

# ClimAite Control

*MSc Thesis in Building Technology*

Sebastian Fischer Stripp

2023



**MSc thesis in Building Technology**

# **ClimAlte Control**

Sebastian Fischer Stripp

June 2023

A thesis submitted to the Delft University of Technology in  
partial fulfillment of the requirements for the degree of Master  
of Science in (AUBS) Building Technology

Sebastian Fischer Stripp: *ClimAIte Control* (2023)

Supervisors: Dr. Michela Turrin  
Dr. Regina Bokel  
External Examiner: Dr. Olindo Caso

# Abstract

Current building operations can be improved through smart predictive operation based on weather and use patterns in order to save energy with minimal impact on the building fabric and daily use. The existing literature has investigated implementations, and potential savings through combining with variable tariffs, however, this thesis addresses the issue of how different buildings differ in their suitability for such smart control.

In this thesis, a digital twin of a school is created and adjusted to test differences in building fabric factors. These are combined with multiple Deep Reinforcement Learning (DRL) agents, which are trained to operate the schools more efficiently by controlling the heating set-point as well as natural ventilation of the buildings in order to save energy while maintaining comfort. The DRL agents vary in their ability to observe future weather as well as their internal network model architecture.

The results show a high energy saving compared to a simple baseline, despite the few building controls available to the agents. In addition, some algorithms out-compete even rule-based controllers, which were tested as a stricter baseline. The results also confirm a theory revealed through the literature review, that buildings with higher energy input, storage and control have a larger potential for energy savings. Additionally, the types of DRL models used also greatly influences the agents' ability to perform well, and generally more advanced models performed better. The findings can be used to assess a building's suitability for and potential benefits from such predictive smart control.



# Acknowledgements

First and foremost I would like to thank my two mentors Dr. Michela Turrin and Dr. Regina Bokel who supported me and challenged my ideas throughout this journey.

I also could not have done this without the building data provided by Architype and the weather data provided by Dr. Eleonora Brembilla. In addition, thanks goes to my friend and PhD student in AI for autonomous vehicle safety, Frederik Mathiesen, for his advice and technical support. I also thank Dr. Liangliang Nan for first igniting my spark for AI through his course on ML techniques and programming. @HOK and the VR Lab at BK, TU Delft also deserve recognition for allowing me special access to their local supercomputer, Beast, for running my experiments. Finally, thanks goes to the TU Delft Computer Science department and Dr. Wendelin Böhmer for letting me audit the course CS4400 Deep Reinforcement Learning in preparation for this thesis.





# Contents

<b>1. Introduction</b>	<b>1</b>
1.1. Background	1
1.1.1. Developments in Artificial Intelligence	1
1.1.2. Climate Change	1
1.1.3. Building Energy Prediction and Operation	2
1.2. Problem Statement	2
1.3. Research Objective	3
1.4. Research Question	3
1.5. Scope	4
1.6. Methodology	6
1.7. Relevance of Research	7
1.7.1. Scientific Relevance	7
1.7.2. Societal Relevance	7
<b>2. Literature Review</b>	<b>9</b>
2.1. Indoor Comfort	9
2.2. Building as Energy Balance System	11
2.2.1. Energy Storage	11
2.2.2. Dynamic Energy Supply	13
2.2.3. Conclusion	13
2.3. Building Energy Modelling	14
2.3.1. Digital Twins	15
2.3.2. Calibration	15
2.3.3. Conclusion	17
2.4. Machine Learning	18
2.4.1. Reinforcement Learning	18
2.4.2. Neural Networks	20
2.4.3. Deep Q-Learning	22
2.4.4. Recurrent Neural Networks	22
2.4.5. Curse of Dimensionality	23
2.4.6. Surrogate Models	24
2.4.7. Challenges in Deployed Machine Learning Applications	24
2.5. Building Control	26
2.5.1. Building Operation	26
2.5.2. Advanced Building Control	26
2.5.3. Deep Reinforcement Learning Building Control	31
2.5.4. Conclusion	32
<b>3. Case Study: Welsh School</b>	<b>33</b>
3.1. Digital Twin	36
3.2. Methodology	36
3.2.1. Software Tools Descriptions	36

Contents

3.2.2.	Digital Twin Creation Workflow . . . . .	36
3.3.	Digital Twin Documentation . . . . .	38
3.3.1.	Building Volume . . . . .	38
3.3.2.	Thermal Mass . . . . .	41
3.3.3.	Construction Build-ups . . . . .	42
3.3.4.	Schedule and Equipment Usage . . . . .	44
3.3.5.	Hot Water and HVAC . . . . .	47
3.3.6.	Natural Ventilation . . . . .	49
3.3.7.	Custom Parameters . . . . .	50
<b>4.</b>	<b>Calibration and Validation</b> . . . . .	<b>53</b>
4.1.	On-site data . . . . .	53
4.1.1.	Temperature Data . . . . .	54
4.1.2.	Calibration Equations . . . . .	54
4.1.3.	Digital Twin Validation . . . . .	55
4.1.4.	Conclusion . . . . .	57
<b>5.</b>	<b>Reinforcement Learning Algorithms and Experiments</b> . . . . .	<b>59</b>
5.1.	Programming Setup . . . . .	59
5.2.	Reinforcement Learning . . . . .	60
5.2.1.	Agent and Environment Interaction . . . . .	60
5.2.2.	RL Agents . . . . .	61
5.2.3.	Hyperparameters . . . . .	64
5.2.4.	Training Processes and Developments . . . . .	67
5.3.	Building Digital Twin Variations . . . . .	67
5.4.	PyTorch Model Code . . . . .	69
<b>6.</b>	<b>Climate AI Results</b> . . . . .	<b>71</b>
6.1.	Experiments Completed . . . . .	71
6.2.	Consistency with Literature . . . . .	73
6.2.1.	A Easy Winner - Low Energy and Higher Comfort . . . . .	73
6.3.	Further Analysis . . . . .	75
6.3.1.	Energy Savings vs Comfort . . . . .	75
6.3.2.	Comparing Based on Metrics . . . . .	77
6.3.3.	RL24hAllRNN vs RL04hAllRNN . . . . .	81
6.3.4.	Better Than Lowering the Thermostat . . . . .	83
6.3.5.	Low-insulation Buildings . . . . .	85
6.3.6.	Comparing Long Term Trends . . . . .	86
6.4.	Alternative Baselines . . . . .	86
6.5.	Reinforcement Learning Metrics . . . . .	90
<b>7.</b>	<b>Discussion</b> . . . . .	<b>93</b>
7.1.	Results . . . . .	93
7.1.1.	Achieving a 1% CO <sup>2</sup> Reduction . . . . .	93
7.1.2.	Limitations . . . . .	93
7.2.	Real-world Impact . . . . .	93
7.2.1.	Stakeholder Decisions . . . . .	94
7.2.2.	Standardisation of Equipment . . . . .	95
7.3.	Future Improvements and Excluded Complexities . . . . .	95
7.3.1.	Advanced Building Control . . . . .	95

7.3.2. Building Physics Variations . . . . .	96
7.3.3. Reinforcement Learning Improvements . . . . .	96
7.4. Benchmarking . . . . .	97
<b>8. Conclusion</b>	<b>99</b>
8.1. Key Findings . . . . .	99
8.2. Changes to Design and Retrofitting . . . . .	99
8.3. Limitations . . . . .	100
8.4. Future Research . . . . .	100
<b>9. Reflection</b>	<b>101</b>
9.1. Research and Design . . . . .	101
9.2. Ethics . . . . .	101
9.3. Societal Impact . . . . .	102
<b>A. Glossary</b>	<b>103</b>
A.1. Building Physics . . . . .	103
A.2. Honeybee and EnergyPlus . . . . .	103
A.3. Machine Learning . . . . .	105
<b>B. Alternative Calibration Methods</b>	<b>107</b>
B.1. Calibration Methods . . . . .	107
B.2. Genetic Algorithm for Calibration . . . . .	109
<b>C. Additional Results from Analyses</b>	<b>111</b>
C.1. Metrics Performance Scores . . . . .	111
C.2. Degree-hours vs Energy use for all building types . . . . .	111



# List of Figures

1.1. Chart of the problem statement. . . . .	3
1.2. Research Landscape, within which this thesis is placed. . . . .	4
1.3. Chart of the methodology . . . . .	8
2.1. Diagram of common Indoor Environment Quality metrics. . . . .	9
2.2. Diagram of the relationship between predicted mean vote (PMV) and the percentage of dissatisfied. This shows that even at the average comfort point, not everyone can be satisfied as individual preferences differ. From <a href="#">Satake et al. (2016)</a> . . . . .	10
2.3. Simplified diagram of building energy flows. A building has energy gains and losses, as well as different types of energy storage. (The sizes of flows are only indicative and not representative of their true proportions) . . . . .	11
2.4. Regardless of whether energy performance is predicted using common tools as in <a href="#">2.4a</a> or using more detailed tools such as PHPP in <a href="#">2.4b</a> , the measured results will always deviate due to differences in use. . . . .	14
2.5. An example from <a href="#">Muehleisen and Bergerson (2016)</a> of how the calibration takes in a prior (dashed red lines) assumption of the factors being calibrated, and then calculates the posterior density function (blue bars) of the factors, in order to match simulated data to measured data. . . . .	16
2.6. Simplified overview of the taxonomy of Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL), data science. . . . .	18
2.7. An overview of the steps for the agent interacting with its environment. After each action, the environment outputs a new state and a reward to the agent. Then the agent decides the next action and the process continues. At each iteration the agent learns from its previous action and reward. Adapted from the book by <a href="#">Sutton and Barto (2018)</a> . . . . .	19
2.8. Diagram of the optimality of actions chosen by an agent given different levels of exploration in the model. At 0 exploration, greedy green, the agent will not progress over time as it does not explore alternative actions. From the book by <a href="#">Sutton and Barto (2018)</a> . . . . .	20
2.9. Graph comparing the GELU, ReLU and ELU functions. From ( <a href="#">Kon, 2022</a> ) . . . . .	21
2.10. The diagram shows a tabular example of a Reinforcement Learning (RL) agent's value function on the left and a neural network example on the right. S denotes the state and A denotes the action. In the tabular example, each state and action is mapped to a predicted value shown as a number. The agent will pick the highest expected value shown encircled in blue. However, it may be forced to a random choice encircled in orange. In DRL the table is replaced with a Neural Network (NN), which can simplify complex problems as the possible state space grows very large. There rather than each state and action having particular matching values, the state serves as the input to the Neural Network (NN), outputting expected values for each possible action. . . . .	22
2.11. Illustration of the repeating structure of the RNN. Diagram by <a href="#">Olah (2015)</a> . . . . .	23

2.12. Detail of RNN and the NN with a tanh activation function. Diagram by Olah (2015) . . . . .	23
2.13. Detail of LSTM and its multiple NNs with tanh and Sigmoid activation functions. Diagram by Olah (2015) . . . . .	24
2.14. These four diagrams show an example of a dimensionality problem. In (a) there are two dimensions, which allows us to cluster the samples according to which dimension they are closest to. In (b) there are 8 dimensions. However, this leads to overfitting of the data and no valuable abstraction can be made. In (c) the lower-level features (dimensions) are mapped according to their exposure to the higher-level features. The result is plotted in (d). The step from (a) to (c) allows us to maintain more detailed information about each individual sample, while still being able to abstract out information about them as a group. Graphics from (Yiu, 2021). Note: Features, dimensions, and categories can be used to refer to the same thing depending on the context. . . . .	25
2.15. MPC uses variations of what can be likened to a tree search to find an optimum action, whereas DRL learns the optimum action based on experience derived from data . . . . .	27
2.16. Non-exhaustive chart of the taxonomy of optimal control. From the paper by Arroyo, Manna, Spiessens and Helsen (2022) . . . . .	28
2.17. Plot of the level of control and energy storage potential of the buildings investigated in the literature. The circles' diameter indicates the level of savings achieved for each building when using smart or AI-type control algorithms for the simulated operation of the buildings. This diagram can be displayed with 3 or 4 axes but has been simplified for a 2D view to differentiate only between energy and control, however, there is an overlap between energy and control in certain aspects of the buildings. . . . .	30
2.18. Simplified diagram of the Deep Reinforcement Learning agent's building control mechanisms as well as the environmental constraints to satisfy and the objective to achieve. . . . .	31
2.19. A diagram of a standard DRL agent-environment loop adjusted to a building operation case. . . . .	31
3.1. Case study school exterior view, by Lowfield Timber Frames . . . . .	33
3.2. Ground floor, first floor and section drawings of the school, by Architype . . . . .	34
3.3. Case study school aerial view, by Powys County Council . . . . .	35
3.4. Case study school site plan, by Architype . . . . .	35
3.5. Diagram of software used and flow of information . . . . .	37
3.6. Grasshopper - Honeybee 3D energy model of case study . . . . .	38
3.7. Grasshopper - Zoning. Zone 1 ground floor and zone 2 first floor . . . . .	39
3.8. Grasshopper - Energy model surface definitions . . . . .	40
3.9. Grasshopper - Internal mass, ground floor . . . . .	41
3.10. Grasshopper - Floor construction . . . . .	42
3.11. Grasshopper - Wall and roof construction . . . . .	42
3.12. Grasshopper - Interior elements constructions . . . . .	43
3.13. Grasshopper - Window and louvre build-ups . . . . .	43
3.14. Grasshopper - Use schedule setup . . . . .	44
3.15. Grasshopper - Internal and building loads . . . . .	46
3.16. Grasshopper - Hot water system, mechanical and natural ventilation . . . . .	48
3.17. Grasshopper - Close-up of windows and vertical louvres in green. Shading overhangs not visible in view. . . . .	49

3.18. Grasshopper - <i>HB Set Identifier</i> , for forcing consistent naming in IDF file . . . . .	50
3.19. Grasshopper - Creation of energy management system (EMS) Actuator objects for EnergyPlus in EnergyPlus Runtime Language (ERL) . . . . .	51
3.20. Grasshopper - Actuators in EnergyPlus Runtime Language (ERL) format . . . . .	51
3.21. Grasshopper - EMS and output files added to simulation parameters output . . . . .	52
4.1. Cumulative energy usage measured from the building. . . . .	53
4.2. Temperature data for calibration is from classrooms G36, G13, F04, and F16. . . . .	54
4.3. Temperature data measured from site and simulated by energy model. Blue: Ground Floor measured data. Purple: First floor measured data. Red and Yellow: Simulated data for ground and first floor zones. . . . .	55
4.4. Averaged temperature data measured from site and simulated by energy model. Green: Measured data. Red: Simulated data. . . . .	56
5.1. Diagram of agent and environment state, action and reward cycle. . . . .	59
5.2. Diagram of multiple time-steps in the simulation and agent interaction. At each hourly time-step the agent interacts with the EnergyPlus application programming interface (API). . . . .	60
5.3. Diagram of agent and EnergyPlus interaction in detail. . . . .	61
5.4. Diagram of the deep recurrent Q neural network of the agent . . . . .	62
5.5. Types of RL agent, how far in the future they see specific weather information. In addition, an agent without foresight and an agent without any Recurrent Neural Network. . . . .	62
5.6. Graph of RL agent's penalty function for internal temperatures . . . . .	66
5.7. Illustration of the 9 types of buildings created as variations from the original digital twin model. . . . .	68
6.1. Diagrams of relative and absolute savings compared to the EPBaseline operation. The diagrams are laid out similarly to the building variants illustration in figure 5.7, with increasing insulation on the x-axis and increasing thermal mass on the y-axis. . . . .	73
6.2. Comparing the RL04hFlatInput and the RL24hAllRNN performances to the EPBaseline for MJ energy use and internal temperature over the first 17 days. The first day is Monday. . . . .	74
6.3. Bar chart of final reward/penalty scores for each agent. Note: the scores are negative, as the agents are given negative rewards rather than positive rewards. The algorithms aim to maximize reward, meaning the least negative is still the highest performing. . . . .	76
6.4. Bar charts of total score over 52 weeks using each metric. $f = 4$ and $k = 1.00005$ used. . . . .	78
6.5. Line chart of the scoring over 52 weeks used in Figure 6.4b with <i>linear ratio multiplication</i> . . . . .	79
6.6. Bar charts of each building variant using the <i>linear ratio division</i> metric over the first 8 weeks of the year to avoid $d$ being significantly less than 1. . . . .	80
6.7. Baseline building energy and temperature comparison of RL24hAllRNN and RL04hAllRNN . . . . .	81
6.8. Interior and exterior hourly temperature plot for the first 6 months. . . . .	82
6.9. Interior and exterior hourly temperature plot during <b>all hours</b> for EPBaseline, EPBaseline190, and RL04hAllRNN. First 6 months of the year. . . . .	83

*List of Figures*

6.10. Degree-hours plotted against energy use MJ for the baseline building for weeks 48 to 52. . . . .	84
6.11. Degree-hours plotted against energy use MJ for the low-insulation, baseline thermal mass building for weeks 48 to 52. . . . .	85
6.12. Degree-hours plotted against energy use MJ for the baseline building for weeks 1 to 25 . . . . .	86
6.13. Degree-hours plotted against energy use MJ for the rule-based controllers with trend line against the RL04hAllRNN agent (green star) for weeks 1 to 6 . . . . .	88
6.14. Degree-hours plotted against energy use MJ for the rule-based controllers with trend line against the RL04hAllRNN agent (green star) for the months of January, February, March, April, and May. . . . .	89
6.15. Agents' reward with 2-year mean for the final 11 years of the total 35 years of training simulation . . . . .	90
6.16. Agents' mean square error loss for the whole training period of 35 simulated years. . . . .	91
B.3. Genetic Algorithm concept overview . . . . .	109
B.4. Results from multiple runs of the genetic algorithm . . . . .	110
C.1. Bar chart of the accumulated score of the algorithms using the linear ratio division metric for all 9 building variants. . . . .	111
C.2. Bar chart of the accumulated score of the algorithms using the linear ratio multiplication metric for all 9 building variants. . . . .	112
C.3. Bar chart of the accumulated score of the algorithms using the power ratio metric for all 9 building variants. . . . .	113
C.4. Degree-hours plotted against energy use MJ for all 9 buildings for weeks 48 to 52. . . . .	114



# List of Tables

2.1. Thermal Properties of Common Materials . . . . .	12
4.1. Table showing the The American Society of Heating, Refrigerating and Air-Conditioning Engineers ( <i>ASHRAE</i> ) criteria and results of the simulation calibration for varying periods of week by week up to the full 9-month period. . .	56
6.1. Table of specific agents trained highlighted in blue. The vertical axis has the 9 building variants and the horizontal axis has the different control algorithms starting with the EnergyPlus baseline. . . . .	71
6.2. Table of the agents trained on the baseline building used to control all the building types. Degree-hours too cold and warm also shown. Blue highlighted shows the agents trained specifically for that building type similar to table 6.1 . . . . .	72
B.1. Parameters used for digital twin calibration in the genetic algorithm setup. . .	110



# Acronyms

LETI	London Energy Transformation Initiative	1
EPC	Environmental Performance Certificate	2
TFA	treated floor area	44
ASHRAE	The American Society of Heating, Refrigerating and Air-Conditioning Engineers	xvii
PMV	predicted mean vote	9
HVAC	heating, ventilation, air-conditioning, and cooling	6
MVHR	mechanical ventilation with heat recovery	26
PID	proportional–integral–derivative	26
NMBE	Normalized Mean Bias Error	17
CV(RMSE)	Coefficient of Variation of the Root Mean Square Error	17
ML	Machine Learning	xiii
AI	Artificial Intelligence	xiii
RL	Reinforcement Learning	xiii
DRL	Deep Reinforcement Learning	v
DQN	Deep Q-Learning	22
DRQN	Deep Recurrent Q-Learning	60
DL	Deep Learning	xiii
NN	Neural Network	xiii
CNN	Convolutional Neural Network	20
RNN	Recurrent Neural Network	22
GRU	Gated Recurrent Unit	22
LSTM	Long Short-Term Memory	22
ReLU	Rectified Linear Units	21
GELU	Gaussian Error Linear Units	21
API	application programming interface	xv
GUI	graphical user interface	104
EMS	energy management system	xv
ERL	EnergyPlus Runtime Language	xv
MPC	model predictive control	26
NLP	natural language processing	1
B4B	Brains 4 Buildings	2
CLT	cross-laminated-timber	12



# 1. Introduction

## 1.1. Background

### 1.1.1. Developments in Artificial Intelligence

Before the recent boom in natural language processing (NLP) models such as [ChatGPT](#) and image generator models such as [Midjourney](#) and Stable Diffusion by [Stability AI](#), the focus was on using AI to beat games like Chess and Go. DeepMind created multiple iterations of RL agents to play the traditional board game Go. The knowledge developed within the company was then applied to the problem of protein folding prediction as they successfully developed [AlphaFold](#). The board games of Chess and Go as well as protein folding represent full information games or scenarios, in that there is no explicitly hidden information. This is different from games like poker, where the opponents' cards are hidden and the board game Stratego, where all the opponent's pieces are hidden. Late in 2022, DeepMind announced they have made significant progress in Stratego, ranking in the top three of human players on the online Stratego platform Gravon ([Perolat et al., 2022](#)). This came at the same time as another research group tackled Diplomacy, which is a game using natural language rather than predetermined moves to negotiate with other players in real-time about the game strategies and outcomes ([The Economist, 2022](#)).

These advances in Reinforcement Learning (RL) research and applications are promising for the industries they are applied to, however, there are also many challenges in applying ML in the real world, which is often much noisier than a simulated environment.

### 1.1.2. Climate Change

The building and construction sector is responsible for 40% of energy consumption and 36% of greenhouse gasses in Europe, according to a report by the [European Commission \(2020\)](#). Countries with an old building stock and poorly insulated houses such as the UK and partially the Netherlands ([Yanatma, 2022](#)) may have an even more polluting building sector. According to the London Energy Transformation Initiative ([LETI](#)) [2020](#) the UK's building sector is responsible for 49% of greenhouse gas emissions, with 80% being solely from the operation of existing buildings. This means any impact on the building sector will have a large impact on global resource use. Looking towards the 2050 targets the poor performance of existing buildings becomes even more apparent. According to McKinsey & Co. 80% of the buildings in 2050 already exist today ([Blanco et al., 2021](#)), highlighting that renovation is paramount and any small impact on existing buildings today will pay dividends long into the future. A solution that can be applied to all existing buildings, reducing their energy consumption by just 3% will reduce the UK greenhouse gas emissions by 1.18%.

## 1. Introduction

Demand response, peak shifting and grid peak demands are also important factors (LETI, 2020), as this will help reduce the reliance on fossil fuels, reduce the strain on the electricity grid, as well as being able to take better advantage of fluctuating renewable energy supply.

### 1.1.3. Building Energy Prediction and Operation

With an increased focus on sustainability as well as accountability, the building industry is going through a quiet reckoning. Current energy models and predictions used in building design may be just that - predictions. Bloomberg reports that a few buildings in the UK with an Environmental Performance Certificate (EPC) of B, are actually measured as performing worse than buildings rated with a C (Hesketh, 2022). This may be caused by economic incentives to make a building appear to perform better in the design stages.

According to Brains 4 Buildings (B4B) 10-30% of energy going into buildings is wasted due to poor control or faulty installations (B4B, 2023) (Chitkara et al., 1/july/2022). This is without even counting the additional energy that can be saved through predictive control. Broadly, these are referred to as *smart* systems, *smart* buildings or *smart* operations. This can include fault monitoring, error detection, or even predictive capabilities. Going forward smart operation will also be referred to as AI operation, as some systems use tools and algorithms from the field of AI.

## 1.2. Problem Statement

**As we move towards a new paradigm of AI-operated buildings there is a lack of understanding of how RL control algorithms compare with building fabric factors, and how they will influence what is considered to be a low-energy building**

As outlined in the Background, there is a move towards more sophisticated control of buildings, through the use of Artificial Intelligence. This means a shift in how buildings operate, but also a change in what is perceived as an energy-efficient building. This comes at a pressing time when building operators are trying to save both energy and money to mitigate climate change and reduce costs.

Hundreds of studies have been carried out investigating and testing control algorithms for individual buildings (Yu et al., 2015), and some also look at two buildings within one study (Cígler et al., 2013), allowing for a more direct comparison of algorithms or buildings. However, the literature is lacking studies focusing directly on the influence of variations within a control algorithm in **comparison** to building fabric factors such as u-value, airtightness, and thermal storage. This also influences how we approach retrofitting and new design, as the new paradigm will unlock a better understanding of how to combine certain control algorithms with certain types of building design for the best operational savings while maintaining indoor comfort.

To align with this future trend of AI building operation, both designers and decision-makers need to know to what extent various RL control algorithms and building fabric factors impact the operation of the building.

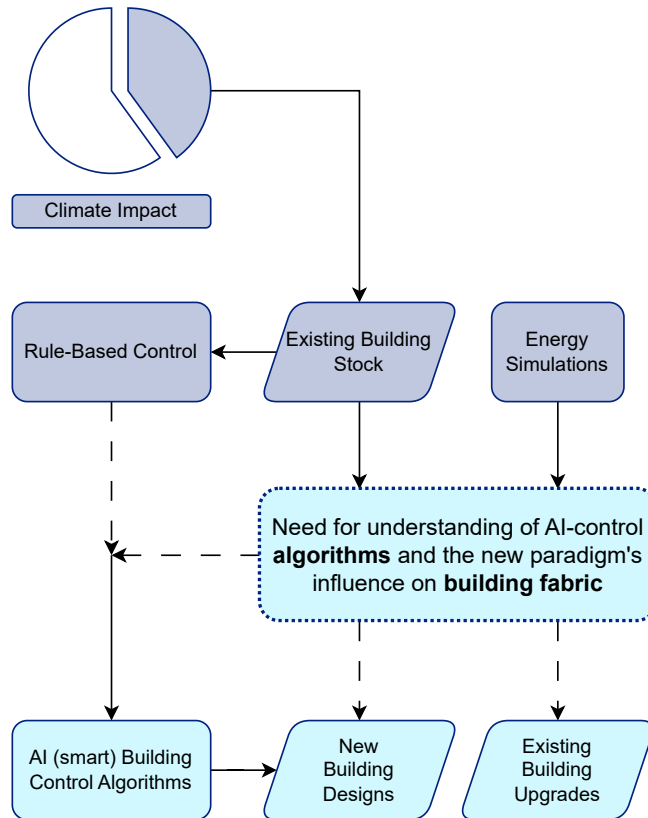


Figure 1.1.: Chart of the problem statement.

### 1.3. Research Objective

The main objective of this research is a better understanding of how Reinforcement Learning *control algorithms* compare with *building fabric factors* with regard to impacting the energy performance of a building.

This is done through the use of building energy modelling and testing various RL control algorithms and building scenarios to establish a picture of the relative impacts of each variable.

Finally, to explore to what extent designers and decision-makers need to change how they operate, design, and renovate buildings to be fit for a future of AI building operation.

