layer connects to a hidden layer and the hidden layer has a direct non crossing connection to the output layer this allows to change the topology of a specific prediction without changing the topology of other predictions this makes the implementation a lot more responsive to future need simply because since all predictions use the same input, to add a new prediction all that is needed is to append to the network a new arm, there is no need to retrain the other arms, this also allows for different number of neurons and different number of layers to better suit each prediction needs.

Because this was the first implementation it only predicted the state of charge of the different batteries and the internal temperature of the house, the inputs for this model are the current observations plus the actions to take.

4.2.1.2 Timeseries

This implementation was a multiple output system. That predicted all observations that a building could have, as shown in image 10 this model heavily relies on a large, interconnected network and not the individual one that appear in the LSTM implementation seen prior.

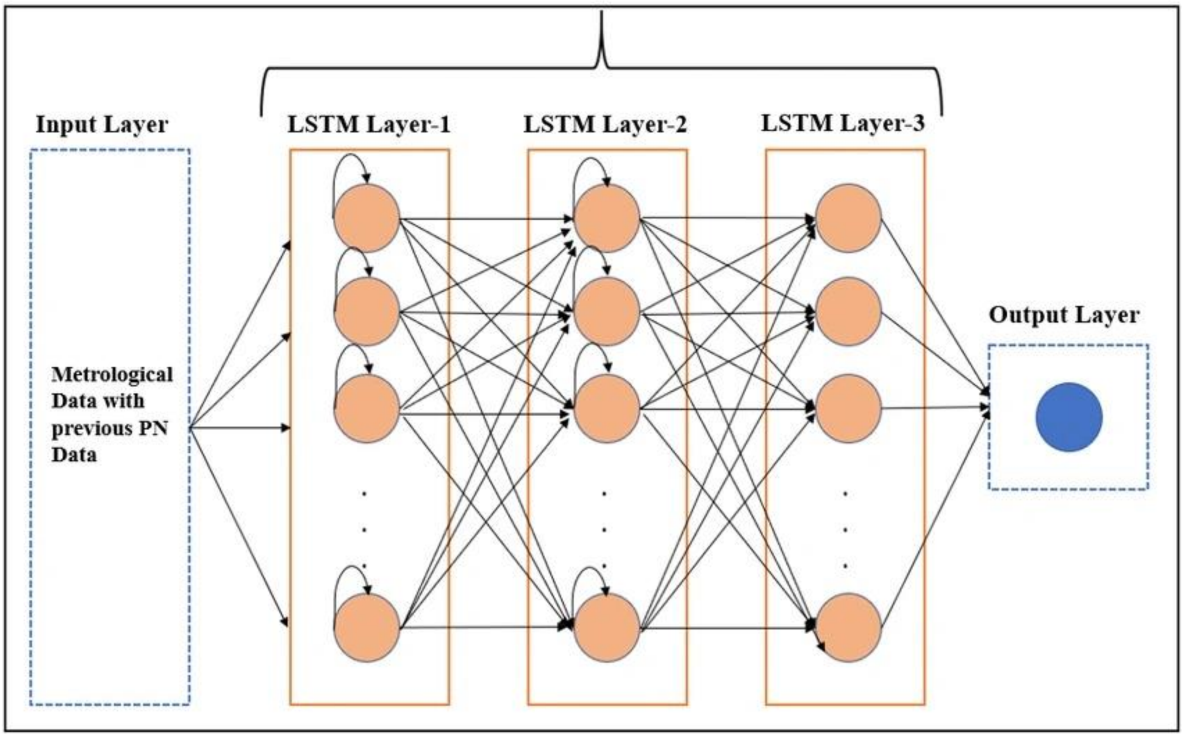
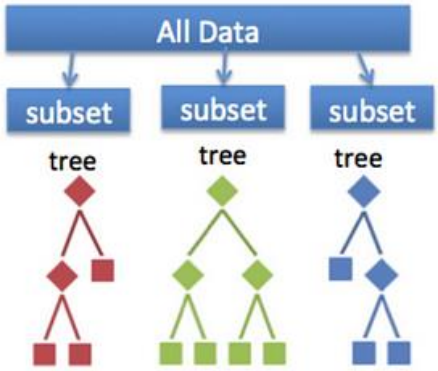


Fig 11 - LSTM base Timeseries model architecture [49]

This has the disadvantage of being less easy to optimize and to react to future changes. The timeseries implementation is also a lot more memory intensive than both the LSTM and ensemble methods. This method was also implemented a standardization of the input values. The input values of this method were the same as before with the difference of the historical values of the last 100 days also present in the input.

4.2.1.3 **Ensemble/regression algorithms**

Ensemble methods work by using multiple estimators. An ensemble method will use a set number of estimators, those factors can be fully used or split into different subsets of the original information, depending on the method chosen, that in the end give a prediction value, based on the estimations from the estimators as shown in image 11.



A random forest takes a random subset of features from the data, and creates n random trees from each subset. Trees are aggregated together at end.

Fig 12 - Example of a Random Forest with 3 decision trees [50]

The regression algorithms work much simpler, to better understand the image below shows a linear Regression algorithm.

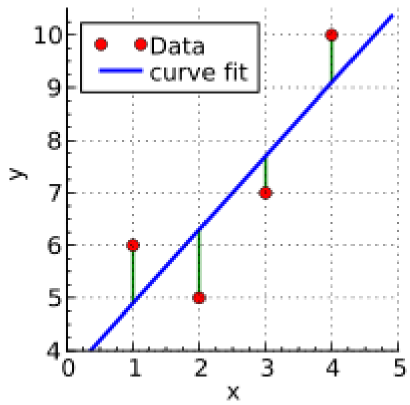


Fig 13 – linear model example [51]

A line function can be simplified to $y = \beta x + C$ where β is the slope, and c is the offset on X=0 of the line from the image above it can be noted that the linear regression cannot approximate a curve this is one of the problems of a linear regression while other methods might, depends on the implementation but the general is always have a function that tries to be as accurate as possible to the training data [52].

4.2.2 Optimization algorithm

In this section the optimization algorithm used on the CS5 implementation is described, this algorithm should optimize the action values.

4.2.2.1 Particle Swarm

The particle swarm algorithm is a meta-heuristic optimization algorithm that is inspired by swarm behavior. It is based on having multiple points that by using the knowledge provided by other points try and search the best option [53].

The way it works is by creating random particles with random velocities at the first-time step, then evaluating and saving the current best. After that iteration it alters the velocity of the neighboring particles to adjust to the best neighboring particles, this process of evaluating the current particles and updating the speed is then done again until a sufficient solution is found [54].

This method was chosen due to its advantages:

- It is derivative free [53], since the user can switch its optimization function at will calculating derivatives can become a problematic situation.
- It is a very efficient global search algorithm [53], since it searches the solution in a large optimization space.
- It is easily parallelized [53], this is a large point since the system needs to search for an optimal solution in a large number of variables, and the way the system is engineered having the option to run multiple searches at the same time is a large advantage.

5 KPIs, Dataset Information and Test cases

5.1 Simulation information

5.1.1 Simulation Environment

The CityLearn environment allows a stable environment to compare different solutions on energy management, the environment allows for centralized or decentralized systems, in this case the decentralized (one per house) system was chosen, the dataset chosen was also provided by citylearn - the 2021 challenge dataset. The citylearn framework already provides a set of KPIs for algorithm evaluation this KPIs were used in CS5, defined in Section 5.2.2 [14].