### 1.4. Research Question

**To what extent do Reinforcement Learning *control algorithms* compared with *building fabric factors* influence a building's energy-saving potential, and how does that impact what is considered a low-energy building for operation, design, and retrofitting?**

## 1.5. Scope

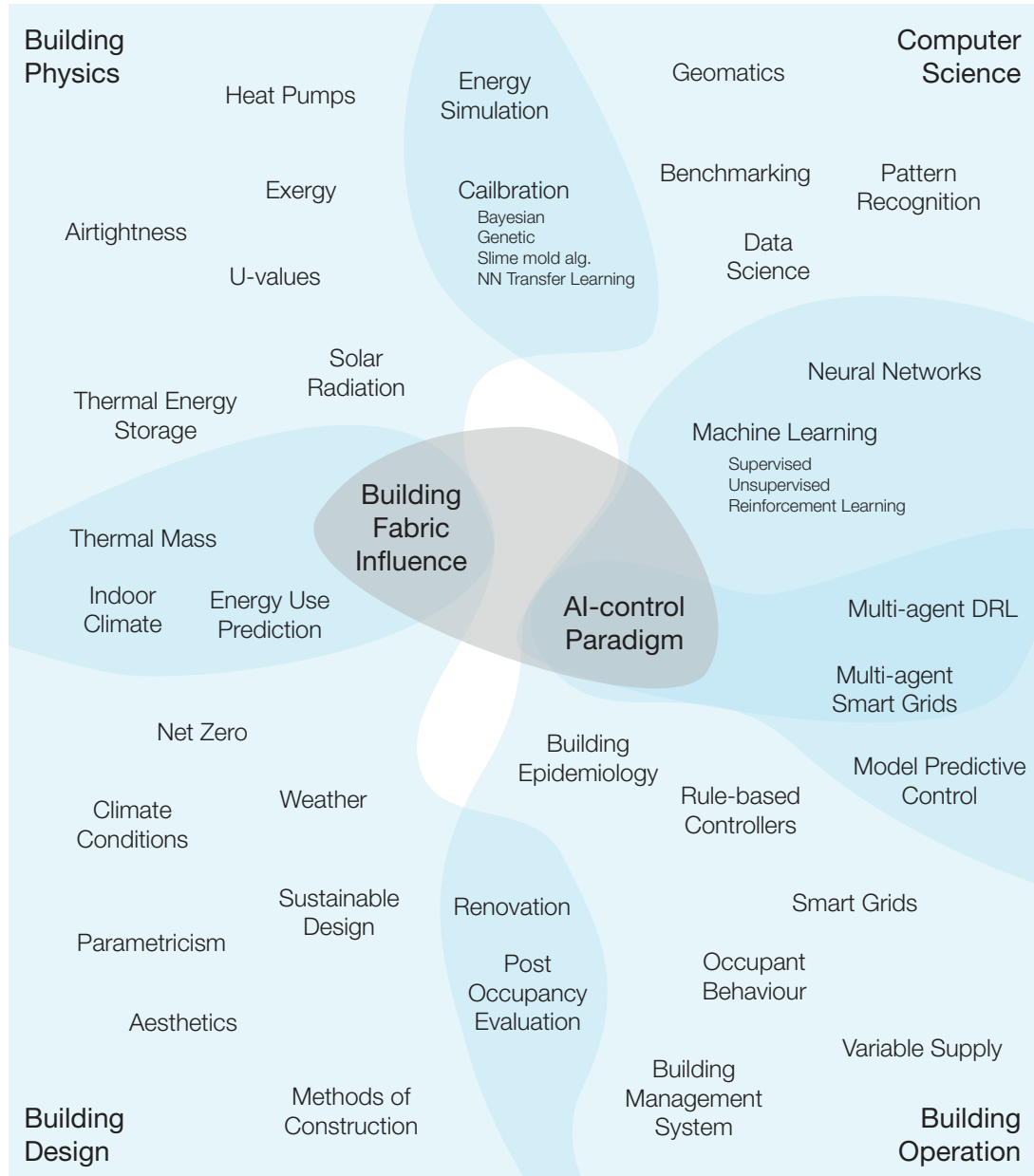


Figure 1.2.: Research Landscape, within which this thesis is placed.

This thesis sits at the intersection of multiple fields and touches upon many sub-disciplines within those fields as illustrated in the Research Landscape, Figure 1.2. The thesis focuses on only a small portion of the larger picture, looking directly at comparing *RL control algorithms* and *building fabric factors*. This is done through a case study of an existing building, which



serves as a means to the end of understanding, rather than being the main focus in itself. The case study is used for testing and simulations.

**While the thesis relates to the following areas in some ways, they fall outside the primary scope of this project. Some are still present in the project, but are not investigated to their full extent:**

- Impacts of single zone vs multi-zone calibration of the energy model.
- Sensitivity of calibration accuracy on performance gains in subsequent simulations.
- Weather forecasting uncertainties and time horizons and their impact on model performance.
- Response to adverse weather and a changing climate leading to both abrupt and gradual changes to the weather.
- Adjusting hyperparameters of the DRL agent improve learning speed, NN size, and generalisability.
- Impacts of variable energy supply pricing and variable renewable energy supply.
- Determining detailed steps for implementation in existing building management systems.
- In-depth analysis of economic aspects and feasibility of various design and upgrade decisions arising from this project.

## 1.6. Methodology

The research is conducted through experiments to achieve the main objective. This is primarily done through building energy simulations with various [RL](#) control algorithms and numerous buildings with one or more building fabric factors adjusted from the baseline. The method is diagrammed in figure 1.3, showing the research framework, literature review, building modelling, AI-control modelling, testing and data analysis. These are elaborated below.

### Building Modelling

The first step is to create a building energy model for the chosen case study using Honeybee in Grasshopper as an interface for EnergyPlus. This is used as the base upon which further research and experiments are conducted. This model is simulated at multiple time intervals and includes both mechanical heating, ventilation, air-conditioning, and cooling ([HVAC](#)) controls as well as natural climate control systems such as opening windows.

This building energy model is then calibrated against the real-world case study to bring the model from a mere simulation to a real-world building digital twin. This is done using data measured on-site over the course of a year and a half. However, that data is partially impacted by local Covid-19 shutdowns in the area.

Weather data from the site is used to calibrate the model to match the on-site monitoring for that specific time period. Separate weather data is then used for training and testing the smart control algorithms in subsequent simulations.

### AI Building Operation

Once the case study is simulated and calibrated, multiple [RL](#) control algorithms for the building operation will be implemented. This requires connecting through the [API](#) of EnergyPlus to manipulate the simulation during runtime.

### AI-Control Testing

In the testing phase, each [RL](#) control algorithm is then tested against the baseline operation of the building to measure any performance improvement of this advanced control of the building. In addition, multiple similar buildings will be created by changing one or more of the building fabric factors, and these will then be tested with each of the [RL](#) control algorithms too.

### Evaluation

The results are then analysed. First, in order to see the improvement in energy-saving when using smart controllers. Secondly, to see if the impactful building fabric factors are different for standard control vs smart control buildings. Thirdly, to compare the relative impacts of [RL](#) control strategies vs building fabric differences.

## Discussion

Finally, the results from the evaluation are condensed into key lessons learned about the relative impacts of RL control strategies vs building fabric differences.

## 1.7. Relevance of Research

### 1.7.1. Scientific Relevance

This research aims to address the current gap in understanding of what makes some buildings better suited for different smart control operations and what role their building fabric plays. This will assist future researchers who are comparing existing literature, which often only assesses one control algorithm or building at a time. It will help researchers who are looking to implement smart control on real buildings to assess what kind of performance difference to expect.

Lastly, the RL control methods developed through this research can also be applied to similar simulations of other buildings to compare algorithms and performance differences.

### 1.7.2. Societal Relevance

This research is part of a larger field looking at better control paradigms for existing and future buildings. This broad body of knowledge has three major implications for society:

**Energy Savings** As we operate our buildings in a better way, we can reduce energy use.

**Economic Savings** Reducing energy use will save costs of operation, which can then be used elsewhere such as building maintenance.

**Climate Change Mitigation** As buildings and their operation account for a large portion of European energy use, a reduction in energy use will help reduce the impact of climate change.

This particular thesis is just part of the larger picture, but can in itself be used by building owners, designers and decision-makers to influence which buildings to first switch to a smart control system, which buildings to potentially upgrade for a future of smart control and finally what to focus on in upgrading.

1. Introduction

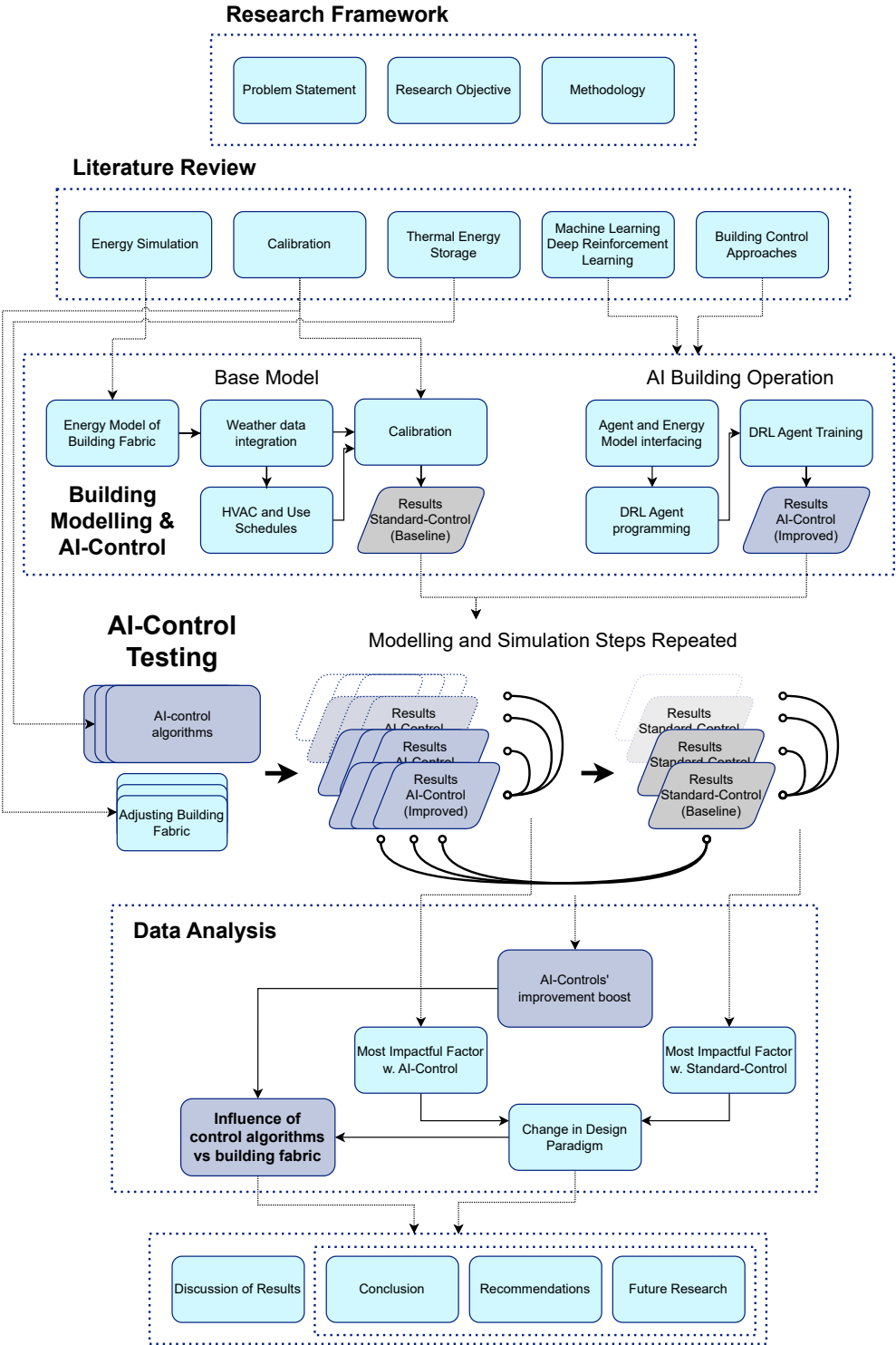


Figure 1.3.: Chart of the methodology

## 2. Literature Review

This chapter 2 gives an overview of the topics of indoor comfort, building energy simulation, calibration, reinforcement and machine learning techniques, building operation, and data analysis.

### 2.1. Indoor Comfort

Indoor comfort is implicitly one of the primary reasons for architecture and engineering - to tame and manage space; to make it comfortable across multiple measures such as temperature, air quality, light quality etc.

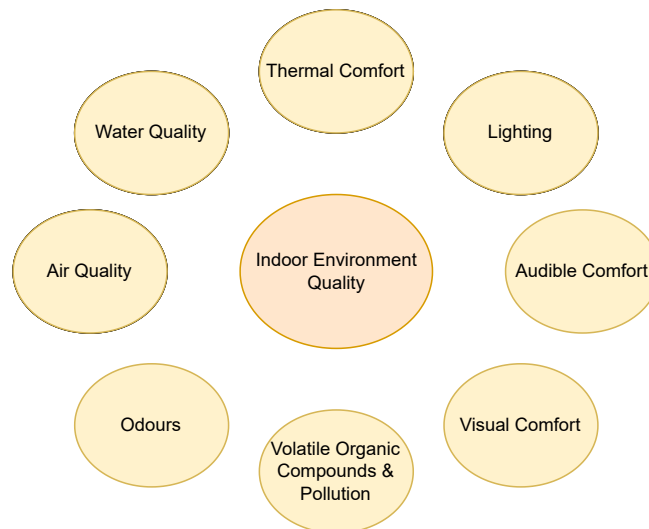


Figure 2.1.: Diagram of common Indoor Environment Quality metrics.

For the operation of existing buildings, one of the dominant factors is indoor temperature and this will also be the primary measure of indoor comfort for this thesis. There exist multiple methods solely for assessing indoor temperature comfort one of which is predicted mean vote (*PMV*), however, newer models such as Dear en Brager are more adaptive in relation to outdoor temperatures (*van der Linden et al., 2013*). One of the key things to note is that most models predict that a certain percentage of occupants will always be dissatisfied, which for *PMV* is at least 5%, see also Figure 2.2. In addition, these measures are further complicated by additional factors such as clothing level, activity level, and temperature asymmetries such as drafts and radiant differences.

## 2. Literature Review

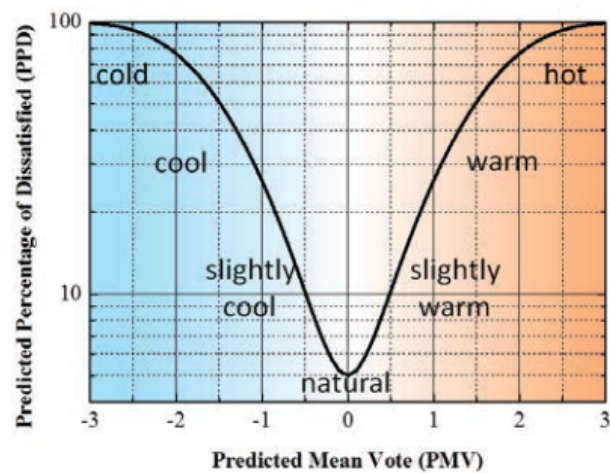


Figure 2.2.: Diagram of the relationship between predicted mean vote (PMV) and the percentage of dissatisfied. This shows that even at the average comfort point, not everyone can be satisfied as individual preferences differ. From [Satake et al. \(2016\)](#)

Many models of indoor temperature comfort also do not differentiate between types of buildings, nor the age of the occupants and this can also have an impact on the level of comfort ([Lamberti, 2020](#)). To simplify the problem of indoor comfort an allowable temperature range can be used as a measure of indoor comfort. The ISO-7730 standard recommends a temperature of 20-24° C in winter and 23-26° C in summer based on the PMV approach ([Lamberti et al., 2020](#)).

## 2.2. Building as Energy Balance System

A building can be thought of as a vessel for storing energy. It gains energy from the sun, the power grid and occupants, but loses energy through its envelope and its operational functions such as ventilation and water use. In addition, it can store energy both short term and long term in the forms of thermal energy in the building materials and hot water storage, chemical energy in batteries, and even the air volume inside the building stores some amount of heat. These flows are outlined in Figure 2.3. The supplies and demands fluctuate throughout the day, week, and year. These fluctuations can be anticipated and taken advantage of to save energy and money as well as to decrease pressure on the shared electrical grids.

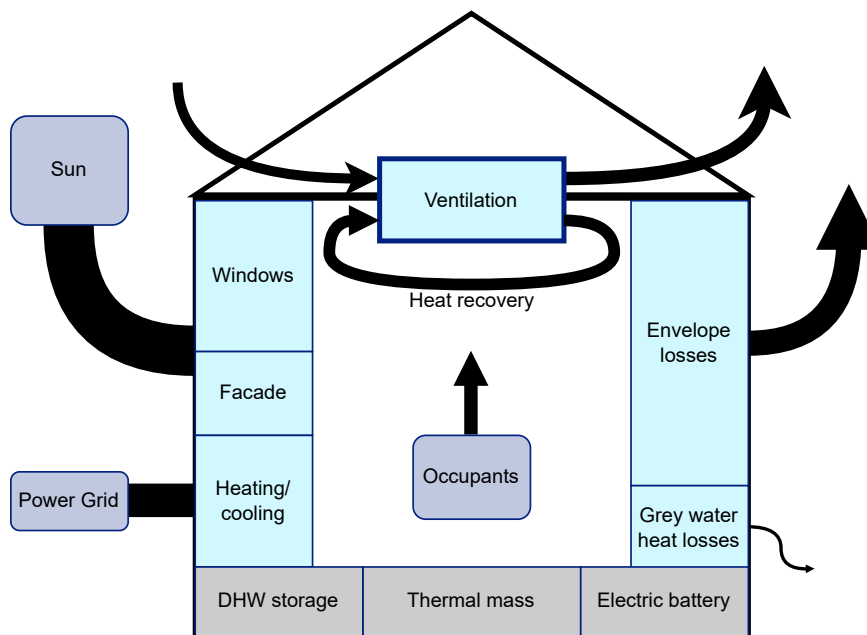


Figure 2.3.: Simplified diagram of building energy flows. A building has energy gains and losses, as well as different types of energy storage. (The sizes of flows are only indicative and not representative of their true proportions)

### 2.2.1. Energy Storage

Energy can be stored in multiple ways including chemical, kinetic, thermal, and potential. In buildings, thermal storage is literally built-in into the building, though it is becoming more widespread to have chemical storage in the form of batteries or power packs. For this reason, buildings of the same shape and size can perform differently depending on what material they are made of, because of the thermal heat storage of the materials. This is most commonly experienced when walking into a historic stone building without any modern heating or ventilation systems. The building temperature will hover around the mean ambient temperature with only smaller fluctuations compared to a modern lightweight building.

## 2. Literature Review

Material	Density [kg/m <sup>3</sup> ]	Thermal Conductivity [W/(mK)]	Thermal Heat Capacity [kJ/(kgK)]	Thermal Heat Capacity [kJ/(m <sup>3</sup> K)]
Concrete	2400	2.250	0.88	2112
Timber	750	0.038	1.20	900
Cellulose Ins.	60	0.035	2.11	127
Water	1000	0.598	4.20	4200
Ambient Air	1.204	0.040	0.70	0.8428

Table 2.1.: Thermal Properties of Common Materials

Table 2.1 shows the thermal properties of a few common materials. The exact figures do vary with composition as concrete and timber come in many different types. The column on thermal conductivity describes how quickly heat transfers through a material. The two columns on thermal heat capacity describe how many kilo Joules of energy the material can store per kg in the fourth column and per cubic meter in the last column. In design it is more common to know the volume rather than the weight of a specific wall or floor, hence the final column is most useful for comparison.

A simple example using Table 2.1 would be to compare a concrete building, a cross-laminated-timber (CLT) building, and a common timber framed building. For ease, we will assume the walls have the same thickness of 10 cm. From the last column, it becomes clear that the concrete will have more than twice the thermal heat capacity of a solid CLT building. Compare that to a timber framed building with cellulose insulation infill and the difference becomes almost 10X. Even if the timber frame is slightly wider due to stiffness constraints, the difference in thermal heat storage capacity is still very significant, simply due to the material choice.

Energy storage is critical in order to be able to shift demand and make use of fluctuating supply. The review paper by Thieblemont et al. (2017) highlights the importance of thermal storage in order to shift demand. To support this, another paper by Cígler et al. (2013) showed that a modern lightweight building with high insulation has little energy-saving potential because it lacks thermal storage. The upside is that it responds quickly to heating inputs, but this can also be a problem for overheating in summer. Insulation is also important in high thermal mass buildings. Take for instance the study by Široký et al. (2011), where a large single university building in Prague was being studied for its energy-saving potential. The study showed that the retrofitted, insulated part of the large repeating building could reduce its energy use by 28% whereas the uninsulated part could only save 15%. This shows that not only is the amount of thermal storage important but also the rate of dissipation of stored energy, as the insulation helps maintain the stored energy.

An alternative would be to upgrade a building's thermal storage potential, preferably with minimal impact on the building's ongoing operation. One solution could be ice storage or geothermal storage coupled with photovoltaic and thermal solar panels to absorb free energy and use the geothermal borehole as energy storage. This combination was investigated by a team at ETH Zürich (Yang et al., 2015). These solutions can also be tested in simulations to assess their potential for demand shifting and dealing with a fluctuating supply.



### 2.2.2. Dynamic Energy Supply

The national and local energy supply fluctuates naturally. This can be due to renewables' varying outputs as the winds change, local to the building as a cloud passes in front of the sun, or due to human intervention such as a power plant going offline or the daily peak in demand when everyone is showering or cooking at the same time. In a future more reliant on renewables, these fluctuations may become even more important, despite the power supplier's efforts to even out the variability.

Time-of-use tariffs and peak tariffs are used as price incentives to smoothen out demand. That also explains why nearly half of all the papers reviewed by [Vázquez-Canteli and Nagy \(2019\)](#) included some form of dynamic energy pricing as an input or peak shifting as an objective. Peak shifting in itself can be important, as peak demand is often met by gas power stations, which are quicker to respond. However, it can lead to greater total energy use as shown by [Thieblemont et al. \(2018\)](#) whose project achieved a 10% cost saving but used 17% more energy. This may appear to be a potential failure, but if the energy used is renewable or has lower emissions, then a slight increase in total energy might still be a decrease in total greenhouse gas emissions. Another project by [Ma et al. \(2012\)](#) also used dynamic pricing, but there the researchers were able to decrease both cost and total energy use with a cost saving of up to 29% and energy saving of up to 23%.

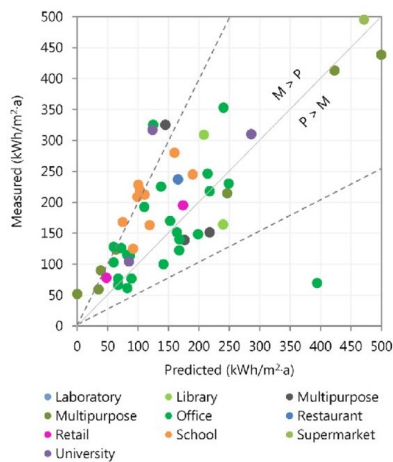
Dynamic pricing presents another opportunity for taking advantage of fluctuations, however, it is beyond the scope of this thesis, and only a flat fee pricing model will be used for the [DRL](#) agent's operation of the building.

### 2.2.3. Conclusion

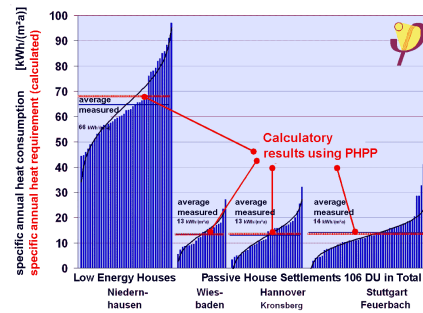
By taking advantage of the fluctuations in energy supply from the sun, occupants, and power grid the building can save energy and cost through smart predictive control. This has been done in countless studies, with many focusing on the variation in power supply prices. However, there is less discussion about which building fabric factors influence a building's potential for storing and shifting energy.

### 2.3. Building Energy Modelling

Energy use, as calculated in energy models, can be split into two primary categories; regulated and unregulated. Regulated refers to the geometry and building systems, whereas unregulated refers broadly to the electric loads from plugs, unexpected occupancy patterns, detailed occupant behaviours and other special uses. Large fluctuations in unregulated energy often stem from user-related activities. Energy models generally agree on the regulated energy, as these are rooted in physics and predictable, however, unregulated energy can have a disproportional impact on the overall consumption (van Dronkelaar et al., 2016). In addition, the workmanship on-site, changes in use after design, and poor commissioning of HVAC systems will cause the building to perform differently than anticipated. Multiple studies (van Dronkelaar et al., 2016) (de Wilde, 2014) highlight not only the lack of precision in predictions but also that the accuracy is skewed towards underestimating. This is true even for residential buildings which are much simpler and smaller (Merzkirch et al., 2014).



(a) Plot of measured over predicted energy use in buildings showing a trend to underestimate. Graphic from (van Dronkelaar et al., 2016).



(b) Plots of measured and predicted energy performance of low-energy and Passivhaus-certified buildings as predicted using the PHPP tool. The differences in measured use in the same type of building highlight the influence of occupant behaviour. Graphic from (Dr. Wolfgang Feist, 2023).

Figure 2.4.: Regardless of whether energy performance is predicted using common tools as in 2.4a or using more detailed tools such as PHPP in 2.4b, the measured results will always deviate due to differences in use.

Better predictions can be achieved through better modelling as demonstrated by the strict Passivhaus standard, however, this comes at the expense of labour and time. For instance, the shading on a single window consists of 6-8 geometric parameters as well as several parameters describing its installation and neighbour conditions. In addition, even the assumed number of toilet flushes is included as the influx of cold water in the cistern represents a heat loss (PHPP 9 (Passive House Planning Package), 2016).

For this thesis, the PHPP tool cannot be used, despite its use in commissioning the case study school, as PHPP only calculates the annual and monthly use. This thesis requires a more granular approach allowing daily and hourly intervals. There are countless energy modelling tools, each with its advantages and disadvantages, including PHPP, EnergyPlus, Modelica, TRNSYS, IES, DOE-2, and Tas to mention a few. These come from different public

and private providers and serve various parts of the wider industry. Architects and climate designers may also be familiar with DesignBuilder or Ladybug (part of Grasshopper, Rhino) but both of these are interfaces for the underlying EnergyPlus engine.

Due to the author's prior experience, EnergyPlus is preferred as it meets the requirements for this thesis and is free to use. It also allows for similar and sometimes better HVAC system control as the tools mentioned above (Bannister et al., 2011). EnergyPlus is a geometry-first approach to building modelling, whereas Modelica language takes a systems-first approach, making it more easily generalisable to different building geometries (Martinez-Viol et al., 2022). However, this thesis focuses on specific modelling of the whole building. Optimizing by taking advantage of unique geometric and local environmental factors. The downside of this approach is the higher level of specificity and level of detail required for the case study building simulation. A lighter system-based model has been demonstrated as showing similar predictions as measured on site by Martinez-Viol et al. (2022), but it relies on data to calibrate the building model, which presented three large flaws. Firstly, using only 15 days of data in 1-minute intervals may skew the model for that season, making it less useful as a full model. Secondly, their system-based model allowed the calibration to change geometric fundamentals such as floor area, window areas and U-values resulting in a model that performed similarly to but did not fully represent the actual building. Thirdly, and most importantly for using a geometry first approach, is that sufficient historical data may not be available, either due to the building being new or the systems not adequately recording the data. Therefore, a geometry-based model should be more accurate from the beginning, especially if calibrated using measured data.

#### 2.3.1. Digital Twins

Digital twins can describe a broad category of computer-based models simulating physical objects or systems. An energy model can be said to be a digital twin of the building, however, its usability and applicability depend both on the model and the use case. A Passivhaus, PHPP, energy model can be considered a digital twin, describing all the information of the building's performance, however, it works only for a monthly and yearly use-case. It is therefore not useful for day-to-day operations. The use case is also important, as an energy model usually does not contain information about traffic or detailed user movement in and around a building.

The utility of a digital twin is that an observation made within the digital model can be assumed, within the constraints of its accuracy, to be true for the physical object or system being represented. Calibration and validation of the model become important, otherwise, the digital twin cannot be relied upon to give any relevant or accurate information.

#### 2.3.2. Calibration

Calibration of energy models is used to bring the predicted values closer to the measured ones. This is important as some of the assumptions put into the energy model may be wrong or slightly different on-site due to various factors mentioned previously. One approach, which incorporates the uncertainty of the parameters is Bayesian calibration (Muehleisen and Bergerson, 2016), which was used in the above-mentioned system-based Modelica model by Martinez-Viol et al. (2022) and in (Chakrabarty et al., 2021). However,

## 2. Literature Review

Bayesian calibration can be computationally expensive (Muehleisen and Bergerson, 2016) hence other approaches such as genetic algorithms (Ramos Ruiz et al., 2016), slime mould optimization algorithms (Guo et al., 2021) or even deep learning with transfer learning (Gao et al., 2020) have been tried, primarily in the research literature. Calibrating models run into other issues such as the whole building model giving the correct results, but each individual zone of the building may be incorrect. This problem of calibrating at both the building level and zone level was explored by Yang and Becerik-Gerber (2015). Historical data may not only be limited in duration, but also in detail. This means that for calibration often the only available data is electricity use rather than detailed performances, inputs, and outputs of the system (Ji and Xu, 2015) making it more difficult to calibrate accurately.

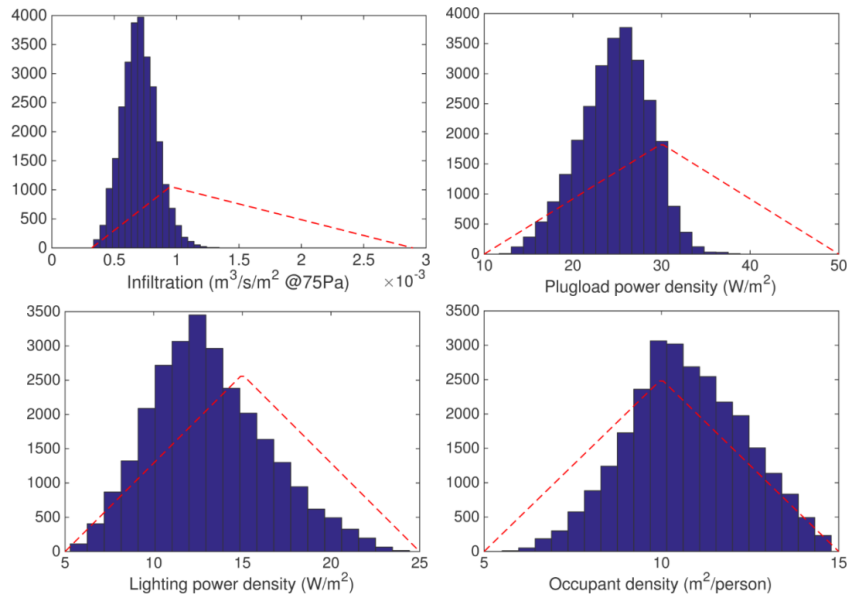


Figure 2.5.: An example from Muehleisen and Bergerson (2016) of how the calibration takes in a prior (dashed red lines) assumption of the factors being calibrated, and then calculates the posterior density function (blue bars) of the factors, in order to match simulated data to measured data.

There is usually a trade-off between the complexity of the model calibration and the time taken to calibrate, hence it is valuable to only focus on the most impactful parameters. A study (Muehleisen and Bergerson, 2016) of Bayesian calibration for buildings Muehleisen and Bergerson found the four factors; infiltration rate, plug load power density, lighting power density, and occupant density as being the most important. This is far fewer than the 45 parameters used in the system-based Modelica model by Martinez-Viol et al. (2022), however, this is likely because Muehleisen and Bergerson used OpenStudio (EnergyPlus) resulting in much slower computation times. The four factors above also represent key aspects of building use that are usually assumed to be fixed but can vary considerably in reality. The only outlier is the infiltration rate as this is part of the regulated energy. It can, however, vary due to workmanship (van Dronkelaar et al., 2016) and even detailing and component specification as emphasized in Passivhaus practice (McLeod et al., n.d.).

Models are calibrated against historical data measured on-site, such as energy consumption or interior temperatures. These data are often monthly, daily or hourly depending on the

type of data and the monitoring setup. The American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) has guidelines for calibrating energy models based on monthly or hourly data, giving an objective measure for validating the accuracy of the energy model. The measures include Normalized Mean Bias Error (NMBE) and Coefficient of Variation of the Root Mean Square Error (CV(RMSE)) between the measured data and the simulated data. When using hourly temperature data the NMBE should be less than 10% and the CV(RMSE) should be less than 30% (ASHRAE, 2002) via (Bokel, 2019).

In this thesis the ASHRAE guidelines are used to validate the digital twin energy model using the NMBE and CV(RMSE) measures. A genetic algorithm approach was also tested, however, this was not used in the final version and more information can be found in the appendix. From the base model further calibration was not required as the energy modelling was detailed enough from the beginning to meet the validation criteria.

#### 2.3.3. Conclusion

Building energy modelling is used to make a virtual representation of a building's energy performance. However, they can be biased, containing optimistic, outdated, or mistaken assumptions about the building as well as inaccuracies. The model quality also depends on the intended use of the model. Calibration can be used to bring a model closer to reality in the case where the building has been built and monitored. By doing this, the building energy model becomes a more accurate and reliable tool for prediction and analysis. This building energy model can then be used as a test-bed for simulating alterations and alternatives of the building to measure the performance difference of these scenarios.

## 2.4. Machine Learning

ML dates back to the previous century, but it is hard to pinpoint a single starting point and sometimes even to agree on the nomenclatures within AI, especially as they are creeping into daily language. One of the earlier books to describe machine learning is by Mitchell (1997), who described it as a system able to learn a task, based on experience given a performance measure. ML is a subbranch of AI, and DL a further sub-component of ML (Lotus Labs, 2020). Classifying what is and is not AI or ML can be difficult, especially as many companies incorrectly claim to be using AI in order to secure investment (HEC, 2019).

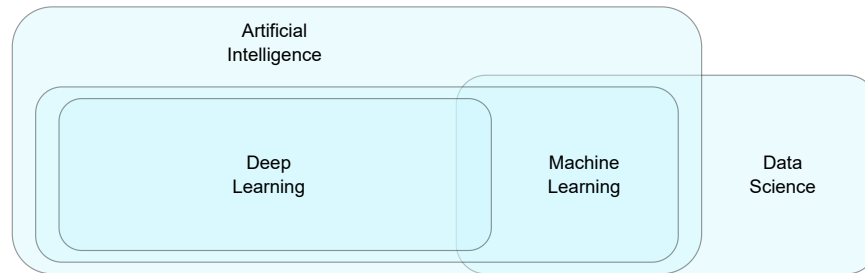


Figure 2.6.: Simplified overview of the taxonomy of AI, ML, DL, data science.

The three main branches of ML are supervised, unsupervised and Reinforcement Learning (RL) (Zhang, 2010). Supervised learning usually takes the form of having a training set of labelled data to train the model and another set of unlabelled data to test the model's performance on unseen data as a proxy for being able to generalise. Unsupervised learning is concerned with finding patterns in unlabelled data such as clustering or dimensionality reduction. Reinforcement learning is slightly different in that it uses an environment in which an agent is learning through trial and error to increase its performance based on a specified metric (Sutton and Barto, 2018).

### 2.4.1. Reinforcement Learning

Reinforcement Learning shares a primary characteristic with supervised and unsupervised ML in that it is a system that is able to learn a task, based on experience given a performance measure as outlined above in the book by Mitchell (1997). The first crucial difference, however, is that RL methods learn from interacting with an environment, rather than from existing data.

In this thesis, agent refers to the Reinforcement Learning agent, which is responsible for making decisions. The RL model describes the process used to train said agent. Therefore, later in the thesis, it will be assumed and said that; the agent is operating the building - which refers to the trained outcome of the RL model.

In RL the agent will interact with an environment. This environment represents everything that is available to the agent and can include seen as well as hidden information, which will have to be interpreted or inferred by the agent as it learns. In this thesis, the environment can be thought of as the building simulation, the operation controls and the weather. Only some of these can be influenced by the agent, but all of them are crucial in representing the environment. The agent is said to take an action, whenever it interacts with the environment.

This happens at every time step of the learning process. Continuing the same action also signifies taking an action. Depending on the system that is being modelled the time steps may only be for every action as in turn-based games like chess, but it can also be every second, minute or day depending on the system when applied to continuous time problems (Sutton and Barto, 2018).

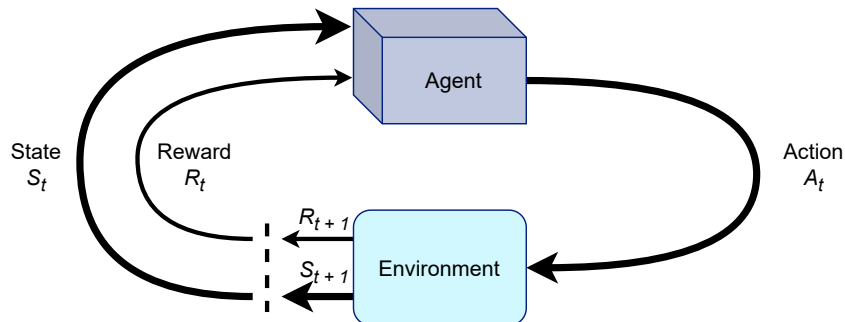


Figure 2.7.: An overview of the steps for the agent interacting with its environment. After each action, the environment outputs a new state and a reward to the agent. Then the agent decides the next action and the process continues. At each iteration the agent learns from its previous action and reward. Adapted from the book by Sutton and Barto (2018)

After each action, the environment returns a new state and reward to the agent and the process continues with the next action. The state refers to the new state of the environment. This is also illustrated in Figure 2.7. In chess that new state would be the new layout of the chess board after the agent's move and the opponent's subsequent move. The reward is more difficult to define, as it has to be specified by the programmer and is not always obvious from the outset, as delayed rewards can lead to long learning times. The reward serves as the signal for performance measures, but if that signal is sparse, the speed of learning becomes slower and the potential for correct learning becomes slower. That is because if the reward is delayed, or is due to multiple prior moves being correct, it is difficult to infer exactly what made that sequence of actions correct and what part of the sequence could be improved (Russell et al., 2010).

For example, for an agent learning chess, if the reward function was such that the agent would be rewarded with 1 point after each win, 0 for a draw and -1 for a loss, the agent would learn very slowly as there would be no reward signal after every move. Alternatively, the agent could receive 0.1 points for each round it is still alive and -0.5 and 0.5 points each time it loses a piece or takes a piece from the opponent respectively. This would be a better reward signal as it is less sparse, but can also have adverse consequences. For instance, the agent may start dragging out the game hundreds of moves, simply because staying alive is now more rewarding than winning in the long term. This makes the reward function the most difficult part to engineer, as it should provide learning feedback for the agent while also representing exactly what it is desirable that the agent learns. Unlike the environment, the state and the action, this is also the primary factor that can be influenced by the programmer besides the agent's model architecture.

The agent chooses an action based on its policy or value function. If the agent's model architecture uses a policy it means that for each state there is an explicit mapping to a single action. Using a value function, on the other hand, assigns each possible action with a value and the agent performs the action with the highest value. Most models also include

## 2. Literature Review

early exploration, which will force the agent to make a random action in order to explore new and untested actions early on in the learning process (Sutton and Barto, 2018). This is also shown in Figure 2.8, where a higher level of exploration  $\epsilon$  leads the agent to choose better actions in the long run as it explores different actions. In a stationary environment, exploration can decrease as the agent starts to have explored all possible actions, but in a dynamic environment where the conditions change exploration must remain a core part of the agent.

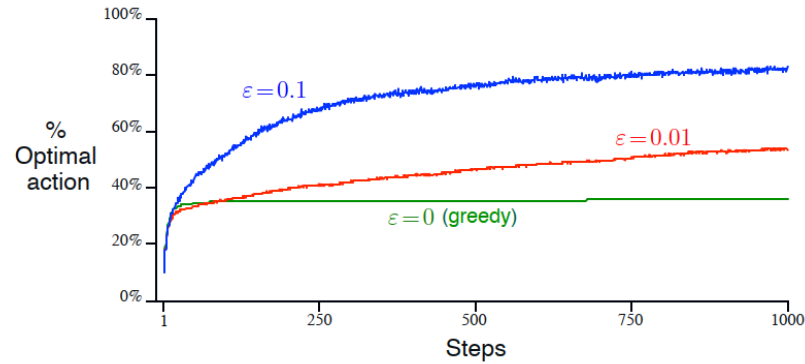


Figure 2.8.: Diagram of the optimality of actions chosen by an agent given different levels of exploration in the model. At 0 exploration, greedy green, the agent will not progress over time as it does not explore alternative actions. From the book by Sutton and Barto (2018)

Reinforcement Learning can be either model-based or model-free. A model-free agent only experiences the rewards coming from the environment, and must then update its value function based on the reward signal. This is done by observing the difference between the expected reward for the given action and the real reward received and moving the expected reward value closer to the real reward. For model-free agents updating the expected values based on the rewards is the main way of learning. In model-based RL the agent has a model of the environment and is able to estimate the future state given an action. This helps the agent plan as it can model future outcomes, with uncertainty, even if it has not seen that state previously. This is in opposition to a model-free agent, which only learns from previously explored actions (Sutton, 1988).

### 2.4.2. Neural Networks

Neural Networks (NNs) are inspired by how the brain's neurons and synapses operate by receiving an electric signal and triggering if a certain input threshold is passed. A single neuron is rarely used, but when put together in parallel and series, also referred to as the width and depth of the network, these single neurons can be used to perform classifications, regression, and function approximation (Kinsley and Kukiela, 2020). Using Neural Network is also commonly referred to as Deep Learning (DL).

Convolutional Neural Networks (CNNs) are a type of NNs that not only use layers of neurons but also use convolutional layers, sometimes called filters, as well as pooling layers. This reduces the computational complexity of an image by distilling its pixels down to a lower number of features. This reduces the network size and calculation time while maintaining



performance (Goodfellow et al., 2016). Different techniques for specific applications are constantly being invented and CNN is just one of many such techniques.

Neurons can be programmed to trigger in different ways. Each neuron has a weight and a bias. The weight is a scaling factor and the bias is constant, which is added to the product of the weight and the input. The value is then put through an activation function, which determines the output of the neuron. Some activation functions such as Sigmoid output only 0s and 1s, whereas the more commonly used Rectified Linear Units (ReLU) can output any value from 0 to  $\infty$ . While ReLU is the most commonly cited activation function, there are still newer ones being developed, which perform better on most tasks. One such activation function is Gaussian Error Linear Units (GELU) (Hendrycks and Gimpel, 2020). In this thesis ReLU is most likely to be used as it is found in most standard libraries and has a longer track record of reliable use.

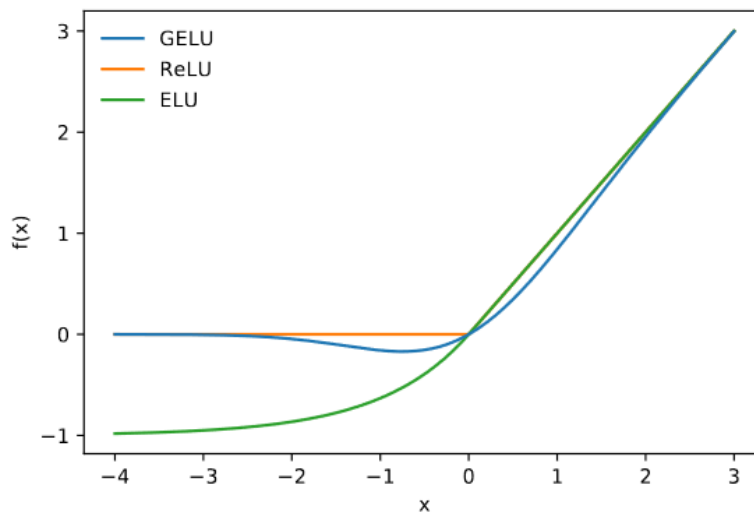


Figure 2.9.: Graph comparing the GELU, ReLU and ELU functions. From (Kon, 2022)

A NN learns from its errors through a process called backpropagation. As the name suggests the error is propagated backwards through the network to adjust the weights and biases of each neuron to bring them closer to the correct result for that particular sample data (Kinsley and Kukiela, 2020).

Another area of investigation is the effects of the width and depth of a network. It is commonly accepted that improving the size in either dimension will generally increase the accuracy - though overfitting can become more of an issue. In a paper by Zagoruyko and Komodakis (2017) they point out that increasing the size does improve performance, however, a wider network can be trained faster than a deeper network as modern GPUs can train the whole width of the network simultaneously. The drawback is that wide networks may have many residual neurons that become obsolete and do not contribute to the actual performance or output of the network. Researchers at Google (Nguyen et al., 2021) are also investigating the differences of large versus small models and specific tools such as Weight-Watcher (Martin et al., 2021) attempts to analyze objectively whether a network has been sufficiently trained and which layers are contributing the most.

### 2.4.3. Deep Q-Learning

Neural Network and Deep Learning can be combined with Reinforcement Learning to create Deep Reinforcement Learning (DRL), of which one type is Deep Q-Learning (DQN). This differs from standard RL in that instead of a look-up table of policy (Q) actions or expected values for each state, it uses a neural network to estimate the function. This means it can infer knowledge from seen scenarios onto similar but unseen ones (Simonini and Sanseviero, 2022)(Williams, 2022). This is shown in Figure 2.10, where a tabular and NN example are compared. The visualisation of tabular examples primarily works in simple and low-dimensional examples, where there is a linear progression to each state. In reality, most problems are more complex than that, but the tabular examples show a simplified version of linear state progression. The Neural Network (NN) model of state to action can be more intuitive as the model inputs include multiple factors about the state that are weighted and calculated to produce expected outcomes for each action through the network.

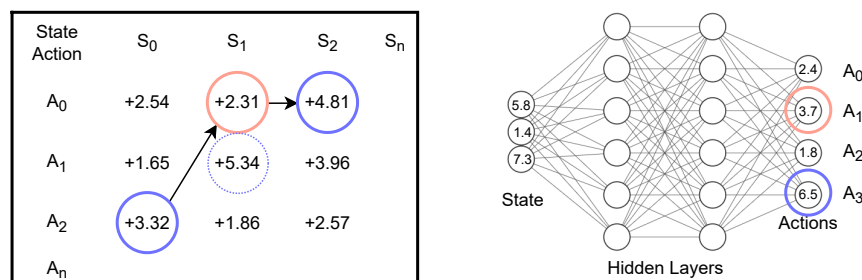


Figure 2.10.: The diagram shows a tabular example of a RL agent’s value function on the left and a neural network example on the right. S denotes the state and A denotes the action. In the tabular example, each state and action is mapped to a predicted value shown as a number. The agent will pick the highest expected value shown encircled in blue. However, it may be forced to a random choice encircled in orange. In DRL the table is replaced with a Neural Network (NN), which can simplify complex problems as the possible state space grows very large. There rather than each state and action having particular matching values, the state serves as the input to the Neural Network (NN), outputting expected values for each possible action.

### 2.4.4. Recurrent Neural Networks

In ML a common data category is time-series data. This includes data related to regular monitoring such as weather data, but can also be abstracted to text as a stream of ordered data. Time-series data does not work well with the standard Neural Network structure of a fully connected network, because there is no order to the inputs. Recurrent Neural Networks (RNNs) were designed to overcome this problem and have been improved upon with Gated Recurrent Units (GRUs) and Long Short-Term Memorys (LSTMs).

Figure 2.11 shows a simple overview of the RNN’s recurrent structure. Each vector from each batch of data is inputted one by one. First  $X_0$ , which is the first item. This is put through the RNN unit A, the result of which then forms part of the input of the next RNN unit together with vector  $X_1$ . Figure 2.12 shows a detailed view inside the RNN unit. The

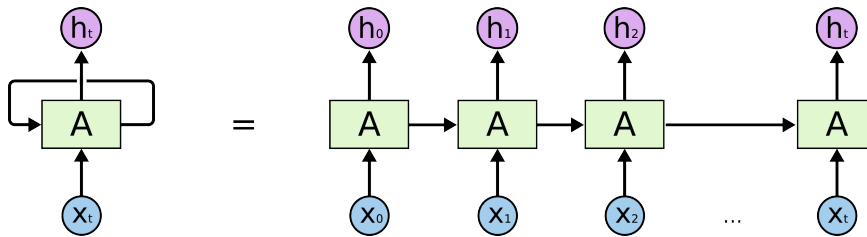


Figure 2.11.: Illustration of the repeating structure of the RNN. Diagram by Olah (2015)

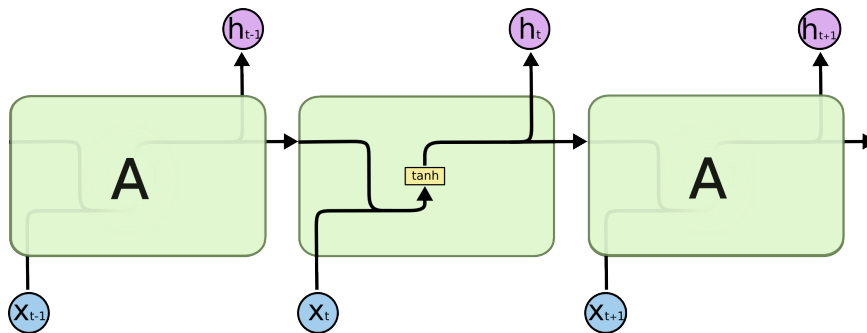


Figure 2.12.: Detail of RNN and the NN with a tanh activation function. Diagram by Olah (2015)

inputs are concatenated before being passed through the single-layer NN with a tanh activation function. Some of the main problems with RNNs are vanishing gradients when backpropagating through long networks, as well as the ability to maintain memories over long sequences (Pascal Poupart, 2018). LSTMs overcome the memory problem by running two connections in parallel with gates dictating the amount of information passed through. Figure 2.13 shows a detailed view of the LSTM unit. The top line is the more persistent memory, which is relatively unchanged as the new input  $X_t$  has to pass through NNs with Sigmoid activation functions, whose output is in the range 0-1, and update the persistent memory in proportion (Olah, 2015). These NN layers are trained through backpropagation on training data.

The final outputs of a RNN can be fed through a standard NN and used for prediction or classification tasks.

### 2.4.5. Curse of Dimensionality

One of the main curses of dimensionality is computing time. As seen above in energy modelling, the complexity of the problem can be a hindrance to performing the desired calculations or optimisations within a desirable time frame. This can be done simply by reducing the parameters being calculated as in (Muehleisen and Bergerson, 2016), where only 4 key parameters were chosen. However, ML methods for dimensionality reduction such as auto-encoders can be used to obtain better results than discarding potentially useful though lesser significant parameters. Despite their usefulness, a meta-study of RL in building control by Vázquez-Canteli and Nagy (2019) reveals that few studies have included auto-encoders

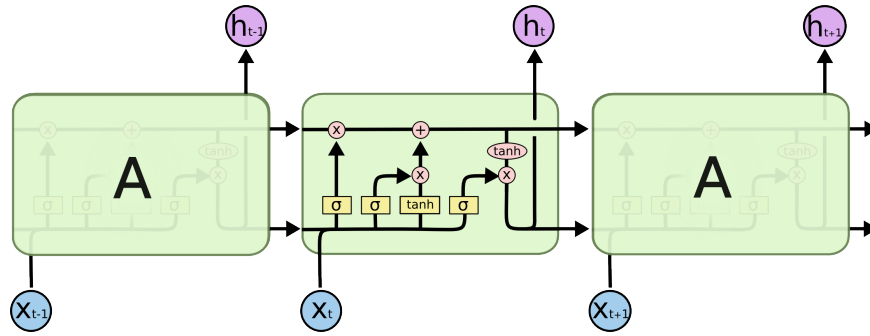


Figure 2.13.: Detail of LSTM and its multiple NNs with tanh and Sigmoid activation functions. Diagram by Olah (2015)

or other dimensionality reduction methods, despite approaching issues of increasingly high dimensionality in their RL models.

Overfitting is another curse of high dimensionality and this is illustrated in Figure 2.14, by Yiu (2021), where there are as many dimensions as there are observations.

#### 2.4.6. Surrogate Models

Instead of relying on the sometimes slow calculations of energy or solar models, surrogate models can be used to approximate a complex model and be trained with calculated data to give similar results to the original models (Attar et al., 2016). This can speed up calculation time considerably but does require a longer setup and training period of the surrogate model and it can only be used within the domain for which it has been trained such as urban wind analysis as done by the company OrbitalStack (2023). Surrogate models do need to be checked for accuracy compared to the traditional full calculations before they can be relied on. With the right modelling and training data they can achieve accuracies of 98% or higher (Ekici et al., 2019). However, different modelling approaches do have an impact on the performance and accuracy, which was explored by Li et al. (2021) who found that using a long-short-term memory model was better than using a multilayer perceptron model for creating a surrogate EnergyPlus model. A surrogate model is also a good mental model for how model-free DRL works, where the Neural Network becomes the computation model of the agent to plan its actions in the environment.

#### 2.4.7. Challenges in Deployed Machine Learning Applications

One of the main challenges of ML deployed on real-world problems is data drift such as when the initial training data is too optimistic, narrow, or the problem changes over time (Sculley, 2022). In addition to this RL in the real world also suffers from high dimensionality of continuous action and state space, safety constraints that should not be crossed, poor explainability of the policies, and large delays in sensors, actuators or rewards (Dulac-Arnold et al., 2019).

	Reddish	Bluish
●	1	0
●	1	0
●	1	0
●	1	0
●	0	1
●	0	1
●	0	1
●	0	1

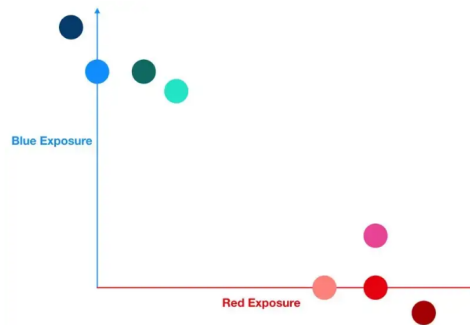
(a) Two dimensions. Clear separation.  
Graphic from (Yiu, 2021)

	Red	Maroon	Pink	Flamingo	Blue	Turquoise	Seaweed	Ocean
●	1	0	0	0	0	0	0	0
●	0	1	0	0	0	0	0	0
●	0	0	1	0	0	0	0	0
●	0	0	0	1	0	0	0	0
●	0	0	0	0	1	0	0	0
●	0	0	0	0	0	1	0	0
●	0	0	0	0	0	0	1	0
●	0	0	0	0	0	0	0	1

(b) As many dimensions as samples leads to overfitting and no useful abstraction information.  
Graphic from (Yiu, 2021)

	Red	Blue
Red	1.00	0
Maroon	1.20	-0.10
Pink	1.00	0.20
Flamingo	0.80	0
Blue	0	1.00
Turquoise	0.25	0.90
Seaweed	0.15	1.00
Ocean	-0.10	1.20

(c) Each lower level dimension's exposure to the higher level abstracted dimensions Red and Blue. Using a vector instead of a Boolean value.  
Graphic from (Yiu, 2021)



(d) Samples plotted along primary features.  
Graphic from (Yiu, 2021)

Figure 2.14.: These four diagrams show an example of a dimensionality problem. In (a) there are two dimensions, which allows us to cluster the samples according to which dimension they are closest to. In (b) there are 8 dimensions. However, this leads to overfitting of the data and no valuable abstraction can be made. In (c) the lower-level features (dimensions) are mapped according to their exposure to the higher-level features. The result is plotted in (d). The step from (a) to (c) allows us to maintain more detailed information about each individual sample, while still being able to abstract out information about them as a group. Graphics from (Yiu, 2021).

Note: Features, dimensions, and categories can be used to refer to the same thing depending on the context.

## 2.5. Building Control

This section gives an overview of building operation methods as well as a look into the research already conducted by other researchers in the field of advanced building control.

### 2.5.1. Building Operation

Modern buildings are primarily controlled through ventilation and heating/cooling systems. These come in many combinations and configurations. In a simple setup, there may be a ventilation fan supplying fresh air to the building and a separate radiator heating up the room. These can also be combined such that the air is preheated, or cooled. This removes the need for local radiators as the air now provides temperature control too. However, this does have several limitations in terms of capacity and distribution. The next step would be to recover the energy, creating a closed loop of ventilation using a mechanical ventilation with heat recovery (MVHR) system. This saves energy for heating and cooling because the indoor air's energy is recovered before it is expelled to the outside. This describes the bare minimum of simpler systems but gives an idea of the many options available.