Since this thesis proposes a decentralized system citylearn proved the best solution to this problem, it handles the simulation environment and has multiple implemented systems, another point was the ease of use with the most basic system being just a standard implementation of an agent, care needs to be taken to make sure that the agent is following the correct implementation since there is the possibility of having a decentralized agent behaving in a centralized fashion, but that high freedom of implementation dose allow for more complex systems like the one presented here. The entire MPC module, with some changes to the internal model behavior like taking into account other predictions that have since been implemented, simply be published online and everyone can use it and check it against another citylearn dataset or their own, CityLearn is also heavily documented and that online documentation helps with figuring out what some less instinctive values are like the actions behavior or how the SOC value is handled.

5.1.2 Simulation scenario

Derives from the dataset described in [55] that features 9 buildings, a medium-sized office, a fast-food restaurant, a standalone retail store, a strip mall and five medium-scale multi-family residences. It has details about air-to-water heat pumps, electric heaters for Domestic Hot Water, and on-site solar panels.

The buildings are referred by their numbers, as presented in Table 1.

Table 1 – Building description to name list.

Building type	Building name
medium-sized office	Building 1
fast-food restaurant	Building 2
standalone retail store	Building 3
strip mall	Building 4
medium-scale multi-family residences	Building 5 to 9

Every building contains the information described in Table 2.

Table 2 – Data set observation listing [14]

Name	Descriptions	Units
month	Month of year ranging from 1 (January) through 12 (December)	
day_type	Day of week ranging from 1 (Monday) through 7 (Sunday)	
hour	Hour of day ranging from 1 to 24	
carbon_intensity	Grid carbon emission rate	kgCO2/kWh
cooling_storage_soc	State of the charge (SOC) of the *cooling_storage* from 0 (no energy stored) to 1 (at full capacity)	kWh/kWhcapacity
dhw_storage_soc	State of the charge (SOC) of the *dhw_storage* (domestic hot water storage) from 0 (no energy stored) to 1 (at full capacity)	kWh/kWhcapacity
electrical_storage_soc	State of the charge (SOC) of the *electrical_storage* from 0 (no energy stored) to 1 (at full capacity)	kWh/kWhcapacity
diffuse_solar_irradiance	Diffuse solar irradiance	W/m2
diffuse_solar_irradiance_predicted_6h	Diffuse solar irradiance predicted 6 hours ahead	W/m2
diffuse_solar_irradiance_predicted_12h	Diffuse solar irradiance predicted 12 hours ahead	W/m2
diffuse_solar_irradiance_predicted_24h	Diffuse solar irradiance predicted 24 hours ahead	W/m2
direct_solar_irradiance	Direct solar irradiance	W/m2

direct_solar_irradiance_predicted_6h	Direct solar irradiance predicted 6 hours ahead	W/m2
direct_solar_irradiance_predicted_12h	Direct solar irradiance predicted 12 hours ahead	W/m2
direct_solar_irradiance_predicted_24h	Direct solar irradiance predicted 24 hours ahead	W/m2
indoor_dry_bulb_temperature	Zone volume-weighted average building dry bulb temperature	ºC
indoor_relative_humidity	Zone volume-weighted average building relative humidity.	%
net_electricity_consumption	Total building electricity consumption	kWh
non_shiftable_load	Total building non-shiftable plug and equipment loads	kWh
outdoor_dry_bulb_temperature	Outdoor dry bulb temperature	ºC
outdoor_dry_bulb_temperature_predicted_6h	Outdoor dry bulb temperature predicted 6 hours ahead	ºC
outdoor_dry_bulb_temperature_predicted_12h	Outdoor dry bulb temperature predicted 12 hours ahead	ºC
outdoor_dry_bulb_temperature_predicted_24h	Outdoor dry bulb temperature predicted 24 hours ahead	ºC
outdoor_relative_humidity	Outdoor relative humidity	%
outdoor_relative_humidity_predicted_6h	Outdoor relative humidity predicted 6 hours ahead	%
outdoor_relative_humidity_predicted_12h	Outdoor dry bulb temperature predicted 12 hours ahead	%
outdoor_relative_humidity_predicted_24h	Outdoor dry bulb temperature predicted 24 hours ahead	%
solar_generation	PV electricity generation	kWh

The actions we can take are cooling_storage, dhw_storage and electrical_storage each number can change from -1.0 to 1.0 and it correlates directly to how much that specific system is to be charged and discharged for example:

- The electricity storage system is currently at 0.5 charge, if the associated action value is 0.5 then the value of the storage system at the next timestep is 1 and the needed energy was absorbed from the diverse types of sources like photovoltaic panels or the energy grid.

- The electricity storage system is currently at 0.5 charge, if the associated action value is -0.5 then the value of the storage system at the next timestep is 0 and the energy removed form it is used to supply the house/returned to the grid.

The observations above are true for the building 1,2,5,6,7,8 and 9 but since buildings 3 and 4 do not possess a Domestic hot water storage element the associated SOC observation and action is not available, this does not influence the final results.

5.2 Key Performance Indicators

The KPIs is a metric that allows a comparation of multiple different solutions using a standardized system, for this application it was decided to have 2 different systems one for the evaluation of the models and another for the comparison between different solution.

This is due to the metrics that are needed to evaluate change, since evaluating parts of the solution against themselves and the completed solution against other solutions have entire different ideas, since in one we want to check how accurate the model is and in another we want to compare the efficacy of the system in managing a household, these different evaluations have the need for different KPIs.

5.2.1 Models KPI

The KPIs for CS1, CS2, CS3 and CS4 (the model KPIs) what it is needed to evaluate is how accurate the future predictions are, for that effect, a 95th percentile of the absolute difference of the prediction and the real value, the standard deviation and the max value of the same error.

This approach was chosen since it can tell the information that it is needed to know, the objective of the model part in this system is to create the most accurate inference model, where it can find the most accurate prediction, these metrics tells us what the max error and most of the error is located, the latter is to understand if the max error is just a outlier. Let's take an example:

If the 95th percentile error of the prediction of the indoor temperature is 1ºC and the max error is 20ºC, it shows that although the model dose give an accurate prediction within 1ºC for 95% of the time on 5% of the time the error would be 20ºC, comparing this to another where the 95th percentile is 1.5ºC and the max error is 2ºC, although the 95th percentile is higher than the previous example, the maximum error is only 2ºC, although this system does have a higher 95th percentile error it is also more desirable since the maximum error is much lower.

The observations that were used to test the effectiveness of the model were also chosen mostly by what it is possible to change with the actions that the system can make, only the following observations were taken into account, but after CS1 all of the observations a building can have were predicted, but since the system cannot change anything related to those observations, they were not taken into account, although the same process was also done for all observations in CS5.

Since we are only evaluating the accuracy of the prediction that is influenced by our actions only one building was used for that effect that being building 1 since it has all the needed observations and thus all the possible actions we can take, this also allows us to compare a possible generalization of the model that will be evaluated in CS5.

The observations chosen to evaluate the models were:

- cooling_storage_soc : it measures the available capability of the cooling storage from 0 to 1 where 0 is empty and 1 is full.
- dhw_storage_soc : it measures the available capability of the domestic hot water storage from 0 to 1 where 0 is empty and 1 is full.
- electrical_storage_soc : it measures the available capability of the electrical storage from 0 to 1 where 0 is empty and 1 is full.
- indoor_dry_bulb_temperature : it measures the indoor temperature.
- net_electricity_consumption : Total building electricity consumption.

Since this are the actual observations that the system can influence these are the ones that were chosen to evaluate the different models experimented.

5.2.2 Systems KPIs

The systems KPIs are used to compare the different solutions, these KPIs are already part of the citylearn AI gym. The KPIs are for single buildings only, since the system evaluated here dose not possess a central management system.