Traditionally systems were controlled by simple on/off switches. A simple example is a thermostat, likely controlled by a bi-metallic strip, which turns the whole system on or off depending on the settings and input values. This works well as an automatic system, however, it does have several limitations. For instance, it only controls the on/off setting, meaning that the radiator or ventilation fans will either be off or at full power. To solve this a proportional-integral-derivative (PID)-controller can be used. The PID-controller works by analysing the current state and desired state and finding the most suitable power setting to reach the desired state. It is proportional in its power setting, meaning it will only increase the power a little if the threshold is passed, whereas a simple rule-based controller will always ramp up to full power in the on-state. The derivative and integral components of the PID also ensure that it does not oscillate or overshoot as it will gradually decrease power output as it gets closer to the target value. However, neither the rule-based nor the PID controllers take into account future changes and are therefore not predictive. For that, we need more advanced control algorithms.

### 2.5.2. Advanced Building Control

This subsection is about the efforts that have gone into more advanced and sophisticated means of building operation control. From the literature, some are concerned only with the mechanical operation and focus only on the HVAC systems (Martinez-Viol et al., 2022), whereas others look directly at peak-demand shifting (Thieblemont et al., 2018) (Luo et al., 2020), which is related to the previous section, and others still focus on whole building simulations (Široký et al., 2011). The methodologies also vary greatly, but a general trend is a move from model predictive control (MPC) towards DRL.

#### MPC vs DRL

Figure 2.15 shows diagrams of MPC and DRL side-by-side. MPC can be compared to a tree search type algorithm, exploring all the branches, subbranches etc. to a certain depth. This

can be improved using beam search, where some branches are abandoned to save processing - this is illustrated by the dotted lines in figure 2.15a. Once it has found the optimal long-term action in the tree it will proceed one step ahead and either recalculate the branches or reuse some for the following step. A trained DRL agent on the other hand only does a single calculation to arrive at the optimal action, however, it has to train over thousands or millions of samples in order to *learn* the correct behaviour and store that in the Neural Network (NN). In general terms MPC is explicitly programmed by experts to simulate a system, whereas DRL is a more generalist model which learns from vast amounts of data.

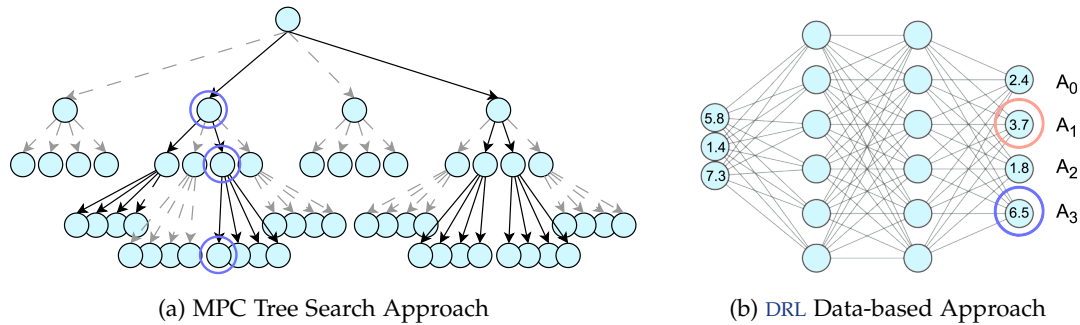


Figure 2.15.: MPC uses variations of what can be likened to a tree search to find an optimum action, whereas DRL *learns* the optimum action based on experience derived from data

## History

Model predictive control MPC is a process adopted from the chemical and industrial sector into the building sector (Ma et al., 2012). With this approach energy savings of 15-30% have been achieved in the past decade (Cígler et al., 2013) (Široký et al., 2011). Through this literature review, a trend is that MPC is being replaced or challenged by RL and ML approaches. MPC and RL have a large overlap and similar features such as state calculation, dynamic optimization, and learning (Arroyo et al., 2022). MPC originates from the field of control theory whereas RL comes from the field of Artificial Intelligence, but both can provide optimal control with proper use (Lin et al., 2021). A DRL agent may beat an MPC system if the MPC's horizon is too short (Arroyo et al., 2022), but the MPC is better at generalizing when moving beyond the training dataset. However, the DRL performs better if there are large errors in the input data (Lin et al., 2021) (Hasankhani et al., 2021). One major drawback of MPC is the required model, which is the key part of the prediction, as it simulates the future steps. Hence, a poorly formulated or improperly calibrated model will yield inferior results. Model-based DRL can be susceptible to the same fate, but model-free DRL can potentially overcome this at the expense of longer training times in deployment (Yang et al., 2015). This also relates to a final difference, in that RL agents cannot be guaranteed to work, instead, they need to be validated through simulation or deployment, whereas the MPC's reliance on a model ensures correct, though not optimal performance (Suryana, 2020).

A linear quadratic regulator, which preceded the MPC in control theory, works similarly to an exhaustive tree search; calculating and thereby predicting the optimal strategy for each time step. MPCs are more advanced in that they are able to update the tree search at each time step, but use a shallower tree. This makes them capable of responding to a changing environment and also more robust to accumulation of errors (Wang, 2009, p. xii).

## 2. Literature Review

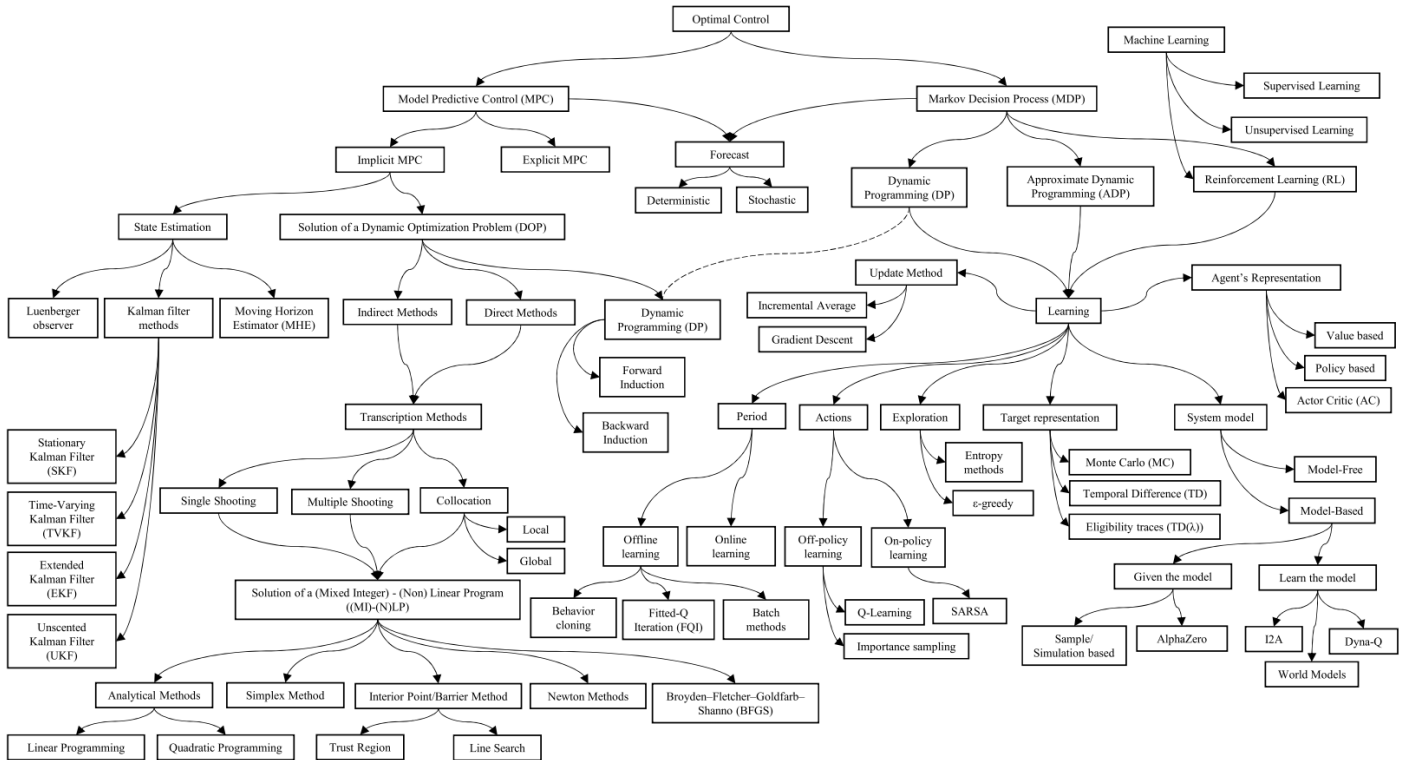


Figure 2.16.: Non-exhaustive chart of the taxonomy of optimal control. From the paper by [Arroyo, Manna, Spiessens and Helsen \(2022\)](#)

The use of Neural Network in building control dates back to 1998 when [Mozer](#) connected a residential house to a centralised control system governed by a NN. This included lighting, speakers, hot water and thermal controls, however, the experiment was unable to make significant savings to general energy use, only to lighting use through sensor response.

### Today

The research in this field of smart building control continues and has also sparked commercial interest at various consumer and business levels. In academia, multiple projects are ongoing and whole research groups are dedicated to this goal. Examples include the [Automatic Control Laboratory's Building and Energy division at ETH Zurich](#) and the [Brains 4 Buildings \(B4B\) at TU Delft](#). The results from academia also translate into commercial start-ups such as the Spanish [The Predictive Company](#) founded by researchers at the Polytechnic University of Catalonia. The consumer market is dominated by smart thermostats from companies such as Google Nest, Tado, Honeywell, Padoova and many more, some of which offer additional *smart home* features. Large-scale building management providers such as ISS Facility Services are also providing smart building management with various monitoring and predictive features. However, as it is still a nascent field, there are still many questions to be answered by academia.



From the surveyed literature two major trends, energy and control, emerge for a building's potential for benefitting from smart control. These can be split into further sub-categories:

**Energy Storage Potential** is the amount of energy that can be stored by a building, both in thermal heat capacity, batteries, or even geothermal storage in some cases. This depends on the construction and additional systems installed, some of which can be upgraded but is mostly considered static and fixed.

**Supply Energy Potential** is the amount of incoming energy and energy variability. This is both from the sun in terms of solar radiation on windows or solar panels, but also variations in the energy supply in terms of pricing differences, which act as a supply indicator.

**(Thermal) Dampening** describes the level of control or delay in the dissipation of the stored energy. This is primarily in terms of insulation on a building, which dampens the exchange of energy with the surroundings. Being able to keep the gained solar heat for longer is very beneficial. Geothermal storage also had a loss rate in storage, and so do batteries, however, that is less significant compared to a building's losses.

**Level of Operational Control** is the final piece, describing what is possible to control within or outside the building. At a minimum, this would likely include temperature settings and on/off switches for HVAC systems. However, it can become more complex by changing the local zones in the building to operate individually and also external systems like movable solar shading can have a significant impact because it can directly control the incoming solar radiation. Equally, window opening for nighttime purging can be automated and have significant impacts on the cooling requirements.

Figure 2.17 shows a handful of studies from literature plotted on a graph with *control* on the x-axis and *energy* on the y-axis. These are the trends discussed above and the graph shows that the higher the energy and control, the larger the saving (size of circle) of that building. A minor disclaimer is that these are only a few samples, and due to varying methodologies and sparse information on the specific building cases, these are just approximations. Going into detail, the two studies of the Czech Technical University by Cígler et al. (2013) and Šíroký et al. (2011) show how insulating the building (increasing control) leads to greater savings and how taking into account solar radiation (increasing potential energy) more precisely leads to larger savings too. In addition, the smallest saving achieved by the Icade Premier building is likely due to it having very little energy storage capacity because it is a lightweight timber frame building. The LowEx Building studied by Yang et al. (2015) is also lightweight, but includes additional geothermal storage, increasing both control and energy storage, leading to higher energy savings.

2. Literature Review

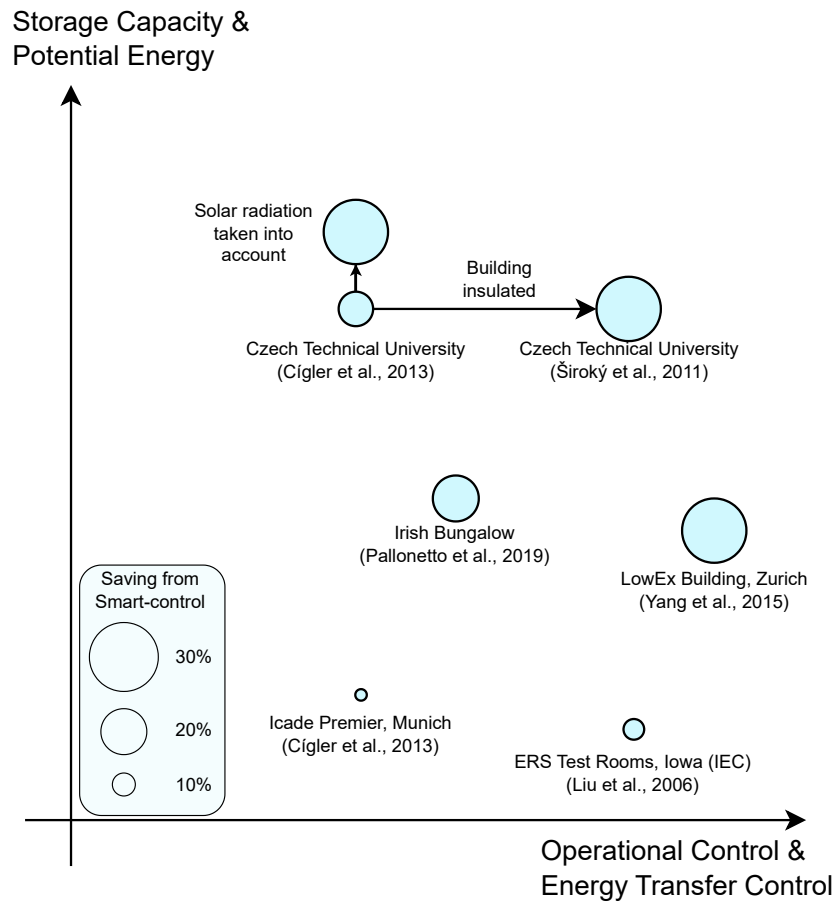


Figure 2.17.: Plot of the level of control and energy storage potential of the buildings investigated in the literature. The circles' diameter indicates the level of savings achieved for each building when using smart or AI-type control algorithms for the simulated operation of the buildings. This diagram can be displayed with 3 or 4 axes but has been simplified for a 2D view to differentiate only between energy and control, however, there is an overlap between energy and control in certain aspects of the buildings.

While many studies have looked at single case studies, few besides the ones mentioned have looked at the same building from multiple angles or with different control methods. There is a lack of understanding of which building fabric factors have the largest influence, what the impact of different control algorithms is and how these two crucial areas add up to a new paradigm of what makes a building low-energy for the future of AI-control. Imagine a **low-energy building with standard control** performing worse than a **normal building with smart control**. Taken further, what about a contemporary low-energy light-weight building, being outperformed by a normal building with more storage potential, when both are using smart control!



## 2. Literature Review

structure of the [DRL](#) system.

### 2.5.4. Conclusion

There are hundreds of studies concerning the smart control of buildings, all of which use different or slightly different methodologies, inputs, and assumptions. However, few of them investigate how different buildings compare to each other and how different smart controls on the same building compares. How the building fabric, such as insulation, influence the energy-saving potential and how there may be a difference between the saving achieved in standard control and the savings achieved with different smart controls.

### 3. Case Study: Welsh School

The case study was chosen as it is a highly controlled and certified low-energy building following the Passivhaus standard. That means there is a high level of certainty about the estimated and real performance of the building, backed up by on-site measurements.


<b>Location</b> Wales, United Kingdom	A timber-framed building with cellulose insulation, built on a concrete slab. It houses	<b>U-values</b> Walls and Roof: $0.105 \text{ Wm}^{-2}\text{K}^{-1}$	2,754 m <sup>2</sup> GIFA (1,992 m <sup>2</sup> TFA)	
<b>Architect</b> Architype UK Ltd.	14 classrooms, a hall, kitchen, nursery, and community rooms.	<b>Floor slab:</b> $0.179 \text{ Wm}^{-2}\text{K}^{-1}$	<b>Airtightness @50 Pa</b> 0.255 ACH $0.38 \text{ m}^3\text{h}^{-1}\text{m}^{-2}$	
<b>Passivhaus-certified</b> Primary School Ages 4-11	455 Occupants	<b>Windows (installed):</b> $0.76 \text{ Wm}^{-2}\text{K}^{-1}$	2.67 ACH via MVHR	



Figure 3.1.: Case study school exterior view, by Lowfield Timber Frames

3. Case Study: Welsh School



Figure 3.2.: Ground floor, first floor and section drawings of the school, by Architype



Figure 3.3.: Case study school aerial view, by Powys County Council



Figure 3.4.: Case study school site plan, by Architype

## 3.1. Digital Twin

The digital twin of the school is created as an EnergyPlus model in order to model energy use and internal comfort. This section describes the workflow of creating the EnergyPlus model. See the appendix A for a glossary of select key terms.

## 3.2. Methodology

### 3.2.1. Software Tools Descriptions

Below is a brief outline description of each piece of software or tool.

**SketchUp** A simple and versatile 3D modelling software ideal for hard surface modelling, but struggles with advanced curves beyond simple arches. SketchUp hides geometric complexity behind simplified interfaces making it user-friendly, but can also lead unpredictable behaviour for inexperienced users.

**Rhino** A more advanced 3D modelling software with multiple geometry definitions such as Nurbs making it superior for complex curves surfaces.

**Grasshopper** A visual programming (node-based programming) interface for Rhino. This allows for parametric definitions of models as well as interaction with other plug-ins and more advanced programming features.

**Ladybug & Honeybee** Part of a family of tools for Grasshopper for building energy modelling, comfort analysis, weather, urban analysis and more. Honeybee is specifically designed to translate to EnergyPlus via OpenStudio from within the Grasshopper environment.

**EnergyPlus** A building simulation package developed by [ASHRAE](#) and skilled volunteers. The software is continuously being improved upon and new features added.

**Python** A high-level programming language.

**PyTorch** A library of additional functions for Python containing numerous tools for [ML](#) and [AI](#).

### 3.2.2. Digital Twin Creation Workflow

The simplified workflow for this thesis is shown in Figure 3.5. The digital twin of the building is first created as a 3D model in SketchUp. This was done in SketchUp as it is far faster for simple hard surface modelling with customised shortcuts compared to Rhino. The model was built from the construction drawings of the school. The 3D model is then transferred to Rhino in order to be imported into Grasshopper. In Grasshopper the Ladybug and Honeybee tools are used to define the thermal properties of the building as well as its operation. The energy model is then translated to EnergyPlus before being simulated from within Python via the [API](#) in order to use the [EMS](#) tools together with [ML](#) tools from PyTorch. Figure 3.6 shows the building energy model of the school.



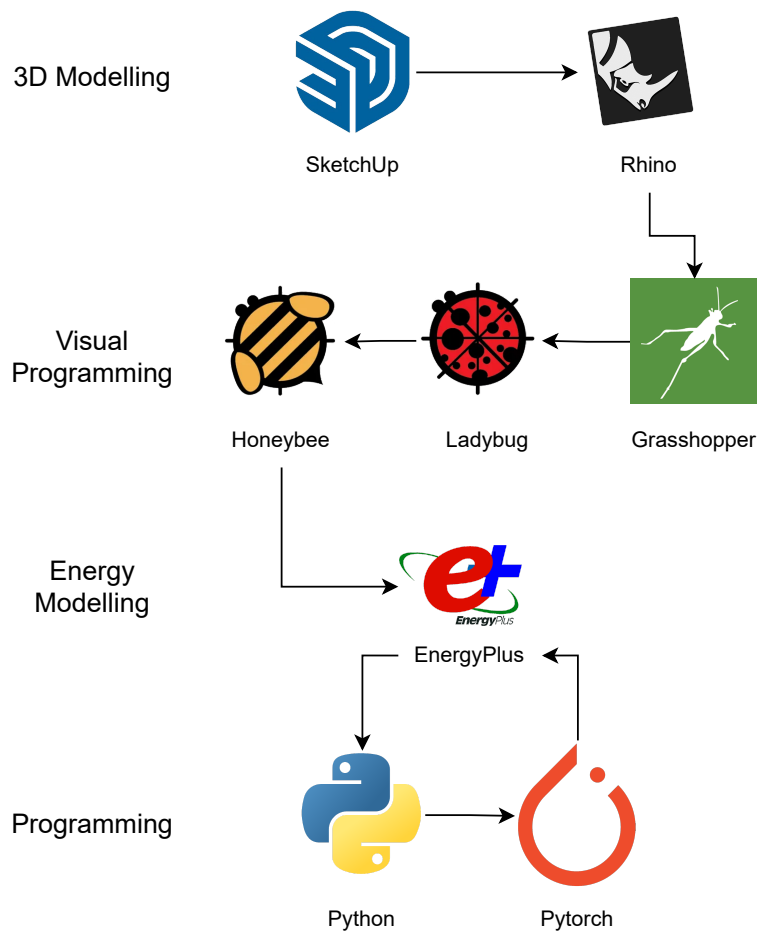


Figure 3.5.: Diagram of software used and flow of information

### 3. Case Study: Welsh School

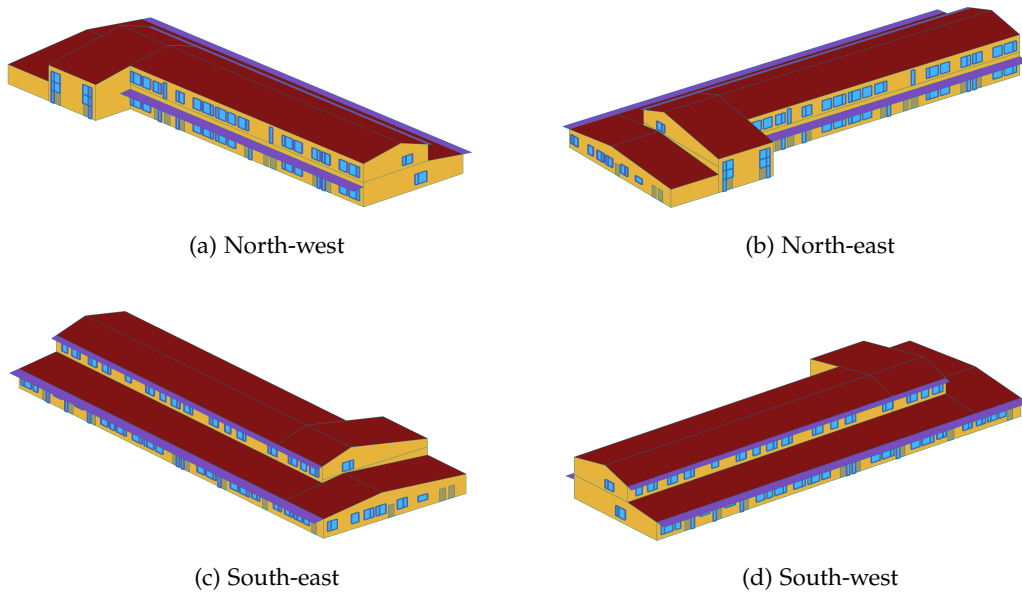


Figure 3.6.: Grasshopper - Honeybee 3D energy model of case study

## 3.3. Digital Twin Documentation

This section describes the digital twin model in detail as it was created in its final version.

### 3.3.1. Building Volume

The building was modelled with 2 zones. 1 for the ground floor and 1 for the second floor as shown in Figure 3.7. In order to simplify the building surfaces in the model the first floor is 250 mm higher on the facade, because of how the lower roof hits the south facade of the first floor. Secondly, the two tall windows on the north facade to the staircases have been shifted up by 500 mm as the windows cannot be split across two separate zones. These adjustments were made to simplify the zoning, without impacting the combined dimensions and heat loss areas.

Figure 3.8 shows how the thermal envelope of the model is defined as Honeybee surfaces in Grasshopper. There are two types of windows defined. Standard fixed-glazed windows and timber opening louvres for ventilation.

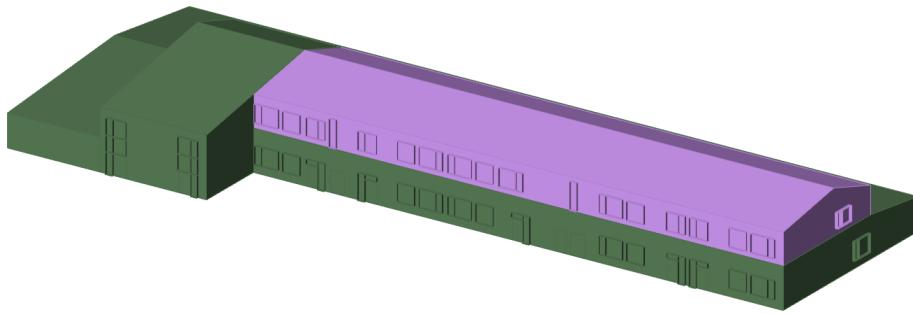


Figure 3.7.: Grasshopper - Zoning. Zone 1 ground floor and zone 2 first floor

3. Case Study: Welsh School

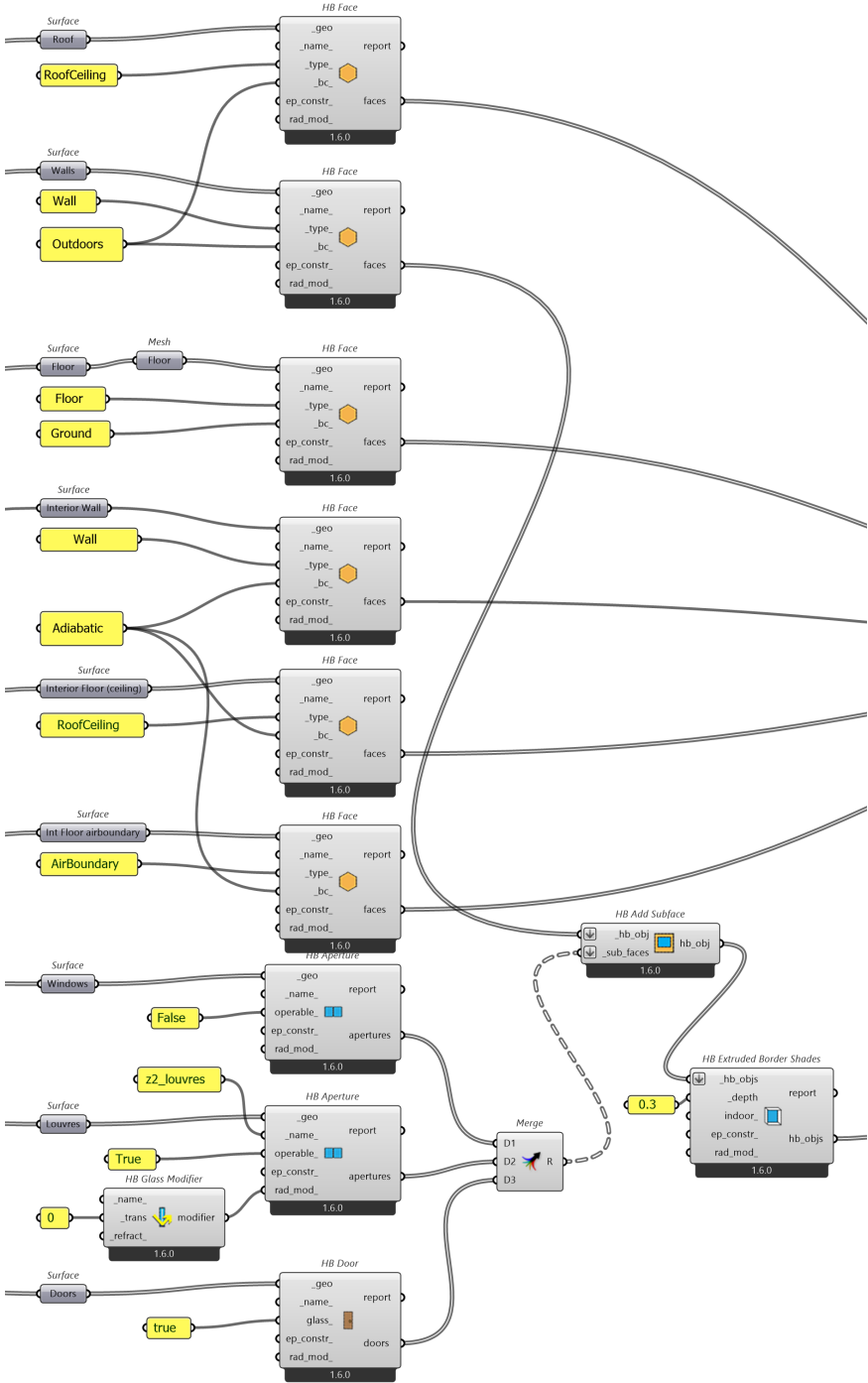


Figure 3.8.: Grasshopper - Energy model surface definitions

### 3.3.2. Thermal Mass

The building is a lightweight timber structure made with pre-fab insulated panels using engineered timber I-joists for minimal thermal bridging. This means the external envelope has a low thermal mass. The ground floor is a concrete slab cast on top of rigid insulation, providing some thermal mass to the building as a whole. In Honeybee the external envelope elements are assumed to have thermal mass based on their construction definitions. However, inside the zones, the additional thermal mass has to be defined separately using the *HB Internal Mass* components. The internal walls are measured from the construction drawings and used to specify the total area of internal walls. The construction consists of a timber frame covered with cement boards. In addition, for the ground floor zone, there are two concrete staircases which have been added in for their thermal mass. This can be seen in Figure 3.9.