- Carbon Emissions –This metric measures the amount of carbon emissions each individual building created to get all the needed energy supply.

$$\sum_{hour=0}^{n} (\text{Energy from grid}) * (\text{Carbon Intensity})$$

- electricity consumption – This metric measures the amount of electricity that was originated from the grid, it does not consider other energy sources.

$$\sum_{hour=0}^{n} (Energy\ consumeed) - ((energy\ from\ solar\ panes) + (Energy\ from\ eleticity\ SOC))$$

- zero_net_energy – These metric measures all the electricity consumed from all sources.

$$\sum_{hour=0}^{n} (Energy\ consummed)$$

For all these metrics the lowest number the better the solution is. Of note all other metrics that are building based did not see any changes between models.

5.3 Case Studies

Through the research it was felt the need to further explore how the different types of models would behave those are the case studies 1 through to 4 where multiple different types of models were tested with multiple different model types were tested with varying degrees of success and different methods to better create a base line to choose the proposed model to the 5th case study where the system proposed is tested against other systems to assert if the proposed system can compare to other systems, the full list of case studies with a small description, usage and type can be seen in table 3.

The case studies 1 to 4 were isolated from each other to better assess each performance individually this allows for a simpler understanding of the strong points of each model type without removing the analysis of other models.

The final case studies, case study 5 to 7, then compares the performance and assesses if the model can or not be generalized to multiple other buildings if they have similar inputs, this can then influence future iterations of this system. All case studies used the same simulation scenario described in section 5.1.2.

Table 3 – Case study list

Case study number	Description	Usage	Type
CS1	First attempt at creating a neural network model for prediction of the effects of an action on future observations	model	LSTM neural network
CS2	Changed the model from CS1 the LSTM model a time series model on the prediction	model	Time Series
CS3	Studying the effects of standardization of input parameters in the time series model for this type of problem	model	Time Series Standardized
CS4	Testing ensemble / regression models on the prediction of the effects of an action on future observations	model	Ensemble/regression models
CS5	Test the best model with a Particle Swarm algorithm against other systems on running all the buildings	Full system	Ensemble
CS6	Using previous data from CS5 infer if it is possible to generalize the model	Model generalization	Ensemble

CS7	Model accuracy evolution over time using data obtained in CS5	Model conversion	Ensemble

6 Results

6.1 Case studies

In this chapter, we will evaluate the results of the different models that were tried, it also compares the current system to a base line and the SACRBC both available through CityLearn and the EnergAIze algorithm, further studies to evaluate the convergence, generalizability and accuracy over time of the chosen model for the evaluation against other system were also made.

6.1.1 CS1 – LSTM network

As discussed in 5.2.1 the first attempt was to use an LSTM network to predict the results of t+1. In this first case the LSTM was configured with an input layer, one middle layer that had a configurable parameter and a dense output layer. This case study uses the dataset defined in 5.1.2 using only the first building to analyze the results. This LSTM was configured with the parameter presented in Table 4.

Table 4 – Case study 1 parameter table

Parameter	Range	Additional information
Learning Rate	0,0000001 to 0.1	10x increments
Number of units	5 to 40	5 units in increment
Training Time	3000-time steps	First 1000 ignored

Table 5 – Case study 1 KPI information

observation	95th percentile	Maximum error	Standard Deviation	Learning rate	Units in layer
colling storage SOC	+-23%	+-141%	10%	0.1	25
DHW storage SOC	+-47%	+-111%	15%	0.01	45
Electrical storage SOC	+-71%	+-79%	25%	0.01	45
Indoor temperature	+-0.59ºC	+-5.28ºC	0,25ºC	0.1	5
Net Electricity consumption	+-93.17 kWh	+-157.51 kWh	33,18 kWh	0.1	5

From Table 5 it is possible to extrapolate that the error prediction margins are too big to be an effective model, since 2 of the 3 batteries exhibit SOCs error superior to 100%, meaning that this system is predicting values well above or below any number that those observations might have. The same applies to the indoor temperature, which has an error bigger than 5ºC. The max error of the Net Electricity consumption metric is also too big with highs on the 157.51KWh of error.

The performance of this type of network was deemed not viable and by the time further testing was ready the call to switch to a time series approach was already taken, with errors in predicting the charge lever of a battery with a high level of error sometimes above 100% in error, the author also recognizes that an even earlier implementation without large experimentation did improve on the max error by simply limiting the max prediction by using the sigmoid activation function on the final layer but it did not improve the middle results.

6.1.2 CS2 – Time series – non standardized

As discussed before in 5.2.1 the second attempt was to use an LSTM timeseries network to predict the results to t+1, in this second case was a simple grid search with the input layer, three LSTM configurable layers one dense configurable layer and a dense output layer. This training was cut short because of the implementation of the standardized version of this algorithm. This case study used the dataset defined in 5.1.2 using only the first building to analyze the results The parameters of this training can be seen in table 6.

Table 6 – Case study 2 parameter table

Parameter	Range	Additional information
Learning Rate	0.0001 to 0.1	10x increments
Number of units	65 to 267	64 units in increment
Training Time	6000-time steps	First 2000 ignored

Table 7 - Case study 2 KPI information

observation	95th percentile	Maximum error	Standard Deviation	Learning rate	Units in layer
colling storage SOC	+-84%	+-220%	29%	0.001	[129, 257, 65, 65]
DHW storage SOC	+-86%	+-368%	32%	0.001	[65, 129, 65, 65]
Electrical storage SOC	+-86%	+-354%	32%	0.001	[129, 257, 65, 257]
Indoor temperature	+-8.17ºC	+-11.03ºC	6.97	0.0001	[65, 65, 65, 193]
Net Electricity consumption	+-111,89 kWh	+-219,93 kWh	35,04 kWh	0.001	[129, 193, 65, 257]

As shown in table 7 this model proved unusable for the final project, these results were very bad, and it was quickly moved to a standardized time series model. The SOC error is too high to even consider for use since CS1 had better predictions all around and even it is worse prediction on the SOCs being a maximum error of +-1.41 while the max error of this is +-2.2 at the best possible case. The indoor temperature is also bad with a 95th percentile in the +-8.17ºC in error which in this application is unusable. The net electricity consumption error is still high with +-219.9KWh in max error.

What was presented here does not mean that this type of model should not have further research as explained due to limitations in testing capability this models' tests could not be finalized since the initial results were not satisfactory that the switch to a standardized input was made.

6.1.3 CS3 – Time series – standardized

The third attempt was to use an LSTM timeseries network like the previous case but standardized to predict the results to t+1, in this third case was a simple grid search with the input layer, three LSTM configurable layers one dense configurable layer and a dense output layer. This training was cut short because of the Ensemble/regression models implementation. Due to an error on the logic of how the data is handled only the batteries have the correct data to present. This case study used the dataset defined in 5.1.2 using only the first building to analyze the results The parameters used can be seen in table 8.

Table 8 – Case study 3 parameter table

Parameter	Range	Additional information
Learning Rate	0.0001 to 0.1	10x increments
Number of units	65 to 267	64 units in increment
Training Time	6000-time steps	First 2000 ignored

Table 9 – Case study 3 prediction KPI table

observation	95th percentile	Maximum error	Standard Deviation	Learning rate	Units in layer
colling storage SOC	+-53%	+-60%	15%	0.0001	[193, 129, 193, 129]
DHW storage SOC	+-56%	+-68%	14%	0.0001	[129, 193, 129, 257]
Electrical storage SOC	+-57%	+-65%	17%	0.0001	[193. 257. 129. 65]

This case was not fully tested because after the bad early results another call was made to switch to ensemble and regression systems it did show some promise and further research should be done whether that is by increasing the number of layers or running the full battery of tests, both time series implementation had the problem that the system that was running the implementation could not be ran 24/7, the test case dose show promise.