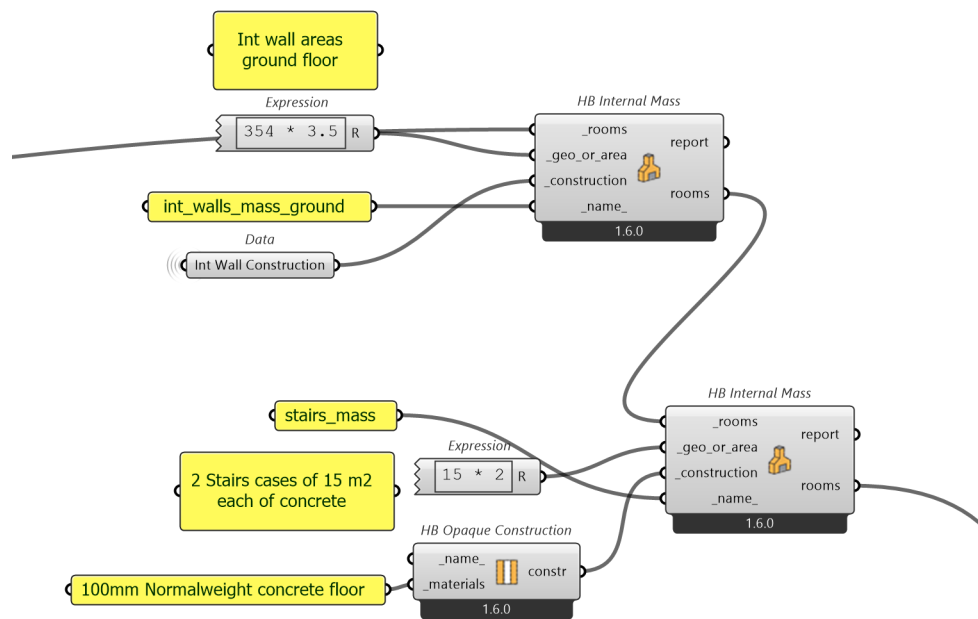


Figure 3.9.: Grasshopper - Internal mass, ground floor

### 3.3.3. Construction Build-ups

The constructions of the thermal envelope have been created to match the on-site construction and the specification from the Passivhaus design of the school. However, EnergyPlus does not support linear thermal bridges  $\Psi$  nor point thermal bridges  $X$ . Therefore, the U-values or thermal resistance of the wall must contain this information as a single weighted average figure. The benefit of a Passivhaus design is that thermal bridges are generally limited as much as possible and only represent 1-5% of the total envelope losses.

The constructions were defined in Honeybee based on standard build-ups with modifications to the specification where necessary. The insulation values were adjusted to both match the impact of the minor thermal bridges as well as to test the impacts for calibrating the model to match the measured data. Figures 3.10, 3.11, 3.12, and 3.13 show the Honeybee definitions of the constructions.

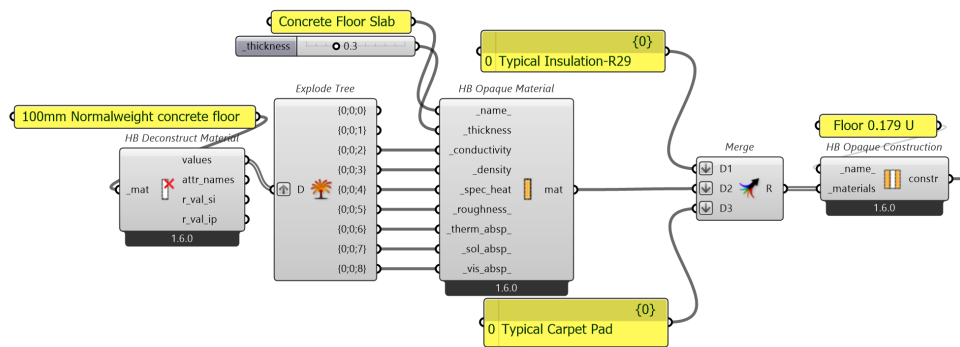


Figure 3.10.: Grasshopper - Floor construction

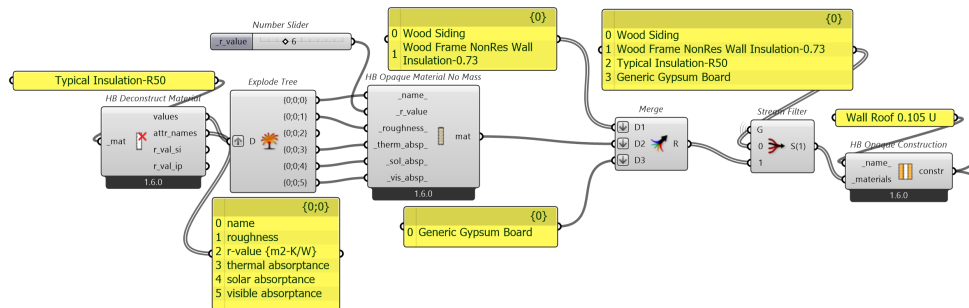


Figure 3.11.: Grasshopper - Wall and roof construction

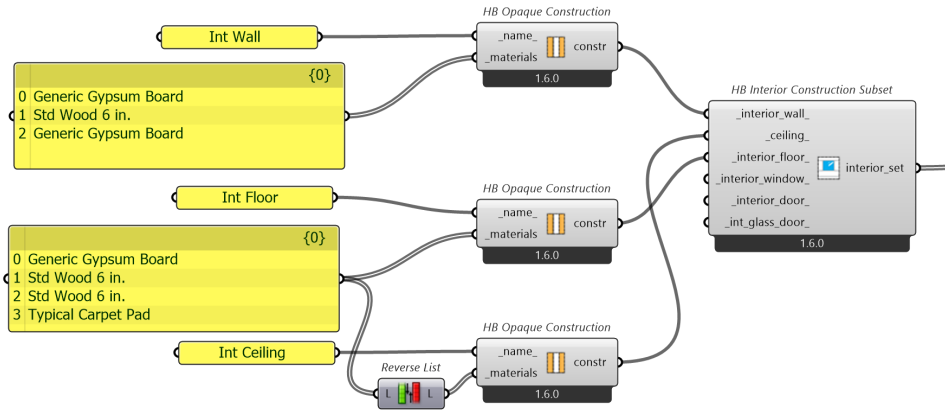


Figure 3.12.: Grasshopper - Interior elements constructions

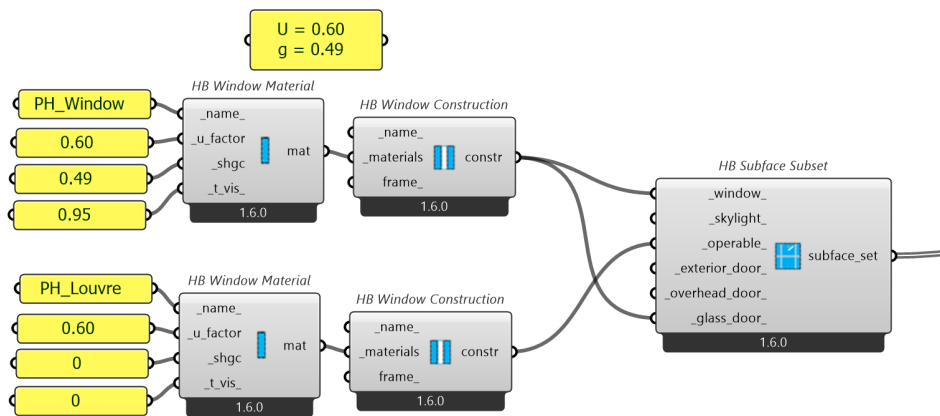


Figure 3.13.: Grasshopper - Window and louvre build-ups

### 3. Case Study: Welsh School

#### 3.3.4. Schedule and Equipment Usage

The programme defines the use schedules, as well as the demands for lighting, electrical equipment (unregulated loads), hot water, and ventilation requirements. In some of the programme definitions, the factors are converted from treated floor area (TFA) to gross area as Passivhaus uses the former and Honeybee the latter.

In addition to the day programme of teaching, the school also houses community functions in the hall and some of the nearby rooms. This means the use pattern is more unpredictable and for creating the digital twin and calibrating it against the on-site data some assumptions in use were made. Figure 3.14 shows the definition of the occupancy schedule for weekdays and weekends, as for well as summer and winter holidays. The first staff arrives at six and seven in the morning before the students arrive at eight o'clock. In the afternoon and evening, the occupancy decreases gradually as students go home but maintains some occupancy to account for after-school activities and community use. On the weekends the occupancy is assumed to be 10% of the peak occupancy to account for community use and other activities.

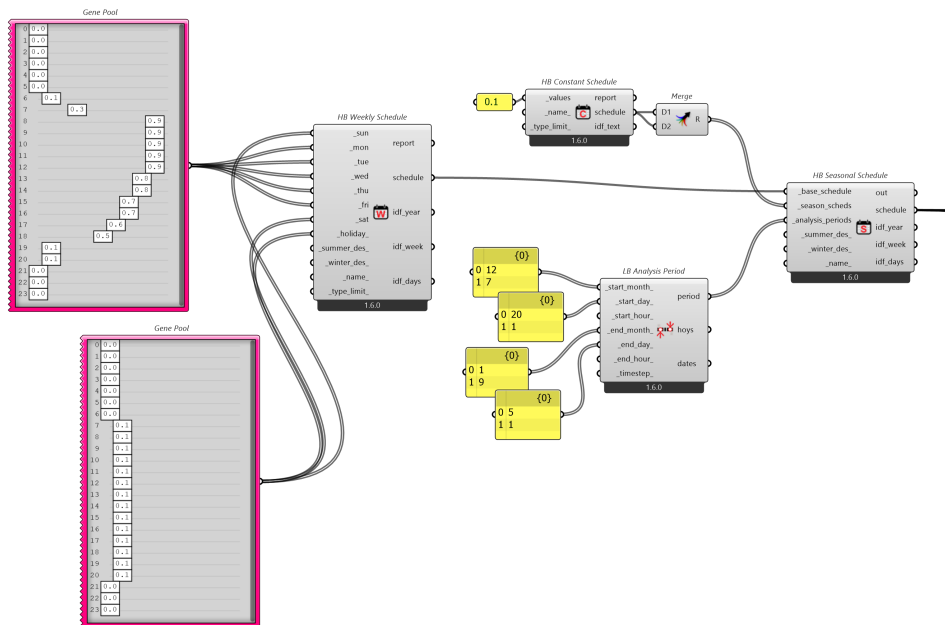


Figure 3.14.: Grasshopper - Use schedule setup

Figure 3.15 shows the internal and building loads defined, some of which have been simplified.

The occupancy of the school is about 450, most of whom are children aged 4-11. The standard assumption of the *HB People* component is 100 W/person. Children have a lower metabolic rate, however, they are generally also more active than sedentary adults. As a potential over-simplification a reduction factor of 0.5 was used to account for the lower heat production resulting in an assumption of 50 W/person.



### 3.3. Digital Twin Documentation

The lighting load is based on construction specifications that average  $5 \text{ W}/\text{m}^2$  for the entire building. The lighting use is assumed to be directly proportional to the occupancy, which is a simplification compared to more advanced approaches incorporating daylight availability to offset lighting use.

The *HB Equipment* component defines the unregulated electrical loads, which are difficult to account for in theory due to the unpredictability of actual use by the occupants. The values used are based on the Passivhaus specifications, which as standard, are checked by and signed off by the building owner to ensure correct modelling of the electrical loads. The data from the site monitoring is not adequately frequent nor segmented to differentiate between the various equipment loads.

The infiltration rate is based on the on-site measured performance of the building during and after construction using a blower door setup in a certified Passivhaus test.

The hot water demand is based on the Passivhaus factor of 2.3 litres per person per day. This only includes hot water production, not cold water use, and covers both tap water and kitchen use for meal preparation.

The ventilation rate in Passivhauses are based on the occupancy as well as minimum exhaust requirements. The ventilation loads defined in the energy model are based on the Passivhaus design specifications of 8 litres per person per second for the whole building including the kitchen and bathroom extracts. This equates to roughly  $13,000 \text{ m}^3/\text{h}$  during operation hours.

3. Case Study: Welsh School

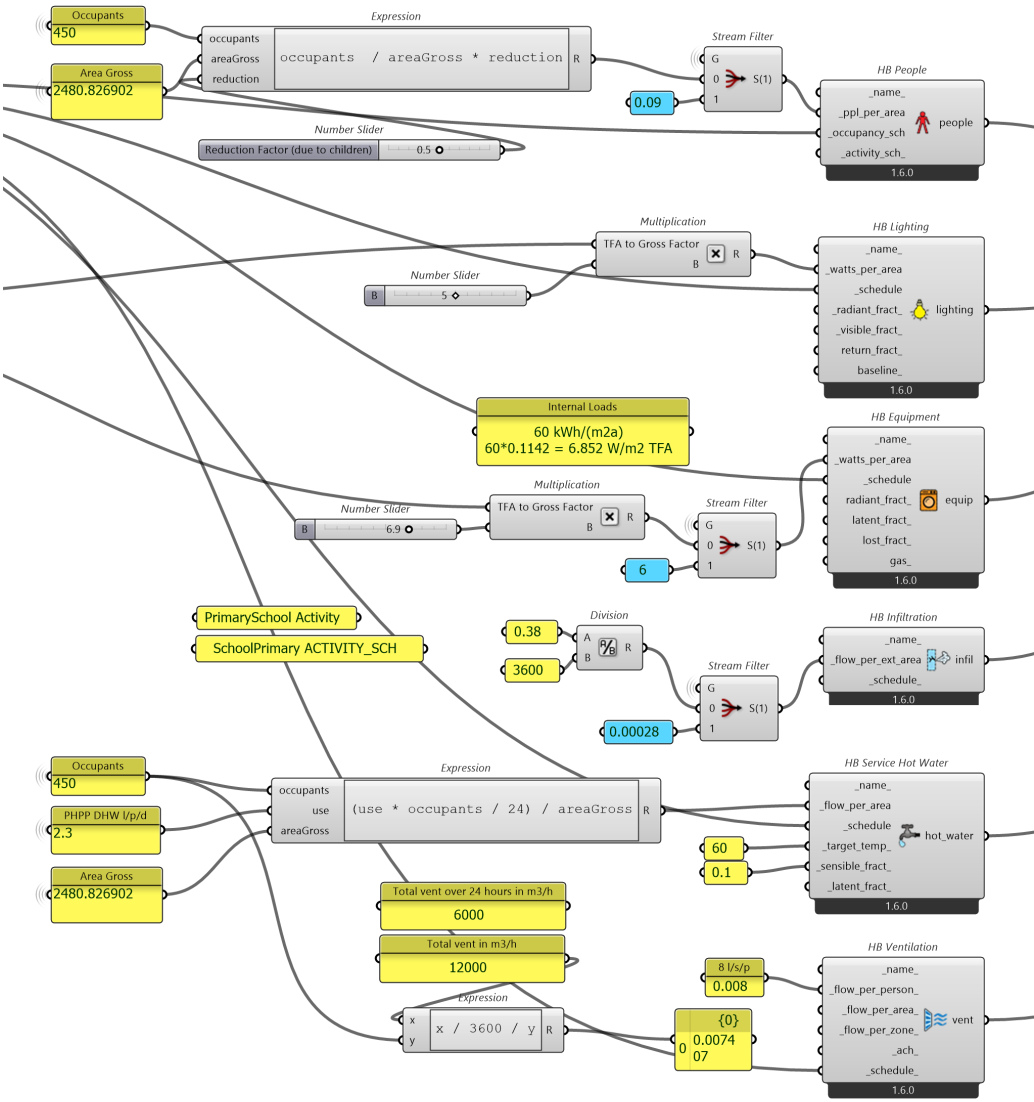


Figure 3.15.: Grasshopper - Internal and building loads

### 3.3.5. Hot Water and HVAC

The school has two hot water systems. One gas heater for the kitchen and space heating with a hot water storage tank with an immersion heater to cope with high demand during kitchen cleaning. The second system to supply all other sinks use small direct electric heaters with little to no storage in order to reduce circulation losses. As a simplification, these two systems were modelled as a single system in Honeybee.

The ventilation and fresh air are provided by mechanical ventilation with heat recovery for the whole school. The kitchen has a separate unit, because of the greasier air and different allowable temperature ranges in the kitchen. For the digital twin, this was simplified into one system, as the kitchen only represented a small part of the overall building and would require additional zoning in the modelling. The ventilation is modelled by the *HB IdealAir* component and is based on occupancy, which dictates the ventilation rate. In the simulation the *HB IdealAir* component also supplies heating through the air, which simplifies the energy model, but also implies that there is no secondary heat source such as the radiators. This limits the maximum heat supply load to what can be supplied through the supply air because the standard maximum supply air temperature is set as 50 °C in the component. This roughly matches the Passivhaus assumption of a maximum 52 °C supply air temperature to avoid the smell of burnt dust.

The heating for the school is provided through low-temperature hot water radiators in each classroom. However, in the digital twin, this is simplified to be heating provided through the supply air. The maximum heating load for the whole school during EnergyPlus simulations was 10 kW via the *HB IdealAir* component. The worst-case scenario heat load estimate by the Passivhaus PHPP tool is 15.9 kW or 8 W/m<sup>2</sup> of treated floor area. The installed radiator capacity in the school is 48 kW, however, this is not currently modelled in the digital twin and heating is assumed to be provided through the ventilation via the *HB IdealAir* component. This is a limitation to the complexity and accuracy of the digital twin, which does imply that the maximum heat load capacity is much lower than the real building since the radiators are ignored. This in turn limits the smart control system's ability to make quick corrections by turning on all the radiators at once.

The school has no active cooling system, but the ventilation louvres are designed to be safe from intruders and are therefore used for nighttime cooling in the warmer season. During the monitoring of the school, this nighttime cooling was found to have a significant effect on the daytime temperatures and could be seen in the temperature data as some teachers occasionally forgot to open the ventilation louvres.

3. Case Study: Welsh School

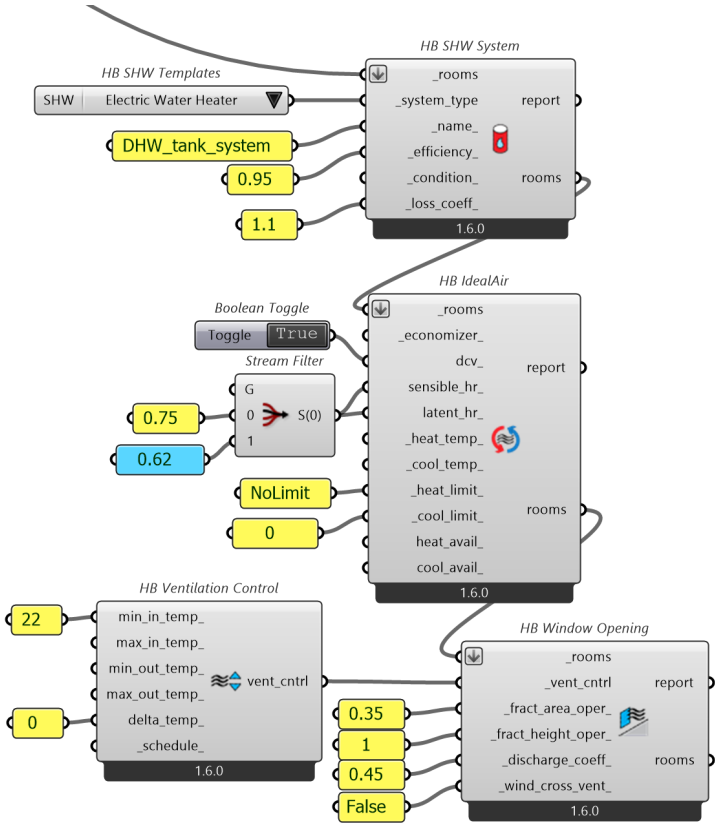


Figure 3.16.: Grasshopper - Hot water system, mechanical and natural ventilation

### 3.3.6. Natural Ventilation

Figure 3.17 shows a part of the south facade of the school illustrating the window and louvre strategy. Each room has fixed windows for daylighting as well as timber panels with louvres for fresh air, ventilation, and nighttime cooling. In the construction, the windows are placed near the centre line of the insulation, to minimize thermal bridging around the frame installation. This also provides 230 mm deep reveal shading for the windows. In addition, the south facade has permanent overhang shading, preventing overheating in the summer.

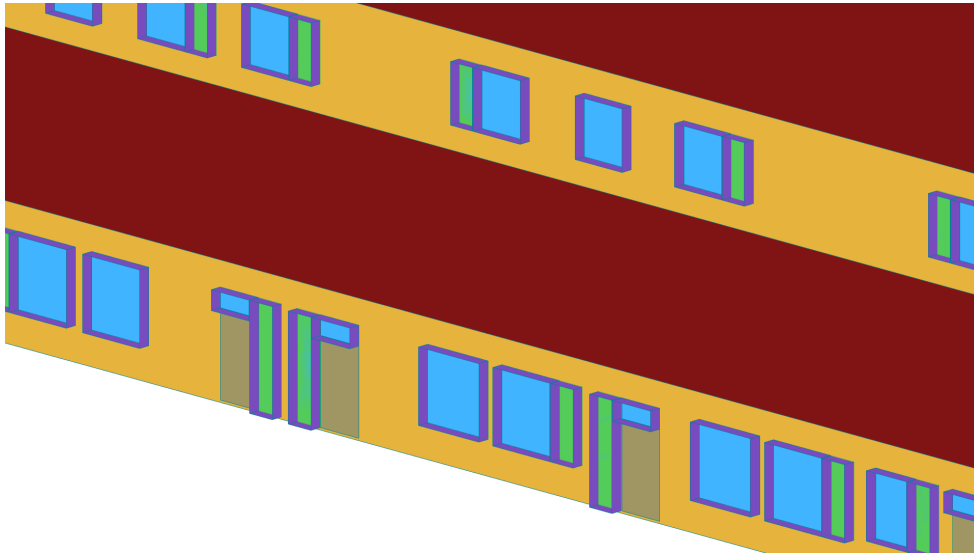


Figure 3.17.: Grasshopper - Close-up of windows and vertical louvres in green. Shading overhangs not visible in view.

### 3.3.7. Custom Parameters

In order to use the energy management system (EMS) to control the EnergyPlus simulation during runtime, it first had to be made visible in the IDF file and secondly, actuator objects had to be created for the louvres. Figure 3.21 shows the *Output:EnergyManagementSystem* object added to the IDF. This also enables the creation of the .edd dictionary file, containing all available actuators.

Figure 3.18 shows the *HB Set Identifier* component, which ensures that the specified naming in Honeybee is transferred verbatim to EnergyPlus. This is disabled by default as a safety feature to avoid naming conflicts or invalid names in EnergyPlus. Using this also means the actuated louvres need to be named uniquely, which is done by numbering. This is important when calling an EnergyPlus object via Python during the simulation. Otherwise, the correct objects cannot be identified.

Figure 3.19 shows how the EMS actuator objects for each opening louvre are created in Grasshopper. Each unique louvre must have its own actuator object in the IDF written in EnergyPlus Runtime Language (ERL). Figure 3.20 shows a handful of the louvre actuator objects, which are then added to the IDF file.