As shown in table 9 using standardized values did improve the prediction but it was still not the best, there would be a loss of the granularity that would happen in the case of the Net Electricity consumption where the data returned had a very different meaning to the other values returned on the other case studies since the theoretical range would be from -infinite to +infinite.

6.1.4 CS4 – Ensemble / regression systems

The fourth attempt was to use an ensemble / regression system it went back to only having 1h of prediction time like CS1 and behaving similarly to CS1 in modularity terms.

The systems tested alone can be seen in table 10 with additional information when needed:

Table 10 – Regressor parameters list

Regressor	**Learning rate (if applicable, if not present then it is the default value)**	**Number of estimators**
RandomForestRegressor		100 to 300 step 100
AdaBoostRegressor	1.0	100 to 300 step 100
BaggingRegressor		100 to 300 step 100
ExtraTreesRegressor		100 to 300 step 100
GradientBoostingRegressor	0.1	100 to 300 step 100
HistGradientBoostingRegressor	0.1	

The other regression methods tested were:

LinearRegression, RidgeCV, ElasticNet, Lasso, ARDRegression, BayesianRidge, RANSACRegressor, PassiveAggressiveRegressor. All these regressors were also tested with all possible combinations between them by using the average of the sum of the results.

It ran for 6000 iterations ignoring the first 2000 as training time. This case study used the dataset defined in 5.1.2 using only the first building to analyze the results

Table 11 – Case study 4 prediction KPI table

observation	95th percentile	Maximum error	Standard Deviation	regressor
colling storage SOC	+-17%	+-36%	5%	ARDRegression
DHW storage SOC	+-17%	+-27%	5%	ARDRegression
Electrical storage SOC	+-42%	+-54%	12%	LinearRegression, RidgeCV, ARDRegression
Indoor temperature	+-0.18 ºC	+-0.55 ºC	0.05ºC	ARDRegression
Net Electricity consumption	+-112,59 kWh	+-237,08 kWh	35,62 kWh	ElasticNet

This was the final model type that was tested, as shown in table 11 although the performance of the Net Electricity consumption value was the worse that it was tested, this type of model was the one that was chosen to be taken into the last test case mostly due to the battery and indoor temperature performance, since it had the best results in 4 out of the 5 predictions this model would be used in the final test.

6.1.5 CS1 to CS4 Resume

The Case studies 1 to 4 refer to the tests to access the effectiveness of different models to use, since the ensemble models were clearly the best in the SOC and indoor temperature prediction the ensemble and regression methods were chosen, the best Net Electricity consumption comes from the first try of a simple LSTM network, but due to the bad performance in the SOC prediction the choice of the ensemble and regression was chosen.

To the effects of this thesis the choice for a single model type was made to streamline the development and test of the final solution, although the system was designed with modularity in mind, the simplicity of using only one model type and the reduction of problems in the final implementation saw that only ensemble and regression models were used.

6.1.6 CS5 – Full system test and comparation to other systems

As was discussed before the system ran the first 2000h using a rules system and after those 2000h the system would start to run the optimization system.

The prediction parameters and the regressor used was chosen the same through the same KPIs used to evaluate each regressor a full list of the parameter and the regressor used can be seen bellow on table 12. This case study used the dataset defined in 5.1.2 using all the buildings that are provided by the dataset to analyze the results.

Table 12 – Case study 5 prediction parameter and the regressor used.

Prediction parameter	Regressor used
carbon_intensity	ARDRegression
dhw_storage_soc	ARDRegression
diffuse_solar_irradiance	RidgeCV, ElasticNet, ARDRegression, PassiveAggressiveRegressor
diffuse_solar_irradiance_predicted_6h	LinearRegression, Lasso, ARDRegression, PassiveAggressiveRegressor
diffuse_solar_irradiance_predicted_12h	ElasticNet
diffuse_solar_irradiance_predicted_24h	Lasso
direct_solar_irradiance	LinearRegression, BayesianRidge, PassiveAggressiveRegressor
direct_solar_irradiance_predicted_6h	RidgeCV, BayesianRidge, PassiveAggressiveRegressor
direct_solar_irradiance_predicted_12h	LinearRegression, ElasticNet, Lasso, BayesianRidge
direct_solar_irradiance_predicted_24h	LinearRegression, ElasticNet, Lasso, BayesianRidge
electrical_storage_soc	LinearRegression, RidgeCV, ARDRegression
indoor_dry_bulb_temperature	ARDRegression
indoor_relative_humidity	Lasso
net_electricity_consumption	ElasticNet
non_shiftable_load	LinearRegression, ElasticNet, Lasso, ARDRegression, BayesianRidge, RANSACRegressor
outdoor_dry_bulb_temperature	Lasso, ARDRegression
outdoor_dry_bulb_temperature_predicted_6h	Lasso, ARDRegression
outdoor_dry_bulb_temperature_predicted_12h	Lasso, ARDRegression
outdoor_dry_bulb_temperature_predicted_24h	ARDRegression
outdoor_relative_humidity	ARDRegression
outdoor_relative_humidity_predicted_6h	LinearRegression, Lasso, ARDRegression, PassiveAggressiveRegressor
outdoor_relative_humidity_predicted_12h	ARDRegression
outdoor_relative_humidity_predicted_24h	RidgeCV, Lasso, ARDRegression, BayesianRidge, PassiveAggressiveRegressor
solar_generation	RidgeCV, Lasso, ARDRegression, BayesianRidge, PassiveAggressiveRegressor

The main 3 values that we are going to compare as specified will be the carbon_emissions_total, electricity_consumption_total and the zero_net_energy for the SACRBC and a baseline system given by the citylearn framework, and the system described on this thesis (HCMCAI).

Color red is the worse on that building, yellow is the second worse, blue is the second best and green is the best of that building.

Table 13 - carbon_emissions_total KPI evaluation per system

System/ Building	BaseLine	SACRBC	HCMCAI	EnergAIze
Building 1	1.22135	1.21826	1.09226	1.00311
Building 2	1.16807	1.16508	1.06304	1.01673
Building 3	1.08856	1.08731	1.04989	1.00012
Building 4	1.39485	1.38985	1.17763	1.04160
Building 5	1.12423	1.12338	1.07219	1.01806
Building 6	1.06824	1.06872	1.06281	1.05141
Building 7	1.04387	1.04413	1.04150	1.03259
Building 8	1.07685	1.07638	1.07650	1.07021
Building 9	1.06937	1.06888	1.06204	1.01101

Table 14 - electricity_consumption_total KPI evaluation per system

System/ Building	BaseLine	SACRBC	HCMCAI	EnergAIze
Building 1	1.23016	1.22716	1.09244	1.00318
Building 2	1.17900	1.17632	1.06454	1.01691
Building 3	1.09751	1.09643	1.05244	1.00011
Building 4	1.42922	1.42224	1.18686	1.04329
Building 5	1.12610	1.12594	1.07175	1.01799
Building 6	1.06890	1.06941	1.06338	1.05207
Building 7	1.04478	1.04495	1.04175	1.03279
Building 8	1.07887	1.07818	1.07756	1.07116
Building 9	1.07123	1.07063	1.06289	1.01100

Table 15 - zero_net_energy KPI evaluation per system

System/ Building	BaseLine	SACRBC	HCMCAI	EnergAIze
Building 1	1.07693	1.07655	1.05367	1.00354
Building 2	1.07614	1.07571	1.04773	1.01683
Building 3	1.03843	1.03842	1.03378	1.00003
Building 4	1.11041	1.10792	1.08865	1.05802
Building 5	1.07860	1.07896	1.05814	1.01826
Building 6	1.06404	1.06424	1.06232	1.05228
Building 7	1.04436	1.04446	1.04159	1.03279
Building 8	1.07559	1.07505	1.07671	1.07116
Building 9	1.06437	1.06411	1.06008	1.01100

From the KPIs used the system had a good performance comparing to the baseline and the SACRBC although not being fully optimal it did greatly reduce the electricity consumption as shown in table 14 and carbon emissions listed in table 13 and increasing the self-reliance of the system by having lower zero net energy consumption as shown in table 15. Although further tests should be carried out this allows the conclusion that the MPC system might be a viable alternative to the more traditionally complex neural network alternatives.