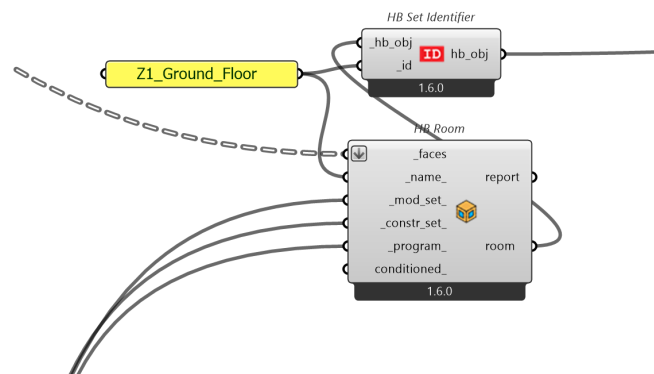


Figure 3.18.: Grasshopper - *HB Set Identifier*, for forcing consistent naming in IDF file

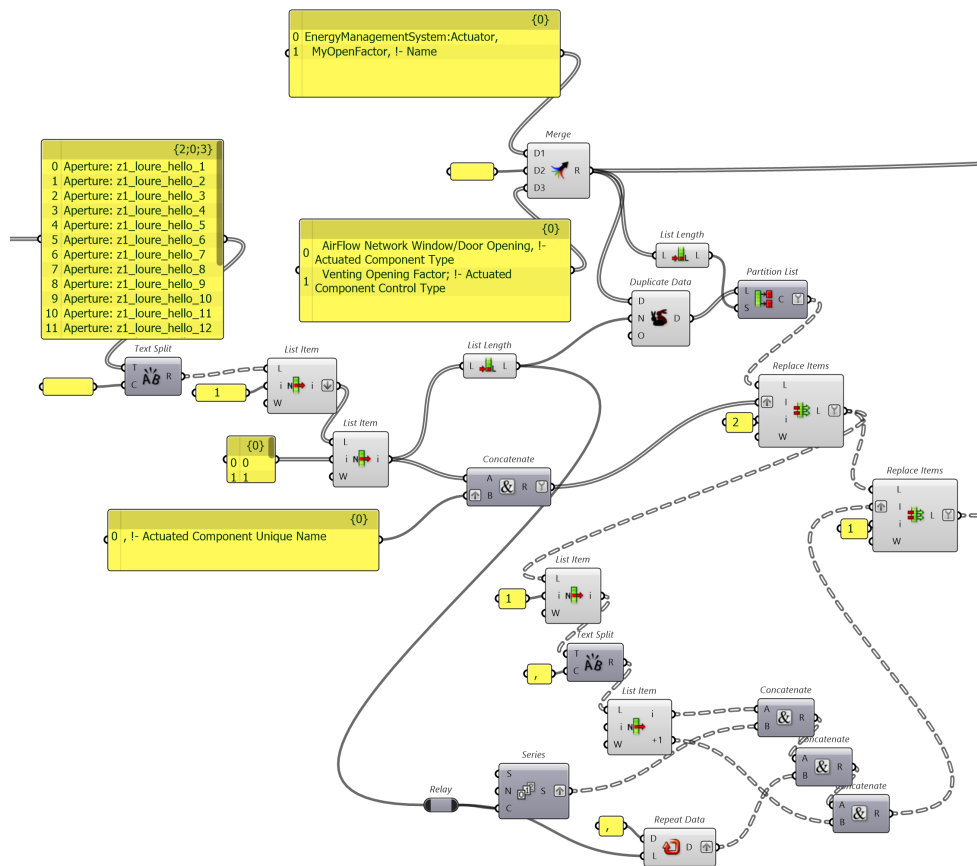


Figure 3.19.: Grasshopper - Creation of EMS Actuator objects for EnergyPlus in EnergyPlus Runtime Language (ERL)

```

{0}
0 EnergyManagementSystem:Actuator,
1 MyOpenFactor1, I- Name
2 z1_loure_hello_1, I- Actuated Component Unique Name
3 AirFlow Network Window/Door Opening, I- Actuated Component Type
4 Venting Opening Factor; I- Actuated Component Control Type

{1}
0 EnergyManagementSystem:Actuator,
1 MyOpenFactor2, I- Name
2 z1_loure_hello_2, I- Actuated Component Unique Name
3 AirFlow Network Window/Door Opening, I- Actuated Component Type
4 Venting Opening Factor; I- Actuated Component Control Type

{2}
0 EnergyManagementSystem:Actuator,
1 MyOpenFactor3, I- Name
2 z1_loure_hello_3, I- Actuated Component Unique Name
3 AirFlow Network Window/Door Opening, I- Actuated Component Type
4 Venting Opening Factor; I- Actuated Component Control Type

{3}
0 EnergyManagementSystem:Actuator,
1 MyOpenFactor4, I- Name
2 z1_loure_hello_4, I- Actuated Component Unique Name
3 AirFlow Network Window/Door Opening, I- Actuated Component Type
4 Venting Opening Factor; I- Actuated Component Control Type

{4}
0 EnergyManagementSystem:Actuator,

```

Figure 3.20.: Grasshopper - Actuators in EnergyPlus Runtime Language (ERL) format

3. Case Study: Welsh School

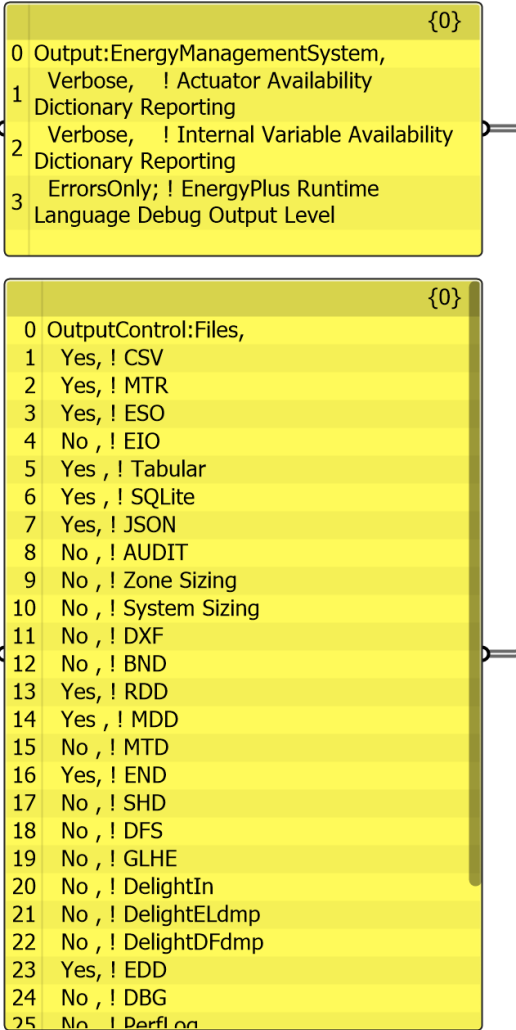


Figure 3.21.: Grasshopper - EMS and output files added to simulation parameters output



## 4. Calibration and Validation

The digital twin energy model is calibrated to match data collected from the school in use. This is to ensure the digital twin is accurate enough before it can be used for further experiments.

The following sections briefly cover the on-site data, calibration in accordance with [ASHRAE](#) standards and finally the results of the accuracy of the calibration.

### 4.1. On-site data

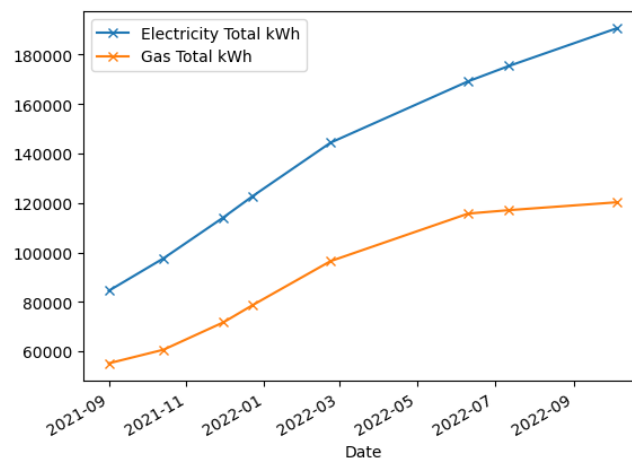


Figure 4.1.: Cumulative energy usage measured from the building.

The school was monitored initially for one year to give an overview of its performance as part of a post-occupancy evaluation by Architype; the architects and the Passivhaus designers of the project. Some measuring equipment was deployed for longer to collect more data.

Figure 4.1 shows the electricity and gas meter readings from the school, taken at 1-3 month intervals. As outlined in Chapter 3 gas is used for kitchen hot water as well as space heating. There is a 2-3 week winter holiday and a summer holiday of about 2 months. The electricity and gas use over the summer does indicate that the building is either in use for non-school functions or the equipment is not turned off, such as freezers, fridges and hot water tanks.

In addition to meter readings, a handful of classrooms were equipped with temperature, CO<sub>2</sub>, relative humidity, pressure, volatile organic compounds, light, and pressure data loggers. In addition, a few windows had window-opening monitors and some occupancy

#### 4. Calibration and Validation

sensors were also trialled. For validation, the most relevant and reliable data was the temperature data taken in four different classrooms. CO<sub>2</sub> and relative humidity could also be simulated and used for validation, but this was not pursued.

##### 4.1.1. Temperature Data



Figure 4.2.: Temperature data for calibration is from classrooms G36, G13, F04, and F16.

Figure 4.2 shows the four classrooms from which temperature data is used for the calibration. Two classrooms per zone of the energy model. The data was recorded at 10-minute intervals, but converted to hourly data for the calibration to match the later work using hourly time-steps. However, a shorter interval could also have been used for the calibration. The temperature values of each of the two classrooms are averaged for each zone, to get closer to a real mean temperature for each zone. The ground-floor classrooms could also be analysed individually to see the difference between north and south-facing classrooms compared to the simulated zone mean temperature, however, this was not pursued.

##### 4.1.2. Calibration Equations

Calibration was done according to the [ASHRAE \(2002\)](#) guidelines as cited in the calibration and validation document created by Regina Bokel [2019](#). To validate the calibration of the model Normalized Mean Bias Error (NMBE) and Coefficient of Variation of the Root Mean Square Error (CV(RMSE)) error measures were used. For hourly data the NMBE must be below 10% and the CV(RMSE) below 30%.

Equation 4.1 shows the NMBE equation. The output is a percentage error which should be as low as possible.

$$NMBE = \frac{1}{\bar{m}} \frac{\sum_{i=1}^n (m_i - s_i)}{n} \times 100\% \quad (4.1)$$

$\bar{m}$  is the mean of all the hourly measured temperature samples.

$m_i$  is the mean of temperature samples from the two classrooms for time  $i$ .

$s_i$  is the temperature outputted by the model at time  $i$ .

$n$  is the number of samples.

Equation 4.2 shows the CV(RMSE) equation. The output is a percentage error which should be as low as possible. The criteria for each measure can be seen in table 4.1.

$$CV(RMSE) = \frac{1}{\bar{m}} \sqrt{\frac{\sum_{i=1}^n (m_i - s_i)^2}{n}} \times 100\% \quad (4.2)$$

### 4.1.3. Digital Twin Validation

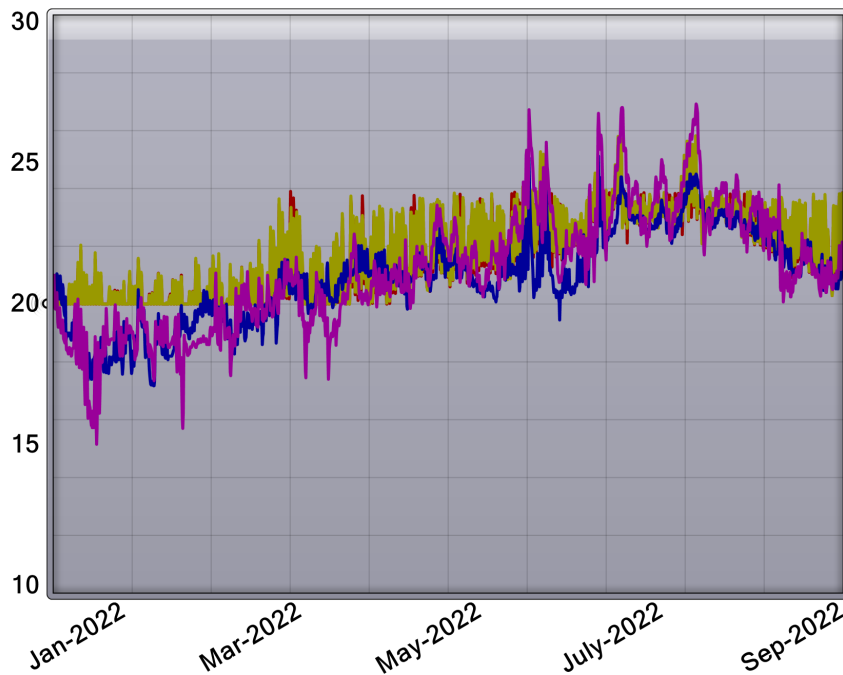


Figure 4.3.: Temperature data measured from site and simulated by energy model. Blue: Ground Floor measured data. Purple: First floor measured data. Red and Yellow: Simulated data for ground and first floor zones.

#### 4. Calibration and Validation

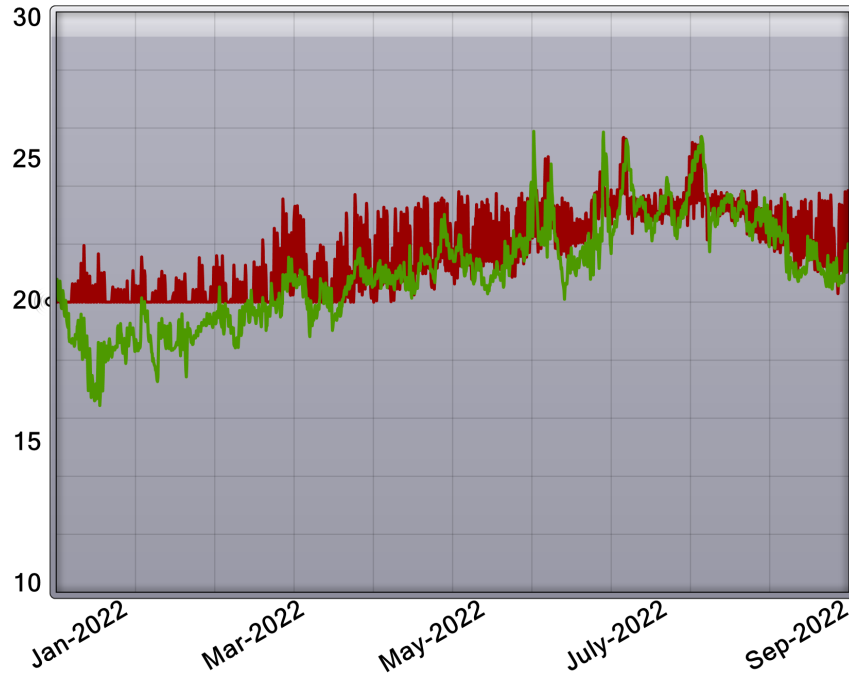


Figure 4.4.: Averaged temperature data measured from site and simulated by energy model. Green: Measured data. Red: Simulated data.

Figures 4.3 and 4.4 shows the temperature data that was measured and simulated for the period of January 2022 to September 2022. This was done using local historical weather data for that period in the EnergyPlus simulation.

Figure 4.3 shows that the first floor in purple has a higher variation than the ground floor in blue. This could be due to their higher relative heat loss area, as the first-floor classrooms have both a wall and a roof. In addition, both of the first-floor classrooms are on the north facade. This could also explain the summer peaks due to the early morning and late evening summer sun hitting the un-shaded north facade. From the on-site monitoring, it was also found that the teachers do not open the windows at night for cooling at the same rate, leading to different classroom temperatures. It is not known which specific classrooms or periods this night ventilation behaviour affects, but it could be part of an explanation for the variation seen in the figures.

Figure 4.4 indicates that the simulation in red has a much higher daily variance, especially in the shoulder season from March to June.

ASHRAE Standard	Hourly Calibration Criterion	Results (varying periods)
NMBE	<10%	-8% to 10.45%
CV(RMSE)	<30%	2% to 11.32%

Table 4.1.: Table showing the ASHRAE criteria and results of the simulation calibration for varying periods of week by week up to the full 9-month period.

## Results

Using the [NMBE](#) and [CV\(RMSE\)](#) measures to validate the simulation, the error stays well within the 10% and 30% limits respectively for hourly data. This was checked for every week, every month as well as the whole period.

The worst result was for the first week of January for which the [NMBE](#) was 10.45%, which is just above the 10% limit according to the guidelines above. However, the [CV\(RMSE\)](#) for the same week was only 11.32%, well within the 30% limit. This is likely due to low temperatures over the Christmas holiday as well as a late start in January, whereas the simulation maintained a higher stable temperature around 20 °C.

### 4.1.4. Conclusion

The validation above was achieved using the energy model developed in Chapter 3 based solely on the construction drawings and Passivhaus design information. There was no subsequent tweaking of parameters in order to bring the [ASHRAE](#) calibration scores closer to 0%. This is a potential area of improvement to get a more accurately calibrated model, but this was not pursued further. This digital twin will now be as a test bed for the Reinforcement Learning and energy-saving experiments.

Note: Prior to the validation done in this chapter based on temperature measurements according to the [ASHRAE](#) guidelines, there was an attempt to calibrate the model to more closely match the *meter data*, instead of *temperature data*, using a genetic algorithm and random sampling approach. This is outlined in the Appendix B. The parameters used in the genetic algorithm calibration experiment were; people per area, watts per area, infiltration rate, windows u-value and g-value, walls thermal resistance, thermals mass of interiors and heat recovery efficiency.



## 5. Reinforcement Learning Algorithms and Experiments

This chapter dives into how the RL agents were designed, coded, and tested together with the 8 building variations of the digital twin in order to answer the research question.

The goal of all the agents is to save energy while maintaining comfort. A simple yet difficult task.

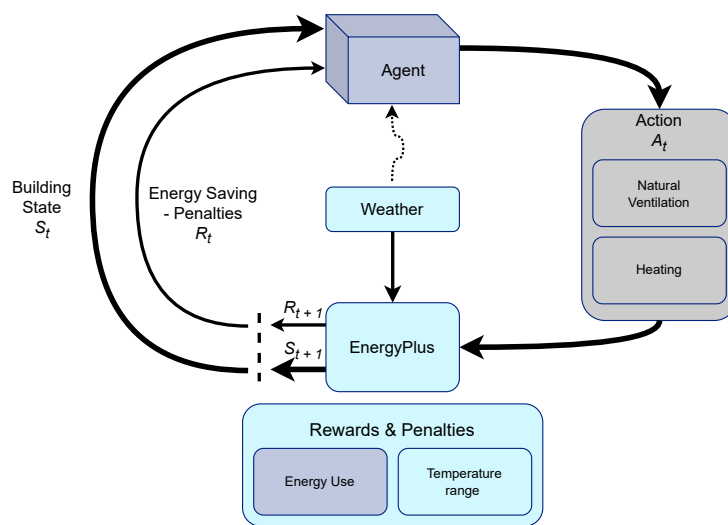


Figure 5.1.: Diagram of agent and environment state, action and reward cycle.

### 5.1. Programming Setup

To create the DRL agents in Python, the library PyTorch (Paszke et al., 2019) was used, as it contains the basic buildings blocks for many AI applications. In addition, the EnergyPlus API, accessed using Python, is used to generate the environment in which the agents operate. The library Eppy was also used for IDF file manipulations from within Python. Furthermore, common libraries such as Pandas, Numpy, os, PyPlot etc. were also used.

The EnergyPlus Python API documentation is sparse and lacks example files for reference. However, there are a few examples from companies and single researchers using the API, who have shared their code and some basic instructions on Github. This thesis uses the framework created by Eubel (2023). Using this still requires familiarity with the .edd and .rdd files as well as the naming conventions, which are not consistent between EnergyPlus

and the work by Eubel. In addition, the framework only provided limited access to the full EnergyPlus API data and some modifications were made by the author to the underlying code and libraries.

## 5.2. Reinforcement Learning

The main method used is DQN which approximates the state-value function and chooses the action with the highest expected reward. This uses a simple fully connected NN. RNNs were also used for the time-series data from the weather forecast. Combining these gives a Deep Recurrent Q-Learning (DRQN) model, which has not been found to be used in previous studies.

### 5.2.1. Agent and Environment Interaction

The agent is the DRQN and the environment is the digital twin in EnergyPlus. Figure 5.2 is the specific case of the diagram in figure 5.1 where the timeline is shown at the bottom. At each time step, which is every hour, EnergyPlus waits and exchanges information with the agent before it continues the simulation to the next time step or hour. This exchange happens at every time step. This would be a similar setup for simpler control algorithms too, the agent would just be exchanged for a simple control algorithm.

Figure 5.3 shows an exchange in more detail. The agent requests information from the EnergyPlus API about the current state and future weather. This information is used to calculate the reward for the previous action, which the agent then uses to update it Deep Recurrent Q-Learning to better predict the next action with the highest reward. In addition, the agent checks the current time and date of the time-step and if it is Sunday at 23:00 it activates *experience replay*, which trains the agent again on previous experiences stored in its memory. After this, using the state information it then chooses the next action with the highest predicted future reward and returns this action to the EnergyPlus API, which allows EnergyPlus to progress to the next time-step. The internal workings of the agent are illustrated in Figure 5.4.

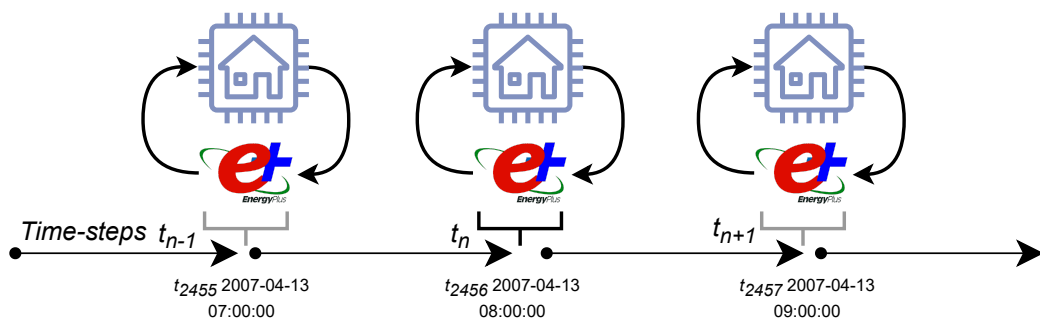


Figure 5.2.: Diagram of multiple time-steps in the simulation and agent interaction. At each hourly time-step the agent interacts with the EnergyPlus API.



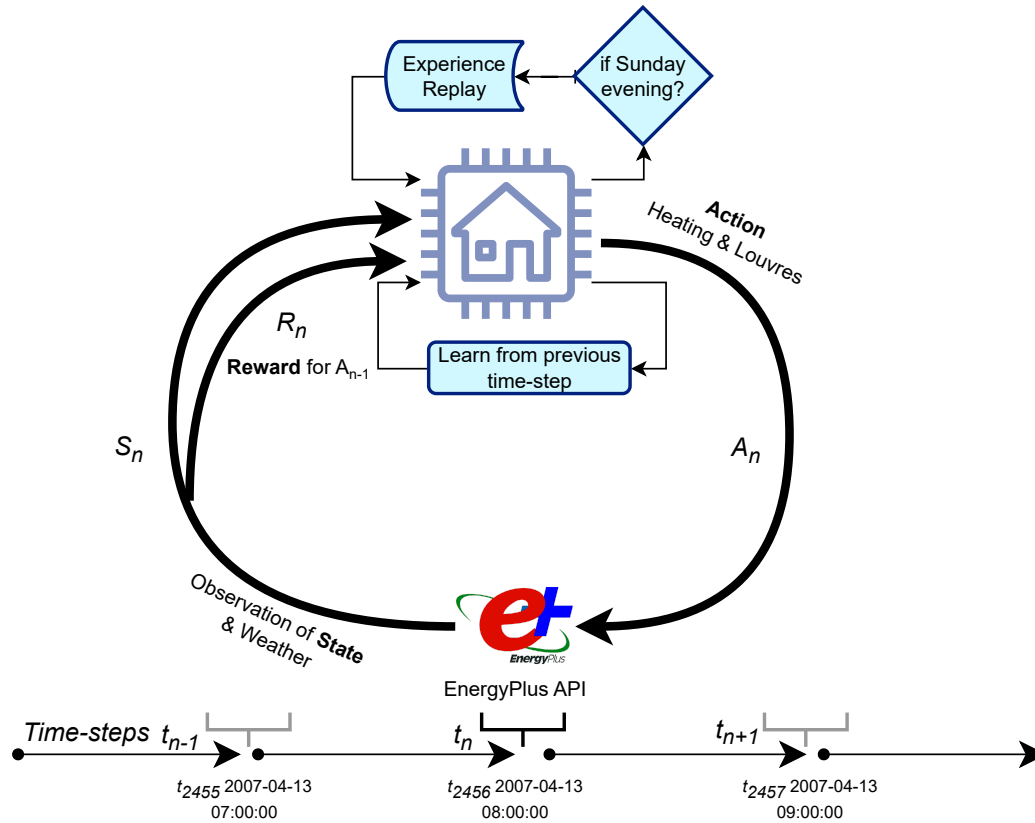


Figure 5.3.: Diagram of agent and EnergyPlus interaction in detail.

### 5.2.2. RL Agents

Figure 5.4 shows the AI network architecture of the largest agent. All other agents are variations of this network.

The recurrent networks sit before the fully connected network and thereby transform the time-series data before it is fed into the NN. There are two RNNs, one for the future open hours boolean and one for the future weather data. The open hours are a boolean describing whether the school is open or closed, which determines whether the comfort criterion is in effect. The rest of the information defining the current state is in the *current readings* which are fed directly into the NN. The future opening hours are a 72-digit long sequence, representing the next 72 hours. The RNN has a size of 32, meaning there are 32 parallel NN nodes in each RNN layer. This also makes the RNN output a 32-dimension vector. The weather information is 24 hours, 4 hours or 0 hours depending on the agent and sometimes includes solar radiation information, but the RNN size is always 32. This does mean the 24-hour agent with full information has more information per node than the agents with less information.

The fully connected network model has 2 hidden layers of 300 nodes each and the output layer has 11 nodes, resulting in 11 possible actions. These actions are combinations of heating set-points and ventilation louvre openings.



-1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. -1. 0.148 0.148 0.036 0.052 0.108 0.104 0.04 0.076  
0.028 -0.028 -0.092 -0.152 -0.104 -0.108 -0.108 -0.08 -0.072 -0.04 -0.028 -0.004 0.048 0.132 0.112  
0.12 -0.654 -0.866 -0.94 -0.914 -0.912 -0.962 -0.94 -0.944 -0.934 -0.332 -0.568 -1. -1. -1. -1. -1. -1.  
-1. -1. -1. -1. -1. -1. -0.98 -0.346 -0.496 -0.328 -0.27 -0.236 -0.208 -0.29 -0.39 -0.534 -0.676 -0.788  
-0.958 -1. -1. -1. -1. -1. -1. -1. -1. -1. -0.982 -0.886 -0.742 0.35902 0.36312 -1. 0. 0.12 0.12 0.68  
0.12 -0.0609 1. 0. -0.11111 -0.58667 0.19616 0.20669 0.35549 0.35969 0.36254 0.36655 0.00606  
0.0185 0.00594 0.01441 0.00107 0.00559 1.

---

### Current Readings

The *current readings* describe the current state of the building. In early developments this was limited to data that would be easily accessible from real-life monitors in a building, but this was expanded to include additional information from the digital twin such as the thermal energy heat transfers of the internal elements. This was done to give the agents richer information about the current state to increase their chances of making optimal decisions. This does not reflect information that would be easily accessible in a real scenario of a deployed agent in a building, but this is not the main objective of this thesis either. This is a trade-off in trying to give the agents more information to make them learn faster. Previous works have looked at such scenarios with poor information about the current state ([Gao et al., 2020](#)) and [Liu and Henze \(2006\)](#) tested agents pre-trained in a virtual environment before being deployed in a real building. This thesis is looking at the relationship between building factors and RL types and horizons instead.

List of **current readings**:

Heating energy use

Louvres open, boolean

Heating set-point zone 1

Heating set-point zone 2

Outdoor relative humidity

Outdoor dry-bulb temperature

Outdoor air pressure

Sun up, boolean

Raining, boolean

Wind direction

Wind speed

Zone 1 operative temperature

Zone 2 operative temperature

Zone 1 relative humidity

Zone 2 relative humidity

Zone 1 air temperature

## 5. Reinforcement Learning Algorithms and Experiments

Zone 2 air temperature

Zone 1 radiant temperature

Zone 2 radiant temperature

Zone 1 internal walls heat transfer rate

Zone 1 internal walls heat energy stored

Zone 1 concrete stairs heat transfer rate

Zone 1 concrete stairs heat energy stored

Zone 2 internal walls heat transfer rate

Zone 2 internal walls heat energy stored

Day of week

### 5.2.3. Hyperparameters

In AI and especially in DRL there are many hyperparameters to control and tweak. These describe the *settings* of the algorithms and can have a large influence on the algorithm's performance. For this thesis, some were more deeply investigated than others and are outlined in this section.

#### Actions

All agents have the same 11 actions to choose from. In previous iterations, the agents had 41 actions, but this was reduced in an effort to speed up training times.

One drawback of Q-learning is that only one action can be taken, which becomes a problem when there are concurrent actions such as choosing the heating set-point and choosing whether or not to open the ventilation louvres. That means each action must be a combination of those sub-actions. For this case, it was computationally manageable, but as suggested in the *Discussion*, chapter 7, other strategies should be investigated for more complex scenarios of multiple sub-actions.

The 11 actions, which are combinations of the 2 sub-actions are as follows:

18.0 °C - ventilation louvres **closed**

18.5 °C - ventilation louvres **closed**

19.0 °C - ventilation louvres **closed**

19.5 °C - ventilation louvres **closed**

20.0 °C - ventilation louvres **closed**

18.0 °C - ventilation louvres **open**

18.5 °C - ventilation louvres **open**

19.0 °C - ventilation louvres **open**

19.5 °C - ventilation louvres **open**

20.0 °C - ventilation louvres **open**

5.0 °C - ventilation louvres **closed**

An obvious drawback of using 11 rather than 41 actions is that the temperature increment is 0.5 °C, rather than 0.1 °C, which gives much less fine-grained control to the agent. The advantage is faster learning as there are fewer actions to test and therefore less complexity for the neural network. 41 actions were used in early iterations but were scaled back to 11 in an effort to speed up learning.

Unlike other buildings mentioned in the literature review, chapter 2, this school does not have movable shading, but only fixed overhangs. In addition, the windows are non-openable, instead, there are solid panels with louvres behind. These solid panels can be opened for natural ventilation. This means there are very few control surfaces and additional systems, compared to examples from the literature review some of which also used active heat storage and other electronically controlled features. This makes this a very difficult challenge for these agents as their best strategy may be to simply predict the future solar and air thermal energy availability and reduce the heating preemptively in an effort to minimize unnecessary heating.