The EnergAIze had the best performance, this can be partially explained due to the algorithms need to pass once through the entire dataset to learn and the second time is when it runs the optimization while all the other algorithms can run at the first pass.

6.1.7 CS 6 – Model Generalization on other buildings

The following tables 16 and 17 present the generalization comparison of the buildings through the KPIs used to choose the model algorithms.

This can give a better idea if the same model can be used for multiple houses or if each house needs it is specific prediction model. This data was obtained by analyzing the data extracted from the case study 5.

Table 16 - 95th percentile error of predictions

Building	colling storage	Domestic hot water storage	Electrical storage	Indoor temperature	Net Electricity consumption
Building 1	78%	27%	102%	0.35 ºC	101.66 kWh
Building 2	49%	15%	71%	1.25 ºC	24.34 kWh
Building 3	40%	X*	62%	0.28 ºC	41.11 kWh
Building 4	65%	X*	33%	0.45 ºC	49.97 kWh
Building 5	27%	42%	146%	0.06 ºC	28.61 kWh
Building 6	166%	16%	32%	0.03 ºC	20.90 kWh
Building 7	43%	23%	93%	0.03 ºC	22.01 kWh
Building 8	25%	2%	69%	0.03 ºC	18.99 kWh
Building 9	76%	3%	164%	0.04 ºC	28.51 kWh

*Domestic hot water storage does not exist for building 3 and 4

Table 17 - Max error of predictions

Building	colling storage	Domestic hot water storage	Electrical storage	Indoor temperature	Net Electricity consumption
Building 1	142%	67%	190%	0.99ºC	253.01 kWh
Building 2	105%	40%	145%	3.54 ºC	70.79 kWh
Building 3	107%	X*	111%	0.64 ºC	96.00 kWh
Building 4	118%	X*	67%	1.10 ºC	112.32 kWh
Building 5	66%	79%	279%	0.12 ºC	72.92 kWh
Building 6	329%	37%	71%	0.07 ºC	56.82 kWh
Building 7	88%	51%	175%	0.07 ºC	68.91 kWh
Building 8	58%	19%	158%	0.82 ºC	54.24 kWh
Building 9	139%	36%	336%	0.12 ºC	70.34 kWh

*Domestic hot water storage does not exist for building 3 and 4

From the original test in CS4 there is a clear reduction in accuracy, the SOCs and a large variance on the accuracy from building to building with some buildings having better prediction than the CS4 and others a lot worse, this means that the system cannot be generalized to other buildings, one metric that improved a large amount was the Net Electricity consumption where although the first building prediction was around the same, the other buildings prediction improved by a large margin, the indoor temperature had a similar performance to the SOC with a large degree of change in between buildings thus this one cannot also be generalized.

6.1.8 CS7 – Model accuracy evolution

In this Case study we will evaluate how the model behaves, and the prediction accuracy changes over time.

This can give a better insight into how the system will behave on the long term and give a clearer idea on how much the system need to be updated to keep a good prediction.

The left side metric represents the prediction error while the bottom metric divides the time that the system worked in 8 equal parts this is the same for all tables. The graphs of the SOCs are the error of prediction in decimal form, the temperature is in ºC, and the net electricity consumption is in kWh. This data was obtained by analyzing the data extracted from the case study 5.