### Rewards

The reward describes either the goals or the desired path towards the goal. Setting the right reward function is a challenge in itself, especially when there are competing goals such as energy saving and comfort. The reward is a 1-dimensional value, meaning that a weighting factor has to be used as the final reward can only be a single number, regardless of how many factors make up the reward function.

For these agents energy use and temperature are the only parameters considered for the reward. The goal is to reduce energy use while maintaining the temperature within a comfortable range of 20-25 °C. The rewards can be positive or negative penalties, the agent will always go for the highest reward, even if it is choosing between negative numbers only.

Figure 5.6 shows the penalty that the agent receives every hour depending on the temperature. The penalty is proportional to how far outside the desired temperature range the agent is. The downside of this is the dead zone in the middle, where the agent receives no penalty nor reward, meaning it has no indication of its performance at that point. One way to change that would be to add a positive proportional reward for being within the 20-25 range. However, if the peak reward is then at 22.5 °C, then the agent might prioritise a 22.5 °C building temperature over saving energy. Using a parabolic function rather than a piecewise one would have a similar effect of indicating that 22.5 °C is better than 20 °C. In reality, this may not be a problem for the agent as energy use is still penalised. This was not tested specifically but is listed as a potential improvement in the *Discussion* chapter 7.

The agent also receives a penalty for using energy to heat the building. Specifically, a penalty of 0.6 per hour per 10 kWh is used. The penalty for temperature is much higher. For instance, if the building is at 19 °C the penalty per hour is 10, and 1 if it is at 19.9 °C. 10 kWh for heating is roughly the maximum the building would be using on the coldest day, therefore, comfort should have a relatively high priority. In previous iterations, the energy use had a higher weighting, but comfort was compromised too frequently.

## 5. Reinforcement Learning Algorithms and Experiments

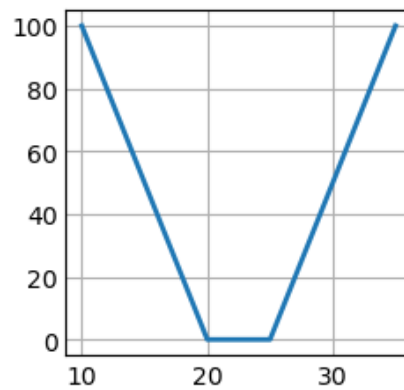


Figure 5.6.: Graph of RL agent's penalty function for internal temperatures

The comfort or temperature penalty is only active during school hours between 8 and 15 on weekdays, also referred to as the opening hours in the context of the agent input data. At all other times, only the energy use contributes to the reward signal. This is a simplification and limitation as the community use of the building outside school hours is unknown.

### Discount rate

The gamma or discount rate was set to 0.9 and describes the weighting of future predicted rewards when estimating the value of an action. A higher gamma roughly means greater priority to future predicted rewards.

### Learning rate

The learning rate was 0.0001 and describes the magnitude of the change in weights during backpropagation.

### Epsilon

Epsilon is the probability of choosing a random action. This is done to encourage and maintain exploration of the environment. At the beginning of training the epsilon is 50-70% but tapers down to a steady 5% for the remaining of the training run. The epsilon is excluded or 0 when testing the final performance of the model - however, in a case where the agent is deployed in a real environment that changes over time epsilon should be greater than 0 to encourage the agent to continue learning from otherwise unseen actions.

### Experience replay

During training the agent learns from every step or action, but these steps can also be stored for later retrieval and training. This is called experience replay and both speeds up training while maintaining old or rare memories to make the model more robust to outliers. The

experience replay for the agent was done once per simulated week on 1,000 hourly memory samples. This was done throughout the whole training period using a *deque* (double-ended-queue) in Python with a set max length of 100,000 equating to roughly 11 years of hourly memories being stored.

#### 5.2.4. Training Processes and Developments

The training period used weather data from the nearby region in Wales to the actual building site. This was done as each weather station only has a limited time range of data available and it is necessary to subject the agent to a wide range of weather instead of using the same 1-year data repeatedly. The weather data was accessed from [OneBuilding \(2023\)](#). After training, the testing of the agents was done using weather data from the weather station nearest the building site.

Throughout the agent design and development, the agents were trained on a 2018 laptop with an Intel i7 12-core processor. Due to the EnergyPlus simulation being the bottleneck, the GPU had little influence on the performance, though larger networks did use more compute to train.

The final batch of agents was trained on a server boasting a 40-core CPU. However, EnergyPlus only uses a single core when calculating solar angles and only a handful the rest of the time, resulting in a low overall utilisation of resources. Each agent was trained for 35 simulated years - a multiple of 7 to avoid some of the quirks of the EnergyPlus calendar. Each run to train an agent took 8 hours on the server.

Improvements are suggested in the *Discussion*, chapter 7.

### 5.3. Building Digital Twin Variations

As part of investigating the influence of the building fabric factors compared to DRL control algorithms, 8 new building variants were created. These are variations of the original building, changing the wall and roof insulation and changing the internal thermal mass. The 9 buildings are illustrated in figure 5.7.

The insulation and thermal mass are halved and doubled in combination for each variant. For the insulation, the total R-values of the roof and wall constructions go from 4.5 to 9 to 18  $m^2K/W$ . The internal mass is relative to the current thermal mass of the internal walls and concrete stairs of the original building. This is then halved or doubled in area for each variant.

One simplification from the original digital twin is that for training and testing, all holidays and weekend activities are removed, resulting in a simpler schedule that repeats every week. This was done to simplify the problem and speed up the learning of the agents. The majority of heating takes place during winter, when there are only 3 weeks of holiday, compared to the long summer holiday. In reality, there can be a different static set-point during holidays, meaning the RL agent would not be active during those times. Alternatively, a more advanced agent can be created to take into account the additional complexity of holiday periods.

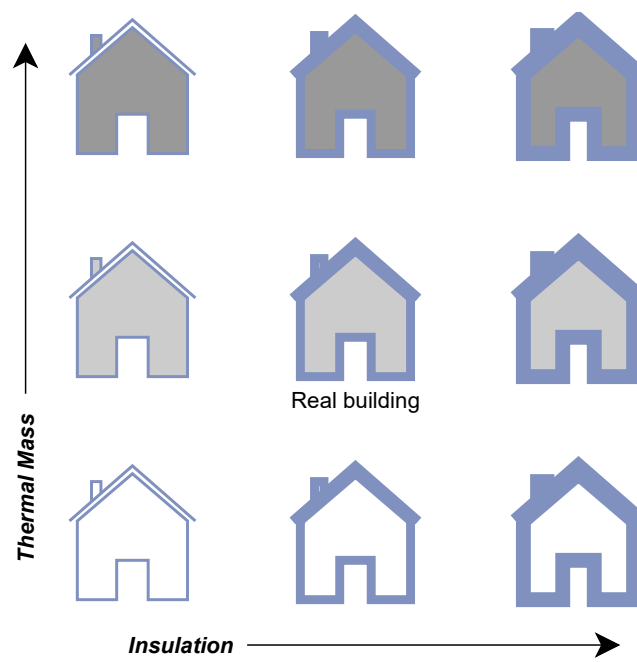


Figure 5.7.: Illustration of the 9 types of buildings created as variations from the original digital twin model.



## 5.4. PyTorch Model Code

The DRL model architecture matches that of figure 5.4 and adjusts to each agent depending on its future foresight length, whether solar is included, and the number of RNNs. Using PyTorch can make it appear deceptively easy to construct a neural network, which it can be with some practice. However, the real challenge is in the `forward(self, x)`, line 29, function when splitting the incoming tensor of unknown batch size into each vector for the different RNNs and fully connected NN.

The model `__init__` function, line 8, initialises the model and takes in hyperparameters from the agent. Namely the number of flat non-time-series inputs (the current readings), the size of hidden layer 1, the size of the hidden layer 2, the output size (the number of actions), the foresight length (future weather data horizon), and whether or not solar weather data is included.

Note: For the full Python code see the author's [Github](#).

```

1 import torch
2 import torch.nn as nn
3 import torch.optim as optim
4 import torch.nn.functional as F
5 import os
6
7 class Linear_QNet(nn.Module):
8     def __init__(self, input_size_flats, hidden_size1, hidden_size2, output_size
9         , fsight_len, solar_included=True):
10         super().__init__()
11
12         # RNNs into linears
13         # input = rnn + rnn + raw from forward x linear 1
14         self.rnn_size = 32
15         self.fsight_len = fsight_len
16         self.solar_included = solar_included
17         self.flats_len = input_size_flats
18
19         self.rnn_work = nn.RNN(1, self.rnn_size, num_layers=1, batch_first=True)
20         if self.solar_included == True:
21             self.rnn = nn.RNN(3, self.rnn_size, num_layers=1, batch_first=True)
22         else:
23             self.rnn = nn.RNN(1, self.rnn_size, num_layers=1, batch_first=True)
24
25         self.linear1 = nn.Linear(self.flats_len + 2*self.rnn_size, hidden_size1)
26         self.linear2 = nn.Linear(hidden_size1, hidden_size2)
27         self.linear3 = nn.Linear(hidden_size2, output_size)
28
29     def forward(self, x): # This is called when calling self.model(state [,
30         state[1] optional])
31
32         if len(x.size()) != 2:
33             x = torch.unsqueeze(x, 0)
34             assert len(x.size()) == 2, 'Forward definition, the unsqueeze to
35                 standardize tensor failed'
36
37         if self.solar_included == True: # work hours, temps, rad glo, rad dif,
38             flats
39             xs = x.split_with_sizes([72, self.fsight_len, self.fsight_len, self.
40                 fsight_len, self.flats_len], 1)
41             xwork, xtemps, xradglo, xraddif, xflats = xs[0], xs[1], xs[2], xs
42                 [3], xs[4]

```

## 5. Reinforcement Learning Algorithms and Experiments

```
38
39     xseq = torch.stack((xtemps, xradglo, xraddif), dim=-1)
40     else: # work hours, temps, flats
41         xs = x.split_with_sizes([72, self.fsight_len, self.flats_len], 1)
42         xwork, xtemps, xflats = xs[0], xs[1], xs[2]
43
44         xseq = xtemps.unsqueeze(-1)
45
46
47         xwork = xwork.view(-1, 72, 1)
48         outwork, hxwork = self.rnn_work(xwork)
49         outwork = outwork[:, -1]
50
51         outseq, hseq = self.rnn(xseq)
52         outseq = outseq[:, -1]
53
54         # combine outputs back into 1 tensor with batch
55         x = torch.concat((outwork, outseq, xflats), 1)
56
57         x = F.relu(self.linear1(x))
58         x = F.relu(self.linear2(x))
59         x = self.linear3(x) # this was used in tutorial where raw numbers are
60         outputted
61     return x # x is a vector containing the values of the output layer
```

## 6. Climate AI Results

This chapter presents the data, interpretations, and findings from the experiments using the 6 different DRQN agents in combinations with the 9 different building variations to answer the research question.

### 6.1. Experiments Completed

Building type name	Insulation	Thermal Mass	Baseline EnergyPlus Heating [kWh/m2/a]	RL 24h Foresight RNN Heating [kWh/m2/a]	RL 4h Foresight RNN Heating [kWh/m2/a]	RL No Foresight Heating [kWh/m2/a]	RL 24h Foresight RNN w/o Solar Heating [kWh/m2/a]	RL 4h Foresight RNN w/o Solar Heating [kWh/m2/a]	RL 4h Foresight flat inputs Heating [kWh/m2/a]
Building-InsuBASE-MassBASE	Baseline	Baseline	8.20	5.83	6.36	8.20	6.55	7.35	8.08
Building-InsuBASE-MassDW	Baseline	Decreased 50%	8.40	5.68					
Building-InsuBASE-MassUP	Baseline	Increased 2x	7.89	5.07					
Building-InsuDW-MassBASE	Decreased 50%	Baseline	25.02	21.02					
Building-InsuDW-MassDW	Decreased 50%	Decreased 50%	25.29	24.04					
Building-InsuDW-MassUP	Decreased 50%	Increased 2x	24.63	21.67					
Building-InsuUP-MassBASE	Increased 2x	Baseline	3.98	2.74					
Building-InsuUP-MassDW	Increased 2x	Decreased 50%	4.14	3.39					
Building-InsuUP-MassUP	Increased 2x	Increased 2x	3.62	2.44					

Table 6.1.: Table of specific agents trained highlighted in blue. The vertical axis has the 9 building variants and the horizontal axis has the different control algorithms starting with the EnergyPlus baseline.

With 9 different buildings and 6 different DRL control algorithms, only a few of all the possible combinations were trained. Table 6.1 shows the 14 agent-building that were trained. Only the RL24hRNN algorithm was tested for all the buildings. Table 6.2 shows the agents from the baseline building being applied to all the different building variants. Comparing the specifically trained agents against the baseline building agent shows that the baseline

## 6. Climate AI Results

Building type name	Insulation	Thermal Mass	Baseline EnergyPlus Heating [kWh/m2/a]	Baseline EnergyPlus deg-hours < 20C	Baseline EnergyPlus deg-hours > 25C	RL 24h Foresight RNN Heating [kWh/m2/a]	RL 24h Foresight RNN deg-hours < 20C	RL 24h Foresight RNN deg-hours time > 25C	RL 4h Foresight RNN Heating [kWh/m2/a]	RL 4h Foresight RNN deg-hours < 20C	RL 4h Foresight RNN deg-hours time > 25C	RL No Foresight Heating [kWh/m2/a]	RL No Foresight deg-hours < 20C	RL No Foresight deg-hours > 25C	RL 24h Foresight RNN w/o Solar Heating [kWh/m2/a]	RL 24h Foresight RNN w/o Solar deg-hours < 20C	RL 24h Foresight RNN w/o Solar deg-hours > 25C	RL 4h Foresight RNN w/o Solar Heating [kWh/m2/a]	RL 4h Foresight RNN w/o Solar deg-hours < 20C	RL 4h Foresight RNN w/o Solar deg-hours > 25C	RL 4h Foresight flat inputs Heating [kWh/m2/a]	RL 4h Foresight flat inputs deg-hours < 20C	RL 4h Foresight flat inputs deg-hours > 25C
Building-InsuBASE-MassBASE	Baseline	Baseline	8.20	6.22	10.16	5.83	153.68	10.08	6.36	40.67	10.13	8.20	6.22	10.16	6.55	67.60	10.13	7.35	11.87	10.15	8.08	6.14	10.16
Building-InsuBASE-MassDW	Baseline	Decreased 50%	8.40	7.43	13.59	6.09	150.59	13.44	6.68	38.07	13.55	8.40	7.43	13.59	6.79	64.87	13.48	7.56	12.45	13.57	8.28	7.69	13.59
Building-InsuBASE-MassUP	Baseline	Increased 2x	7.89	4.95	7.31	5.31	158.76	7.21	5.86	42.77	7.29	7.89	4.95	7.31	6.09	69.93	7.23	7.02	10.03	7.30	7.77	4.87	7.31
Building-InsuDW-MassBASE	Decreased 50%	Baseline	25.02	242.39	15.69	20.85	598.83	14.84	21.89	429.95	15.30	25.02	242.39	15.69	22.16	479.79	15.11	22.91	323.52	15.43	24.64	251.01	15.66
Building-InsuDW-MassDW	Decreased 50%	Decreased 50%	25.29	252.08	21.36	21.21	600.76	19.65	22.30	423.69	20.02	25.29	252.08	21.36	22.48	477.29	20.03	23.32	326.90	20.16	24.95	263.86	21.31
Building-InsuDW-MassUP	Decreased 50%	Increased 2x	24.63	216.35	9.71	20.36	610.84	9.01	21.31	438.42	9.31	24.63	216.35	9.71	21.72	489.77	9.26	22.28	318.70	9.55	24.22	235.37	9.69
Building-InsuUP-MassBASE	Increased 2x	Baseline	3.98	0.09	8.77	2.37	68.18	8.68	2.82	6.19	8.76	3.98	0.09	8.77	2.93	14.97	8.69	3.60	0.31	8.77	3.92	0.10	8.77
Building-InsuUP-MassDW	Increased 2x	Decreased 50%	4.14	0.07	11.98	2.56	61.80	12.00	2.99	5.67	11.96	4.14	0.07	11.98	3.07	13.95	12.01	3.71	0.22	11.98	4.07	0.10	11.98
Building-InsuUP-MassUP	Increased 2x	Increased 2x	3.62	0.08	6.18	1.92	63.17	6.40	2.44	6.33	6.16	3.62	0.08	6.18	2.57	13.55	6.42	3.26	0.19	6.15	3.57	0.04	6.18

Table 6.2.: Table of the agents trained on the baseline building used to control all the building types. Degree-hours too cold and warm also shown. Blue highlighted shows the agents trained specifically for that building type similar to table 6.1

agent saves more energy, but also causes lower comfort - a consistent pattern that is investigated in more depth in *Further Analysis*.

Energy use is measured in  $kWh/m^2/a$  to make comparison easier with other buildings. The comfort was initially measured as the count of hours outside the comfort range of 20-25 °C but has been updated to *degree-hours*. This is the number of hours outside the range multiplied by the magnitude of the temperature difference.

For ease the agents and controls algorithms will be referred to with the following abbreviated names. Alternatively refer back to figure 5.5 for a graphic representation of the agents.

**EPBaseline** EnergyPlus Baseline at 20.0 °C, which is the standard operation.

**RL24hAIIRNN** RL agent with 24 hours foresight with RNN, all 3 weather inputs.

**RL04hAIIRNN** RL agent with 4 hours foresight with RNN, all 3 weather inputs.

**RL24hNoSolarRNN** RL agent with 24 hours foresight with RNN, but only temperature forecast, no solar radiation information.

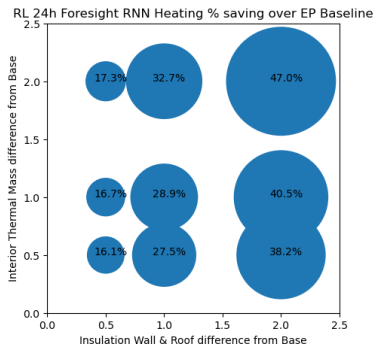
**RL04hNoSolarRNN** RL agent with 4 hours foresight with RNN, but only temperature forecast, no solar radiation information.

**RLNoForesight** RL agent without any weather foresight. Only the work hours boolean is available for the RNN.

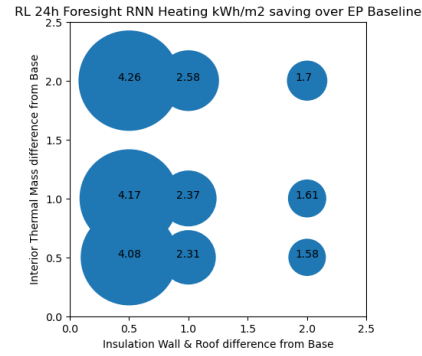
**RL04hFlatInput** RL agent with 4 hours foresight of all 3 weather indicators, but without RNN. The information is the same as the RL04hRNN but without the RNN module in the network.

**EPBaseline190, EPBaseline195, EPBaseline199, EPBaseline205** These additional EnergyPlus baseline models were created with their temperature setting at 19.0, 19.5, 19.9 and 20.5 °C respectively to produce further data for comparing the agents against.

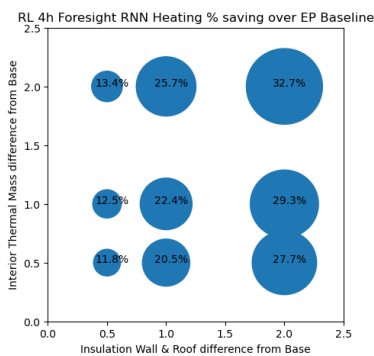
## 6.2. Consistency with Literature



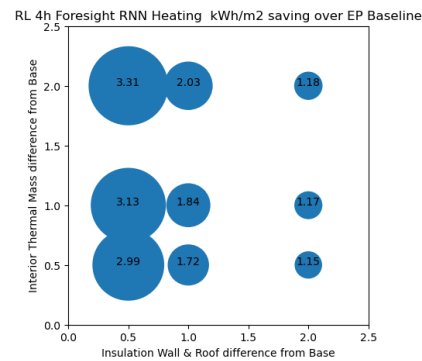
(a) RL24hAllRNN agent's percentage savings for each building type.



(b) RL24hAllRNN agent's absolute savings for each building type.



(c) RL04hAllRNN agent's percentage savings for each building type.



(d) RL04hAllRNN agent's absolute savings for each building type.

Figure 6.1.: Diagrams of relative and absolute savings compared to the EPBaseline operation. The diagrams are laid out similarly to the building variants illustration in figure 5.7, with increasing insulation on the x-axis and increasing thermal mass on the y-axis.

## 6.2. Consistency with Literature

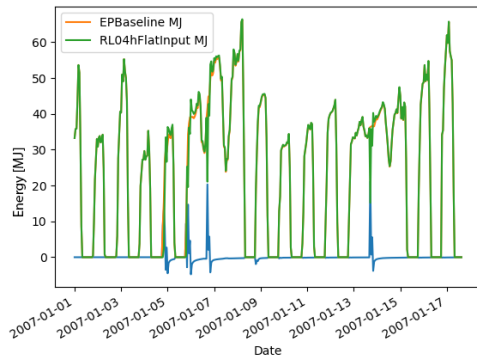
In the literature review, chapter 2, figure 2.17 was presented, showing a plot of previous studies plotted on axes of control and energy. The underlying theory was that a higher level of operational or dampening control as well as higher levels of energy storage and energy availability lead to higher potential relative savings using smart control systems. Figure 6.1 shows the percentage and absolute savings of the RL24hRNN and RL04hRNN algorithms over the EPBaseline. This is consistent with the relationship discovered in the literature review.

### 6.2.1. A Easy Winner - Low Energy and Higher Comfort

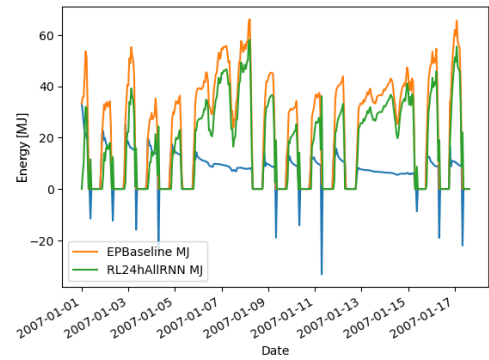
Looking at table 6.2 only the RL04hFlatInput algorithm manages to outperform the EPBaseline on both energy use and comfort. The RL04hFlatInput agent only performs slightly

## 6. Climate AI Results

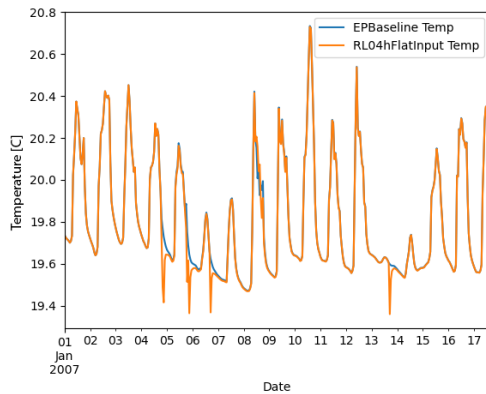
worse on the low-insulated building, where the degree-hours is 5-10% higher, but there is still an energy saving.



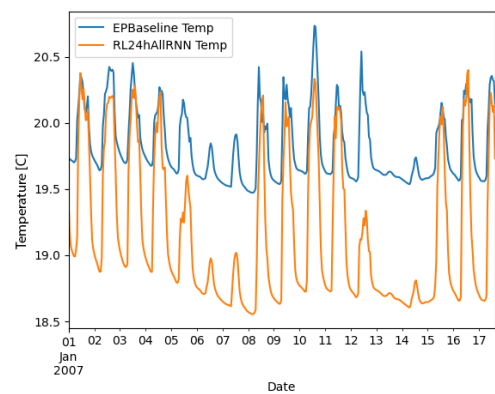
(a) RL04hFlatInput agent's energy use in MJ compared to EPBaseline. Blue is the difference.



(b) RL24hAllRNN agent's energy use in MJ compared to EPBaseline. Blue is the difference.



(c) RL04hFlatInput agent's building temperature compared to EPBaseline.



(d) RL24hAllRNN agent's building temperature compared to EPBaseline.

Figure 6.2.: Comparing the RL04hFlatInput and the RL24hAllRNN performances to the EP-Baseline for MJ energy use and internal temperature over the first 17 days. The first day is Monday.

Figure 6.2a to 6.2d shows the energy use and internal temperatures of the building while being operated by the RL04hFlatInput and the RL24hAllRNN compared to the EPBaseline. The RL04hFlatInput generally performs the same as the EPBaseline, with only minor savings in energy at the end of the school day and slightly lower temperatures at night. The RL24hAllRNN maintains a lower temperature, but both algorithms express the behaviour of reduced temperatures and energy use on the weekends. The RL24hAllRNN energy graph, blue line, shows negative spikes just before the start of the school day. This is also consistent with the temperature graph. The agent appears to be trying to minimize wasted energy during unoccupied hours. It is, however, difficult to tell from the numbers and graphs alone whether this behaviour is equivalent to simply lowering the base temperature and whether the agents are saving more energy compared to the comfort they are compromising. The

only easy winner here is the RL04hFlatInput, which performs slightly better on both parameters, but the overall saving is small.

Figure 6.2d also shows that the RL24hAllRNN decreases the temperature on Friday, either to prepare for the weekend by saving energy or because it does not fully understand the rhythm of the days as described in the *Discussion*, chapter 7, about potential limitations and shortcomings of the individual agents.

## 6.3. Further Analysis

In order to compare energy savings and comfort, the data is broken down per hour, day, week, month, and year to investigate, which algorithms perform best. In addition, new metrics are introduced in order to compare the trade-off between energy saving and comfort, since none of the algorithms, except RL04hFlatInput, are able to maintain the same exact level of comfort while reducing energy use.

### 6.3.1. Energy Savings vs Comfort

#### Reinforcement Learning Reward Function

The reward function, which was covered in detail in chapter 5, works as a measure of the AI's performance. The reward function used gives a penalty for spending energy as well as a penalty for going outside of the temperature comfort range of 20 to 25 °C. We can also use this reward function to give a final, yearly score for an agent by summing all the energy used and counting the degree-hours outside of the temperature range and applying the scaling factors outlined in chapter 5. Figure 6.3 shows the final penalty scores of each agent as well as two EPBaselines. This shows RL04hNoSolarRNN and RL04FlatInput scored higher than both the EPBaseline as well as the RL04hAllRNN. However, referring back to table 6.2, only RL04FlatInput uses less energy and achieves better comfort than the EPBaseline, whereas RL04hNoSolarRNN uses less energy, but has higher discomfort. The performances on this chart are also influenced by how well the agents have learned the task, as their goal is to get as positive a score as possible.

#### Alternative Measures

The following metrics were used to further investigate the trade-off between energy and comfort. With some modifications, these metrics could also be used in place of the current reward function.

The *Linear Ratio Division* was the first metric to be tested but suffers problems as  $d$  approaches or reaches 0. The other two metrics were devised to overcome this problem.