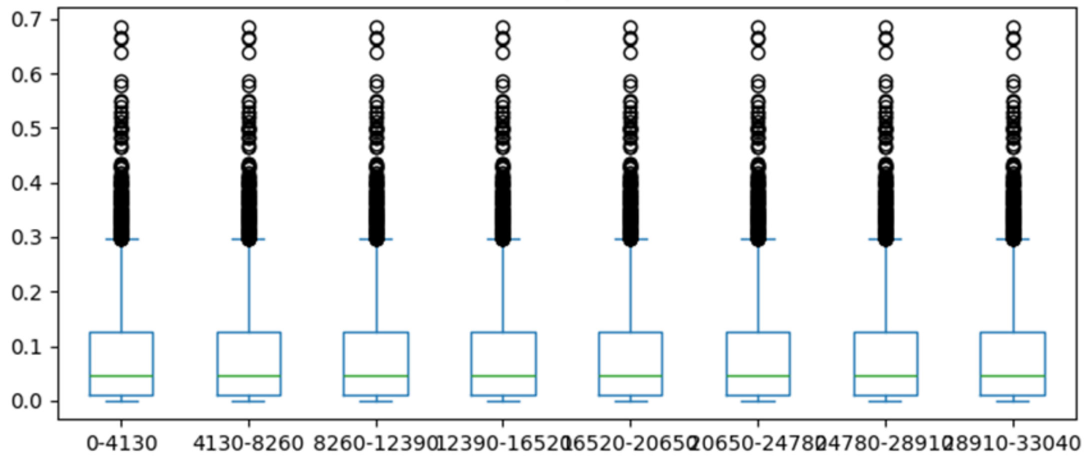


Fig 14 – cooling storage SOC building box plot

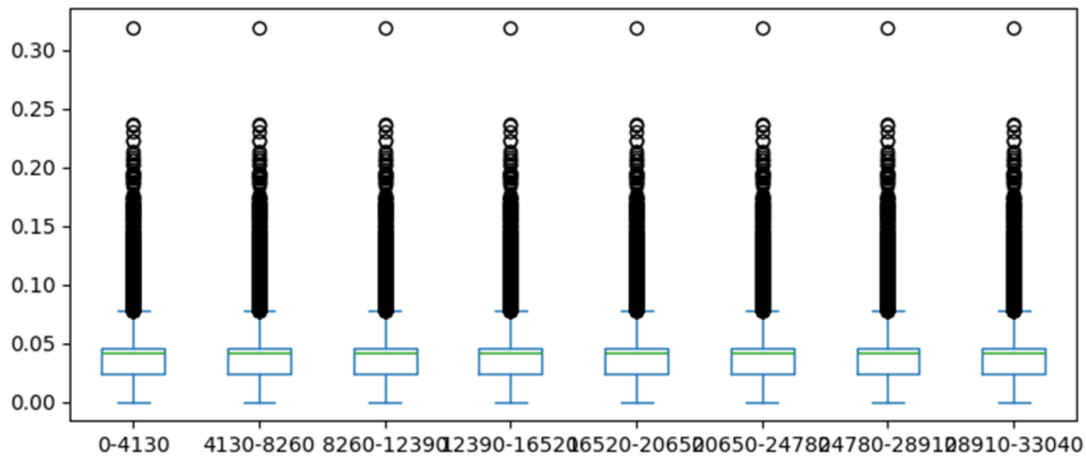


Fig 15 – domestic hot water SOC building box plot

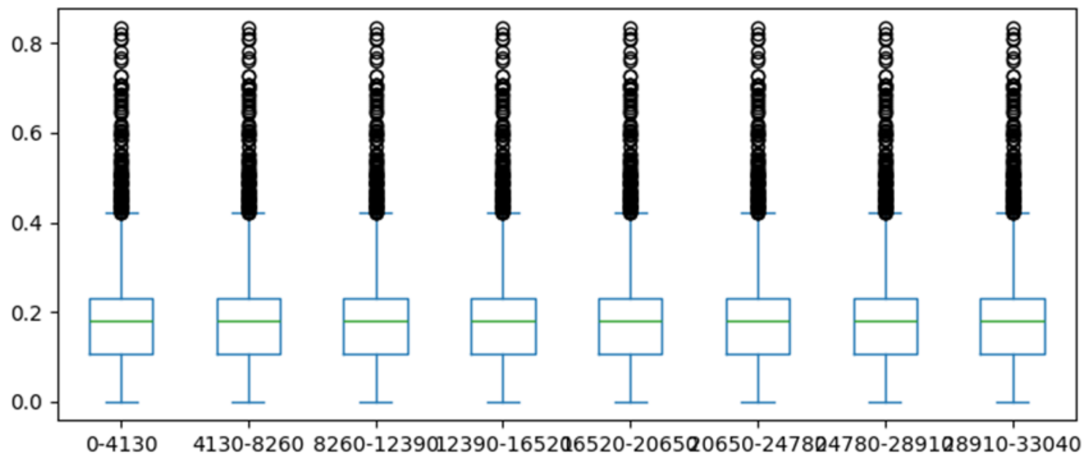


Fig 16 – Electrical storage SOC building box plot

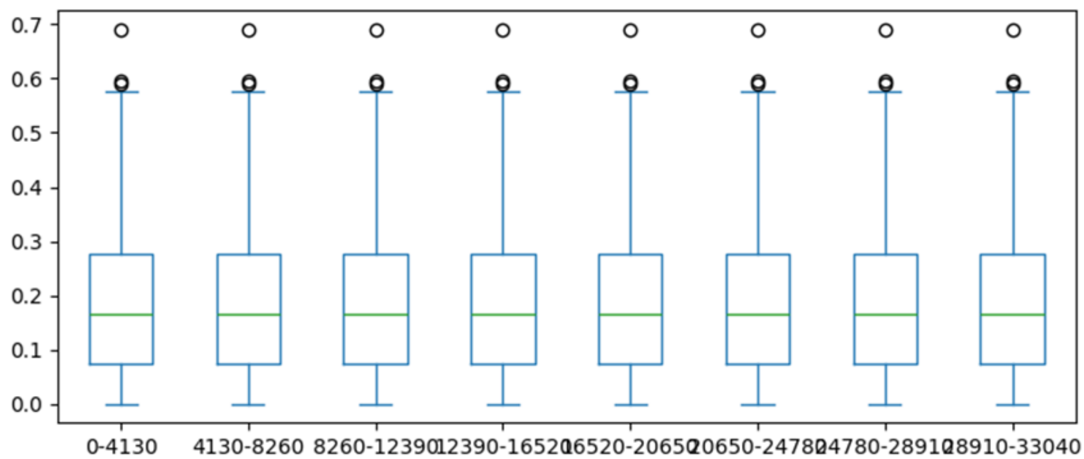


Fig 17 – indoor dry bulb temperature building box plot.

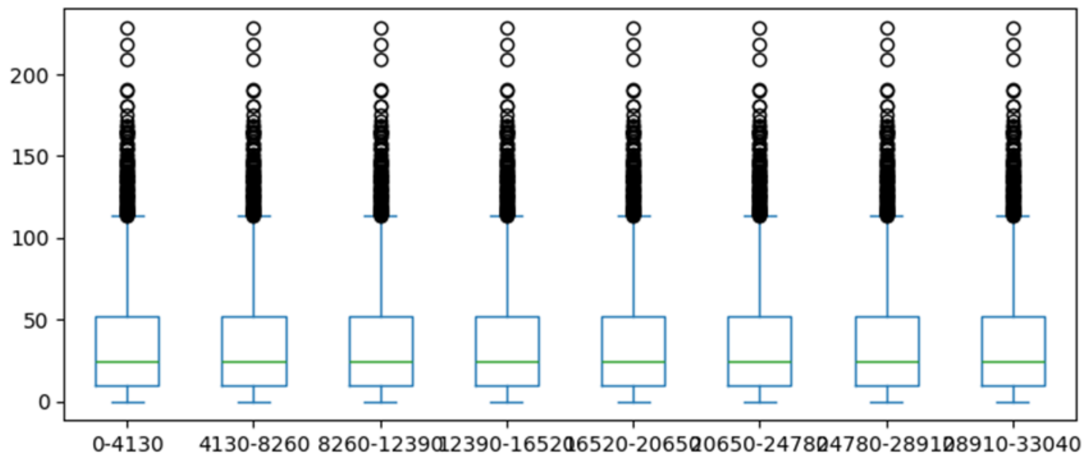


Fig 18 – net electricity consumption building box plot

From Fig 10 to 14 we can see the error of the prediction over time divided in 8 intervals through the data set removing the first 2000 values since the optimization is not yet running and the model is not yet trained those values were removed, through these set of images we can deduce that the model after the initial training, there is not a noticeable increase in getting a better prediction, this shows that the model already achieves the best prediction after a small time frame.

This indicates that the training timings can be increased in this situation, of note that certain observations like the net electricity consumption can be highly volatile, like the addiction of an electric car will change how the prediction is handled, same goes for the indoor temperature, if the user decides do switch the cooling unit that means that the current prediction will lose accuracy, but how to deal with these changes in the environment are not researched on this thesis.

7 Conclusion

As the world inches closer to a full electric future, the need for a system to better manage the energy needs is required, this thesis focused on designing a system to schedule energy consumption of different devices based the usage of Model Predictive Control algorithms to reduce costs, reduce the power consumption from the grid, increase self-consumption and consequently reduce the overall energy footprint. This system aimed to be user controlled, if the user for some reason wants to change the objective of optimization, that can be done without a large training time or a big decrease in quality since it only affects how the actions are optimized and not the model itself that never changes due to user input.

This work presented the results from the simulation of multiple buildings, using the citylearn platform with the 2021 challenge dataset. This dataset contained different types of home systems heat pumps, air conditioning, PV panels, etc.

For the model part of the MPC algorithm we tried multiple models, like a simple LSTM model, two time series models, one with standardized inputs and one without and, finally, it was tested a mix of ensemble and regression methos. The latter was chosen to be the one used for the comparisons due to its better performance on the prediction of the battery SOCs and the indoor temperature.

For the prediction the algorithm to find the best solution a particle swarm algorithm was used since it converges more rapidly into the area of the optimal solution.

The comparation to other systems wielded the expected results, being a better than the simpler models provided by the CityLearn platform those being the BaseLine and the SACRBC, but worse than a fully optimal system like EnergAIze.

The model converged quickly into the best prediction it could make and no improvement was made with the increase in data, the model cannot also be generalized due to high variance between buildings.

7.1 Accomplished Objectives

At the start of the thesis 6 objectives were defined, which we are now going to Analise case by case.

SO1 - Investigate the current state-of-the-art on Renewable Energy Sources, Energy communities and predictive control systems. – this was fully done as the background and state of art chapter.

SO2 - Investigate the current state-of-the-art of energy flexibility management methods of Model Predictive Control and evaluate their applicability. – this again was done at the background and state of art chapter.

SO3 - Design a decentralized multi-agent Model Predictive Control approach that manages and optimizes flexible energy assets considering individual objectives. – this was achieved as we can see that the system was designed to take in consideration the individual objective of a specific building,

SO4 - Test, tune and optimize a solution, use different optimization objectives and EC scenarios, evaluating its performance, scalability, and adaptability to real-world scenarios. – The case studies 1 to 4, 6 and 7 all make this evaluation in the end the regression methods were chosen due to the better performance in the SOC prediction, the models have trouble being generalized and the system converges into it is possible accuracy early in the live span without reducing accuracy with further time passing,

SO5 - Benchmark the proposed solution against other systems. – This was achieved in the case study 6 where this system was compared against a baseline, SACRBC and EnergAIze.

SO6 - Evaluate the tests with regards to environmental, financial and prosumer engagement. This was achieved by optimizing the amount of energy used and reducing the amount of electricity coming from the grid this makes so the amount of pollution lower and the price of electricity also lower, the prosumer engagement is achieved by managing the batteries and photovoltaic panels.

In this thesis all the primary goals were achieved resulting in a system that had the expected performance, unfortunately 2 stretch goals of testing the impact of other optimization systems, and implementing electric vehicles were not achieved this gives space for future work in this area.

7.2 Limitations and Future Work

The current limitations are mostly due to the bad implementation of some modules like the input generation module. This module would need to be reworked to be easier to read, although it can be used as is a simplified and easier to modify version should be implemented. Further work should mostly be focused on 2 main points, testing and improvement of the model and optimization algorithm, and in the implementation of a central system.

A better model should improve on the reliability of the future prediction thus increasing the number of hours the system can optimize which would increase performance. The evaluation of more optimization algorithms is also a good way to increase performance, the author would like to recognize also that this specific use case can make use of a particular part of the optimization algorithms, some work better at optimizing to a general area of the answer space while others work much better at optimizing a small area of search, it is possible to divide the search time into 2 and use 2 different algorithms to find a more optimal solution.

A centralized system, that on this current specification with user freedom and a system that highly tries to obfuscate and reduce the amount of data that is sent to a hypothetical central system due to security concerns, can prove tough and not optimal but a system that implements the 3.1 priority rating is something that will improve the resilience of the electric grid and help in adverse scenarios.

Finally, the implementation and test of the impact of electric vehicles is also a good future work as discussed before electric vehicles are becoming increasingly more common and the owners might want to charge their vehicle at home, this will have an impact on the performance of the MPC system.

8 References

[1] European Commission, “2050 long-term strategy,” [Online]. Available: https://climate.ec.europa.eu/eu-action/climate-strategies-targets/2050-long-term-strategy_en#:~:text=The%20EU%20aims%20to%20be,action%20under%20the%20Paris%20Agreement%20.. [Accessed 26 09 2024].

[2] International Energy Agency, “Smart Grids,” [Online]. Available: https://www.iea.org/energy-system/electricity/smart-grids. [Accessed 10 09 2024].

[3] United States Department of Energy, “The Smart Grid,” [Online]. Available: https://www.smartgrid.gov/the_smart_grid/smart_grid.html. [Accessed 20 08 2024].

[4] T. Fonseca, *Master Thesis A Multi-Agent Reinforcement Learning Approach to Integrate Flexible Assets into Energy Communities,* 2023.

[5] OPEVA, “overview,” [Online]. Available: https://opeva.eu/overview/. [Accessed 12 08 2024].

[6] chint, “How Much Power Loss in Transmission Lines,” [Online]. Available: https://chintglobal.com/blog/how-much-power-loss-in-transmission-lines/. [Accessed 12 09 2024].

[7] S. Lucia and F. Fiedler, “Basics of model predictive control,” DO-MPC, 2023. [Online]. Available: https://www.do-mpc.com/en/latest/theory_mpc.html. [Accessed 20 08 2024].

[8] European Environment Agency, “Renewable energy,” 23 05 2024. [Online]. Available: https://www.eea.europa.eu/en/topics/in-depth/renewable-energy. [Accessed 20 09 2024].

[9] T. Mummert, D. Subramanian, L. Vu and N. Pham, “What is reinforcement learning?,” 14 09 2022. [Online]. Available: https://developer.ibm.com/learningpaths/get-started-automated-ai-for-decision-making-api/what-is-automated-ai-for-decision-making. [Accessed 20 01 2024].

[10] L. Oliveira, “Elementos de Reinforcement Learning,” pilotorobo, 19 January 2018. [Online]. Available: https://medium.com/pilotorobo/elementos-de-reinforcement-learning-cbe45325c6ec. [Accessed 25 09 2024].

[11] J. Drgoňa, J. Arroyo, I. C. Figueroa, D. Blum, K. Arendt, D. Kim, E. P. Ollé, J. Oravec, M. Wetter and D. Vrabie, “All you need to know about model predictive control for buildings,” *Annual Reviews in Control,* vol. 50, pp. 190-232, 2020.

[12] M. Klaučo and M. Kvasnica, “semanticscholar,” [Online]. Available: https://pdfs.semanticscholar.org/a5f5/7548df16c2c44136de3b7f7e8e05d5e0765a.pdf. [Accessed 20 01 2024].

[13] A. Mar, P. Pereira and J. Martins, “Energy Community Flexibility Solutions to Improve Users’ Wellbeing,” *Energies,* vol. 14, no. 12, 2021.

[14] CityLearn, “CityLearn,” 2023. [Online]. Available: https://www.citylearn.net/. [Accessed 20 08 2024].

[15] M. F. Faiz, M. Sajid, S. Ali, K. Javed and Y. Ayaz, “Energy modeling and predictive control of environmental quality for building energy management using machine learning,” *Energy for Sustainable Development,* vol. 74, pp. 381-395, 2023.

[16] J. Arroyo, C. Manna, F. Spiessens and L. Helsen, “Reinforced model predictive control (RL-MPC) for building energy management,” *Applied Energy,* vol. 309, 2022.

[17] S. Brandi, M. Fiorentini and A. Capozzoli, “Comparison of online and offline deep reinforcement learning with model predictive control for thermal energy management,” *Automation in Construction,* vol. 135, 2022.

[18] K. Ojand and H. Dagdougui, “Q-Learning-Based Model Predictive Control for Energy Management in Residential Aggregator,” *IEEE Transactions on Automation Science and Engineering,* vol. 19, no. 1, pp. 70 - 81, 2021.

[19] W. Cai, A. B. Kordabad and S. Gros, “Energy management in residential microgrid using model predictive control-based reinforcement learning and Shapley value,” *Engineering Applications of Artificial Intelligence,* vol. 119, 2023.

[20] W. Cai, S. Sawant, D. Reinhardt, S. Rastegarpour and S. Gros, “A Learning-Based Model Predictive Control Strategy for Home Energy Management Systems,” *IEEE Access,* vol. 11, 2023.

[21] W. Cai, S. Sawant, D. Reinhardt, S. Rastegarpour and S. Gros, “A Learning-Based Model Predictive Control Strategy for Home Energy Management Systems,” *IEEE Access,* vol. 11, 2023.

[22] P. Lissa, C. Deane, M. Schukat, F. Seri, M. Keane and E. Barrett, “Deep reinforcement learning for home energy management system control,” *Energy and AI,* vol. 3, 2021.

[23] Y. Zhang and L. Tong, "Regenerative braking-based hierarchical model predictive cabin thermal management for battery life extension of autonomous electric vehicles," *Journal of Energy Storage,* vol. 52, 2022.

[24] X. Zhou, F. Sun, C. Zhang and C. Sun, "Stochastically predictive co-optimization of the speed planning and powertrain controls for electric vehicles driving in random traffic environment safely and efficiently," *Journal of Power Sources,* vol. 528, 2022.

[25] H. Wei, L. Fan, Q. Ai, W. Zhao, T. Huang and Y. Zhang, "Optimal energy allocation strategy for electric vehicles based on the real-time model predictive control technology," *Sustainable Energy Technologies and Assessments,* vol. 50, 2022.

[26] H. He, M. Han, W. Liu, J. Cao, M. Shi and N. Zhou, "MPC-based longitudinal control strategy considering energy consumption for a dual-motor electric vehicle," *Energy,* vol. 253, 2022.

[27] Y. Tao, D. Schoeneberger, S. Rinderknecht, B. Morhard, D. Schweigert, M. Gerlach and U. Obernolte, "Model Predictive Energy Control Including Mechanical Fatigue Life of a Two-Motor Multi-Speed Electric Vehicle," *IFAC-PapersOnLine,* vol. 53, no. 2, pp. 14109-14116, 2020.

[28] A. Huang, Y. Mao, X. Chen, Y. Xu and S. Wu, "A multi-timescale energy scheduling model for microgrid embedded with differentiated electric vehicle charging management strategies☆," *Sustainable Cities and Society,* vol. 101, 2024.

[29] A. Cabrera-Tobar, A. M. Pavan, N. Blasuttigh, G. Petrone and G. Spagnuolo, "Real time Energy Management System of a photovoltaic based e-vehicle charging station using Explicit Model Predictive Control accounting for uncertainties," *Sustainable Energy, Grids and Networks,* vol. 31, 2022.

[30] C. Diaz-Londono, F. Ruiz, A. Mazza and G. Chicco, "Optimal Operation Strategy for Electric Vehicles Charging Stations with Renewable Energy Integration," *IFAC-PapersOnLine,* vol. 53, 2020.

[31] G. D'Amore, A. Cabrera-Tobar, G. Petrone, A. M. Pavan and G. Spagnuolo, "Integrating model predictive control and deep learning for the management of an EV charging station," *Mathematics and Computers in Simulation,* vol. 224, pp. 33-48, 2023.

[32] C. Wu, S. Jiang, S. Gao, Y. Liu and H. Han, "Event-triggered model predictive control for dynamic energy management of electric vehicles in microgrids," *Journal of Cleaner Production,* vol. 368, 2022.

[33] J. Liu, Z. Wang, Y. Hou, C. Qu, J. Hong and N. Lin, "Data-driven energy management and velocity prediction for four-wheel-independent-driving electric vehicles," *eTransportation,* vol. 9, 2021.

[34] E. M. Asensio, G. A. Magallán, L. Pérez and C. . H. De Angelo, "Short-term power demand prediction for energy management of an electric vehicle based on batteries and ultracapacitors," *Energy,* vol. 247, 2022.

[35] Y. Wu, Z. Huang, H. Hofmann, Y. Liu, J. Huang, X. Hu, J. Peng and Z. Song, "Hierarchical predictive control for electric vehicles with hybrid energy storage system under vehicle-following scenarios," *Energy,* vol. 251, 2022.

[36] C. Srithapon and D. Månsson, “Predictive control and coordination for energy community flexibility with electric vehicles, heat pumps and thermal energy storage,” *Applied Energy,* vol. 347, 2023.

[37] V. B. Truong and L. B. Le, “Building energy management and Electric Vehicle charging considering battery degradation and random vehicles’ arrivals and departures,” *Journal of Energy Storage,* vol. 64, 2023.

[38] A. Ouammi, “Peak load reduction with a solar PV-based smart microgrid and vehicle-to-building (V2B) concept,” *Sustainable Energy Technologies and Assessments,* vol. 44, 2021.

[39] T. Thenmozhi, M. V. Suganyadevi, R. Ramya and P. Muthukumar, “Hybrid energy management on electric vehicles for power grids with renewables system,” *Environmental Challenges,* vol. 9, 2022.

[40] F. Zhou, Y. Li, W. Wang and C. Pan, “Integrated energy management of a smart community with electric vehicle charging using scenario based stochastic model predictive control,” *Energy and Buildings,* vol. 260, 2022.

[41] L. Liu, D. Xu and C.-S. Lam, “Two-layer management of HVAC-based Multi-energy buildings under proactive demand response of Fast/Slow-charging EVs,” *Energy Conversion and Management,* vol. 289, 2023.

[42] J. Gasser, H. Cai, S. Karagiannopoulos, P. Heer and G. Hug, “Predictive energy management of residential buildings while self-reporting flexibility envelope,” *Applied Energy,* vol. 288, 2021.

[43] A. Colmenar-Santos, A.-M. Muñoz-Gómez, E. Rosales-Asensio, G. F. Aznar and N. Galan-Hernandez, “Adaptive model predictive control for electricity management in the household sector,” *International Journal of Electrical Power & Energy Systems,* vol. 137, 2022.

[44] Z. Ma and H. N. Koutsopoulos, “Near-on-demand mobility. The benefits of user flexibility for ride-pooling services,” *Transportation Research Part C: Emerging Technologies,* vol. 135, 2022.

[45] S. Jiang, C. Q. Tran and M. Keyvan-Ekbatani, “Regional route guidance with realistic compliance patterns: Application of deep reinforcement learning and MPC,” *Transportation Research Part C: Emerging Technologies,* vol. 158, 2024.

[46] Z. Lee and K. M. Zhang, “Generalized reinforcement learning for building control using Behavioral Cloning,” *Applied Energy,* vol. 204, 2021.

[47] L. Zhao, T. Yang, W. Li and A. Zomaya, “Deep reinforcement learning-based joint load scheduling for household multi-energy system,” *Applied Energy,* vol. 324, 2022.

[48] M. Akdağ, T. A. Pedersen, T. Fossen and T. A. Johansen, “A decision support system for autonomous ship trajectory planning,” *Ocean Engineering,* vol. 292, 2024.

[49] emergingindiagroup, “Understanding Long Short Term Memory (LSTM) Networks,” emergingindiagroup, 23 May 2024. [Online]. Available: https://emergingindiagroup.com/long-short-term-memory-lstm/. [Accessed 14 August 2024].

[50] E. Lutins, “Ensemble Methods in Machine Learning: What are They and Why Use Them?,” 2 August 2017. [Online]. Available: https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f. [Accessed 22 08 2024].

[51] Wikipedia, “Linear_regression,” Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Linear_regression. [Accessed 15 09 2024].

[52] scikit-learn, “Linear Models,” scikit-learn, [Online]. Available: https://scikit-learn.org/stable/modules/linear_model.html#linear-model. [Accessed 26 08 2024].

[53] geeksforgeeks, “Particle Swarm Optimization (PSO) – An Overview,” geeksforgeeks, 25 05 2023. [Online]. Available: https://www.geeksforgeeks.org/particle-swarm-optimization-pso-an-overview/. [Accessed 16 08 2024].

[54] MathWorks, “Particle Swarm Optimization Algorithm,” MathWorks, [Online]. Available: https://www.mathworks.com/help/gads/particle-swarm-optimization-algorithm.html. [Accessed 02 09 2024].

[55] Z. Nagy, J. R. Vázquez-Canteli, S. Dey and G. Henze, “The citylearn challenge 2021,” 17 11 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3486611.3492226.