## 6. Climate AI Results

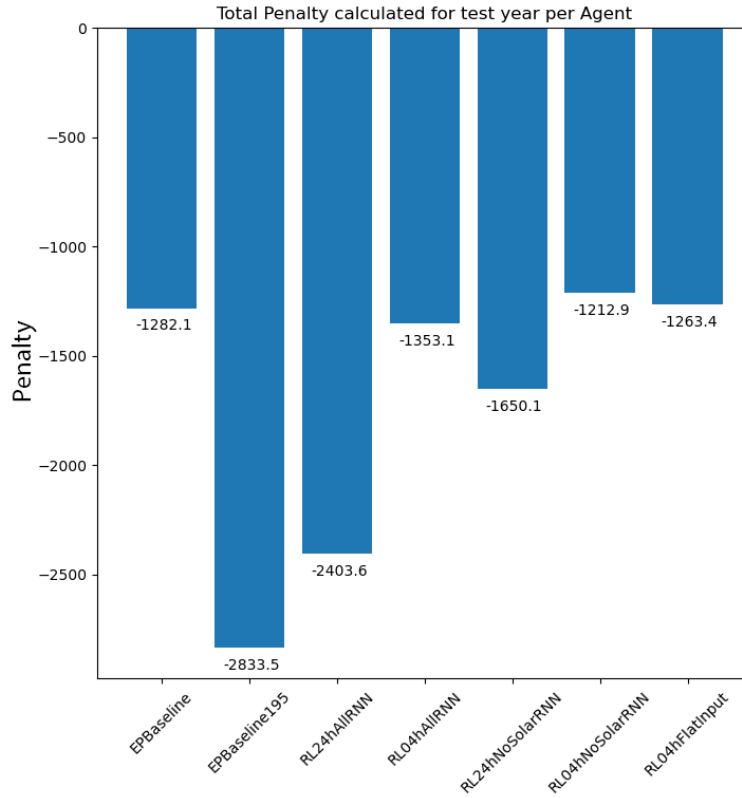


Figure 6.3.: Bar chart of final reward/penalty scores for each agent. Note: the scores are negative, as the agents are given negative rewards rather than positive rewards. The algorithms aim to maximize reward, meaning the least negative is still the highest performing.

$$\text{Linear Ratio Division} \int_{p_0}^{p_i} \frac{Q_b - Q_a}{d} \quad (6.1a)$$

$$\text{Linear Ratio Multiplication} \int_{p_0}^{p_i} \frac{(Q_b - Q_a)(hf - d)}{f} \quad (6.1b)$$

$$\text{Power Ratio} \int_{p_0}^{p_i} \frac{k^{Q_b - Q_a}}{k^d} \quad (6.1c)$$

### Variables in equations

$p$  is the period being investigated and can be either hours, days, weeks or months. The integral over the period  $p_0$  to  $p_i$  is the score for each agent's performance.

$Q_b$  is the energy used by the baseline, usually the EPBaseline.

$Q_a$  is the energy used by the agent.

$d$  is the degree-hours outside the comfort range for the agent.



$h$  is the number of open-hours per period step  $p$ . This can also be a constant.

$f$  is an optional factor, which also prevents  $(hf - d)$  being negative.

$k$  is the power base factor.

The ratios describe the relationship between the energy and comfort and the integral is the aggregate result over a longer period. The higher the better for all the metrics.

### Shortcomings of Metrics

The *Linear Ratio Division* has problems with very low  $d$  and fails if  $d = 0$ . One way to avoid this is to only consider the period where  $d > 1$ , but this is very limiting. The problem with very small  $d$  values is that the ratio starts approaching infinity, making all other readings in the period negligible. For this reason, this metric is very unstable and unreliable.

The *Linear Ratio Multiplication* tries to overcome the division by 0 problem through the multiplication of  $h$  minus  $d$ . This is meant to ensure both multiplication factors are positive. This is because there are edge cases where some poor-performing agents use more energy and decrease comfort, resulting in a double-negative becoming positive.  $h$  and  $f$  can be used in combination to maintain this. As  $f$  increases, the weighting of comfort decreases, making aggressive and compromising agents rank higher.

The *Power Ratio* overcomes the division by 0 problem by raising a factor  $k$  to the power of  $Q$  and  $d$ . This method is also robust to cases where  $d < 1$ , which would in the *Linear Ratio Division* lead to infinities. When  $k > 1$  a higher score is better and when  $k < 1$  a lower score is better. As  $k$  approaches 1, the influence of outliers is reduced and a value of  $k = 1.00005$  was used for most comparisons going forward. The power ratio does not have a factor like  $f$ , which also adjusts the relative weighting.

All three metrics are susceptible to the relative size of the numbers and for these analyses, MJ were used for energy  $Q$  and degree-hours for comfort  $d$ .

In cases where the agent performs better in terms of degree-hours, it may be useful to use  $(d_a - d_b)$  instead of just  $d_a$ . However, this may bring about new edge cases when  $d_a < d_b$  and was therefore not included as all the control algorithms except for RL04hFlatInput have larger degree-hours than the baseline.

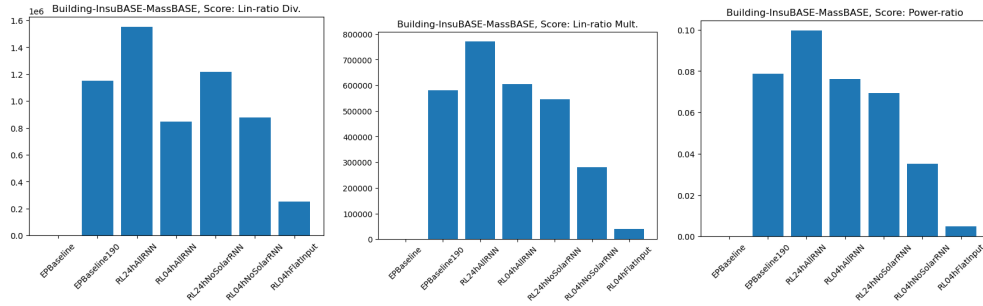
### 6.3.2. Comparing Based on Metrics

Figure 6.4 shows one set of metric bar charts when analysing all 52 weeks using the metrics from equations 6.1. However, this is just one snapshot of their relative performances and depends both on the  $f$  and  $k$  factors as well as the time period examined. The line chart in figure 6.5 shows the plot of the intermediate results that make up the bar chart in figure 6.4b. Looking closely at the first few weeks, RL04hAllRNN outperforms the EPBaseline190 much better than the bar chart summary would suggest. This shows the difficulty of relying on these metrics alone as an indicator of absolute ranking.

For more charts of each building variant and diagram type see the appendix C.

In an effort to return to a simpler metric, namely the *linear ratio division*, the denominator  $d$  must be kept to a value  $> 0$  and preferably  $> 1$ . Therefore, the first 8 weeks were investigated in detail, which is also a period with some of the highest demand for heating due to the cold winter weather. Figure 6.6 shows the bar chart scoring for all 9 building variants

## 6. Climate AI Results



- (a) Bar chart of the accumulated score of the algorithms using the linear ratio division metric.
- (b) Bar chart of the accumulated score of the algorithms using the linear ratio multiplication metric.
- (c) Bar chart of the accumulated score of the algorithms using the power ratio metric.

Figure 6.4.: Bar charts of total score over 52 weeks using each metric.  $f = 4$  and  $k = 1.00005$  used.

over the first 8 weeks using the *linear ratio division* metric. This metric is very susceptible to outliers and therefore the 9 building variants differ, but one significant pattern is that RL04hAllRNN always scores higher than RL24hAllRNN.

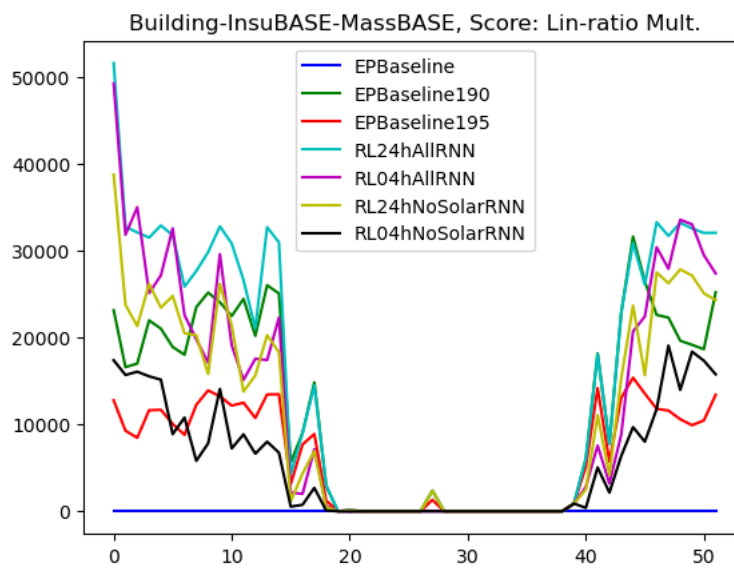


Figure 6.5.: Line chart of the scoring over 52 weeks used in Figure 6.4b with *linear ratio multiplication*.

## 6. Climate AI Results

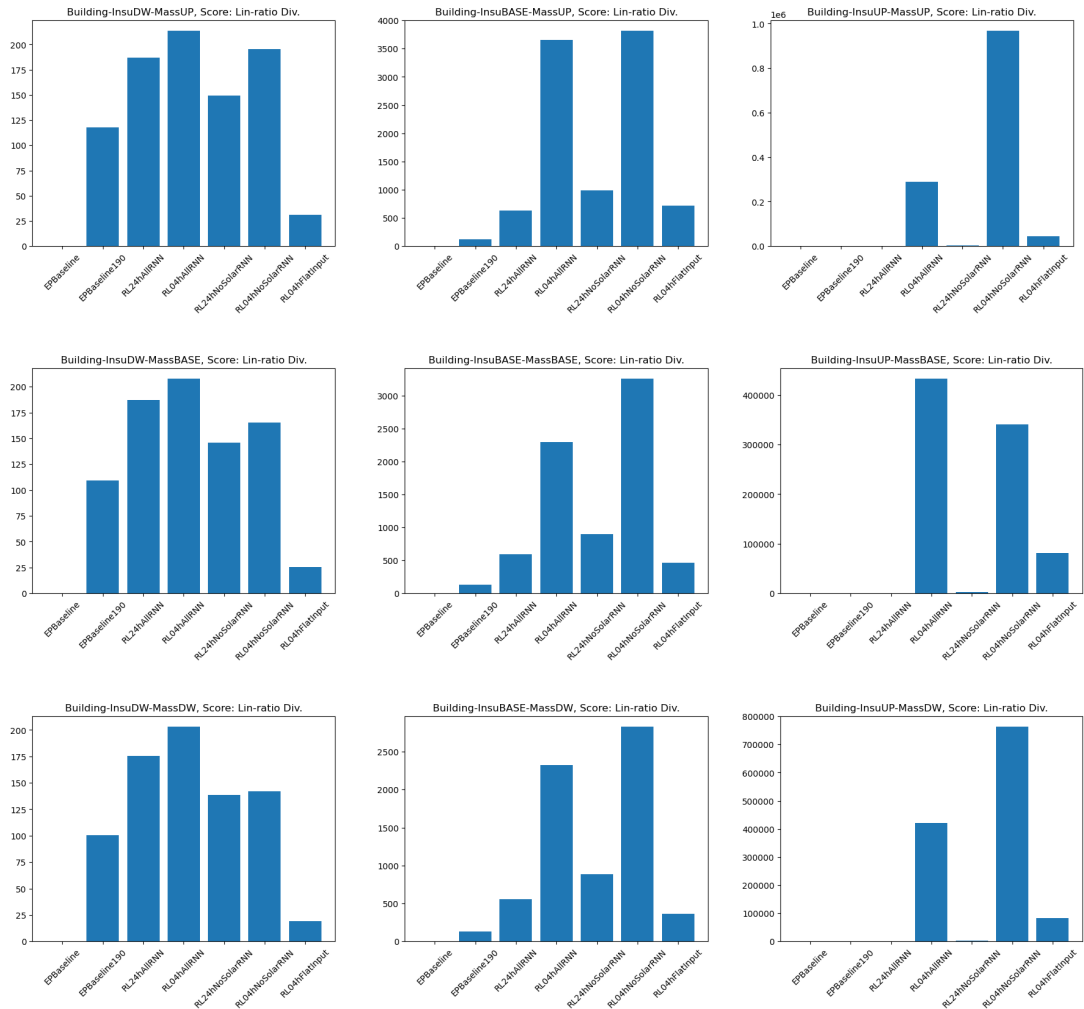


Figure 6.6.: Bar charts of each building variant using the *linear ratio division* metric over the first 8 weeks of the year to avoid  $d$  being significantly less than 1.

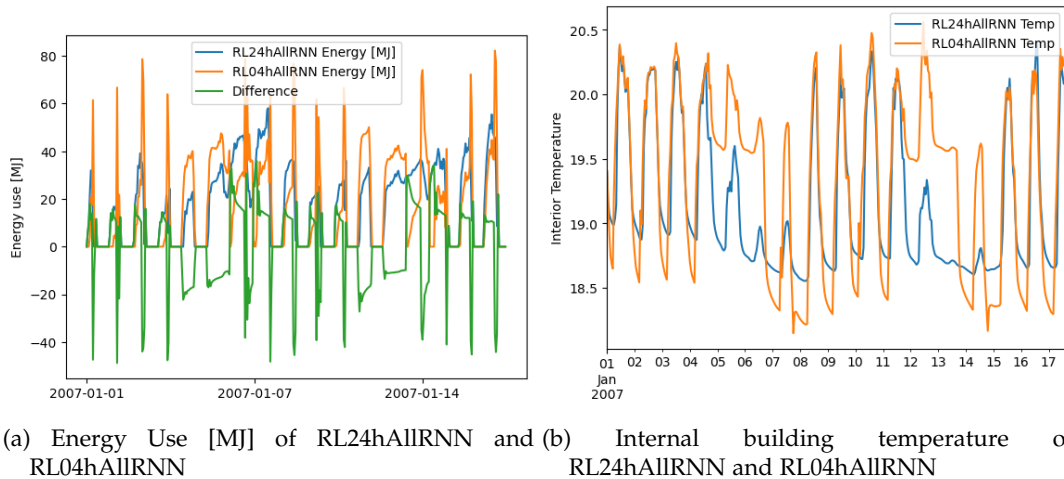


Figure 6.7.: Baseline building energy and temperature comparison of RL24hAllRNN and RL04hAllRNN

### 6.3.3. RL24hAllRNN vs RL04hAllRNN

Other tools than the above-mentioned metrics must be used to more clearly ascertain the performance differences and each control algorithm and further investigate the differences between the RL24hAllRNN and RL04hAllRNN.

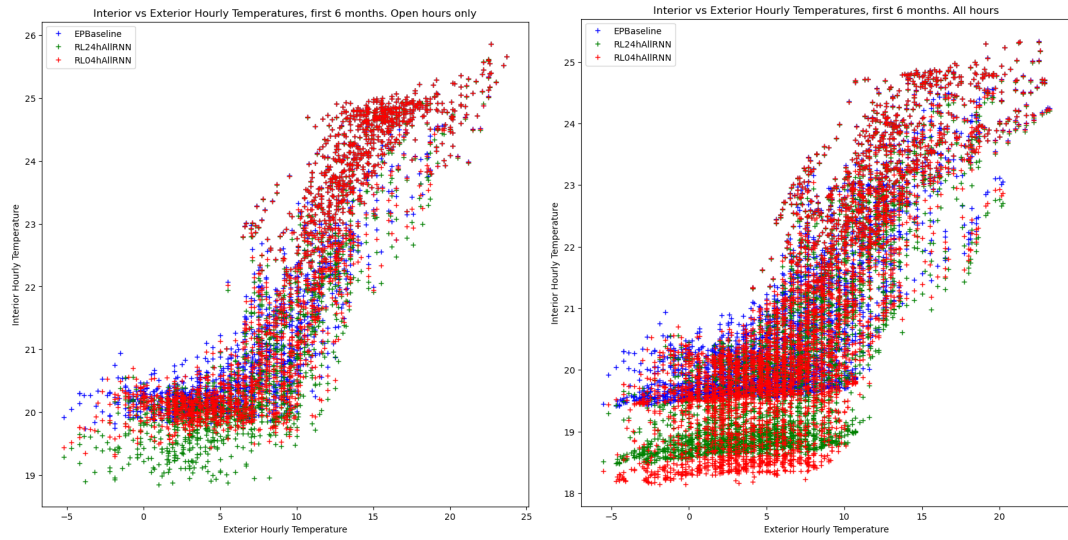
Figures 6.7a and 6.7b compare the RL24hAllRNN and RL04hAllRNN based on energy use and interior temperature. This shows that RL04hAllRNN uses even more energy in the morning than RL24hAllRNN and even the EPBaseline as shown in figure 6.2b. More interestingly the temperature graphs show that the RL04hAllRNN has learnt to heat adequately on Fridays too, but it also lowers the weekend and nighttime temperatures even further than RL24hAllRNN.

Figures 6.8a and 6.8b further illustrates RL04hAllRNN more stable performance against RL24hAllRNN. During the opening hours in figure 6.8a RL04hAllRNN maintains an interior temperature much closer to the EPBaseline. The full plot of all hours in figure 6.8b also shows that RL04hAllRNN reaches much lower temperatures overall, presumably during night and weekend hours. This hints at the two control algorithms having learnt to prefer different minimum heating set-points, which is an interesting detail. This may change with further tweaks to the DRL setup or by allowing them 41 actions with more fine-grained control as discussed in chapter 5.

This result of the RL04hAllRNN performing better than the RL24hAllRNN can also be revealed in the metrics by tweaking the parameters  $f$  and  $k$ . When the parameters are adjusted to prioritise comfort, such  $f$  decreasing in value, RL04hAllRNN ranks highest. Equally when  $k$  increases from 1.00005 towards 1.5, again RL04hAllRNN comes out more favourably, though using this *power ratio* metric with a larger  $k$  outliers do also become more influential.

Overall this suggests that the RL24hAllRNN is more aggressive and saves more energy but perhaps has not learnt as optimally as the RL04hAllRNN, which performs more stably, as

## 6. Climate AI Results



(a) Interior and exterior hourly temperature plot during **opening hours** only for EPBaseline, RL24hAIIRNN, and RL04hAIIRNN

(b) Interior and exterior hourly temperature plot for **all hours** for EPBaseline, RL24hAIIRNN, and RL04hAIIRNN

Figure 6.8.: Interior and exterior hourly temperature plot for the first 6 months.

seen by the behaviour on Fridays, nights and weekends. It also performs better overall and has learnt a lower heating set-point for night and weekend hours. This may also be partially to do with the RNN's issue with vanishing gradients in long sequences, causing the RL24hAIIRNN to not learn well despite having more information available.

### 6.3.4. Better Than Lowering the Thermostat

While the above analysis shows that RL04hAllRNN performs better than RL24hAllRNN, further evidence presented here shows that the DRL agents also outperform a simple lowering of the thermostat. Not only in terms of comfort but also remarkably saves more energy!

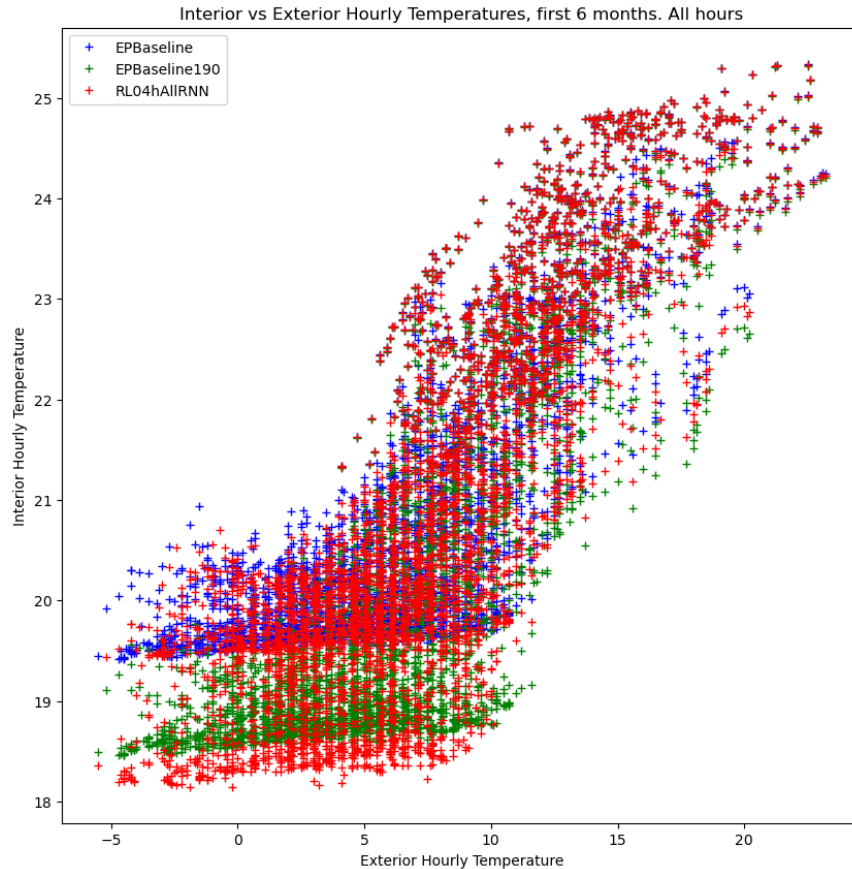


Figure 6.9.: Interior and exterior hourly temperature plot during **all hours** for EPBaseline, EPBaseline190, and RL04hAllRNN. First 6 months of the year.

Figure 6.9 compares a lower thermostat of 19.0 °C with the RL04hAllRNN for all hours in the first 6 months. The RL04hAllRNN has a broader range but maintains a higher temperature during the open hours compared to simply lowering the temperature. It does keep a slightly lower nighttime temperature, which does help it overall. However, this slightly lower temperature is not the only thing that makes RL04hAllRNN better than the EPBaseline190 since the RL24hAllRNN keeps a higher lowest set-point as shown in figure 6.8 and still saves more energy as shown in figure 6.10.

Figure 6.10 shows the weekly degree-hours too cold plotted against the energy use for each algorithm as well as some EPBaselines. The figure shows that the RL control algorithms perform better than the lower temperature set-points both in terms of comfort and even energy use.

## 6. Climate AI Results

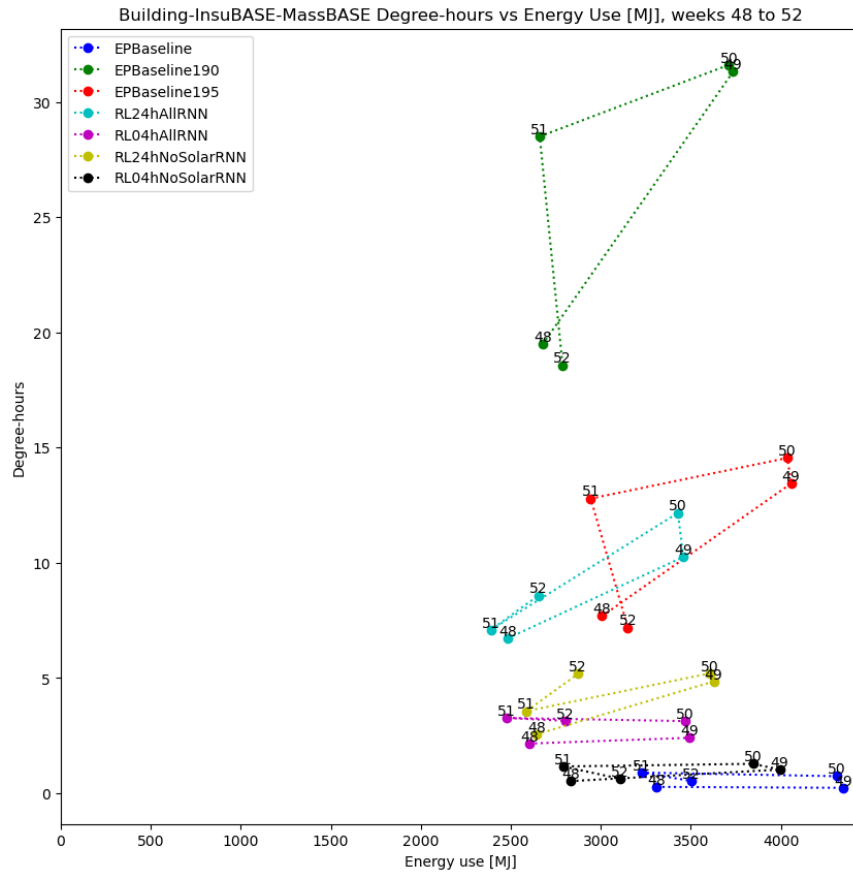


Figure 6.10.: Degree-hours plotted against energy use MJ for the baseline building for weeks 48 to 52.

Another interesting trend in figure 6.10 is that for all controls, except the 4h RNNs, weeks 49 and 50 have a higher degree-hour compared to weeks 48 and 51 as well as higher energy use. This suggests that the 4h RNNs were better able to maintain a good level of comfort for a similar amount of energy, whereas other algorithms including the EPBaselines compromised comfort. This is even clearer when just looking at RL24hAllRNN (light blue) and RL04hAllRNN (purple). The shape of the plots is a flat rectangle for RL04hAllRNN, showing consistent comfort, but for RL24hAllRNN weeks 49 and 50 are less comfortable while using a comparable amount of energy.

The type of graphs used in figure 6.8 and 6.9, which show interior over exterior temperatures also suggest that an adaptive comfort measure might benefit the algorithms' comfort score compared with the current  $< 20$  °C limit.



### 6.3.5. Low-insulation Buildings

Recalling the original pattern from the literature review and from the beginning of this chapter, higher insulated buildings have higher relative savings. This is also evident in figure 6.11 showing the degree-hours over energy use for the low-insulation, baseline thermal mass building. The algorithms are able to save some energy, but the compromise in comfort is much larger. Unlike the baseline building shown in figure 6.10 the algorithms are not able to use less energy than EPBaseline190 while being more comfortable than EPBaseline195. This suggests as the buildings become less insulated the performance of the agents moves closer to the performance of a simple heating set-point because they cannot take advantage of the dampening effect of insulation.

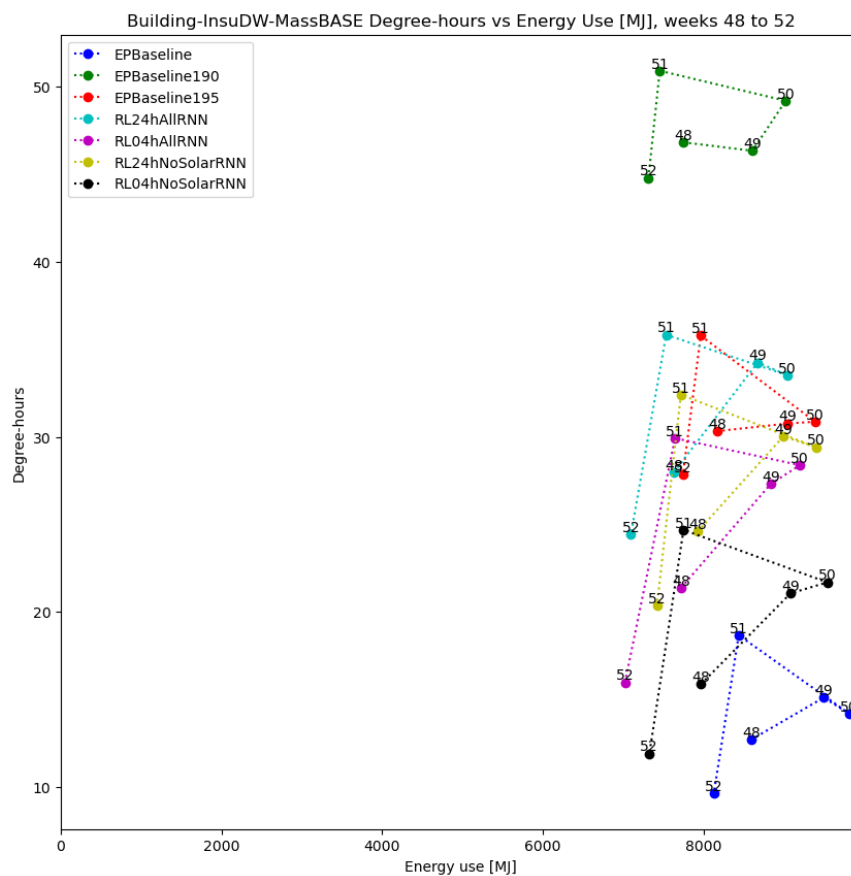


Figure 6.11.: Degree-hours plotted against energy use MJ for the low-insulation, baseline thermal mass building for weeks 48 to 52.

All 9 building variant plots can be found in the appendix C.

### 6.3.6. Comparing Long Term Trends

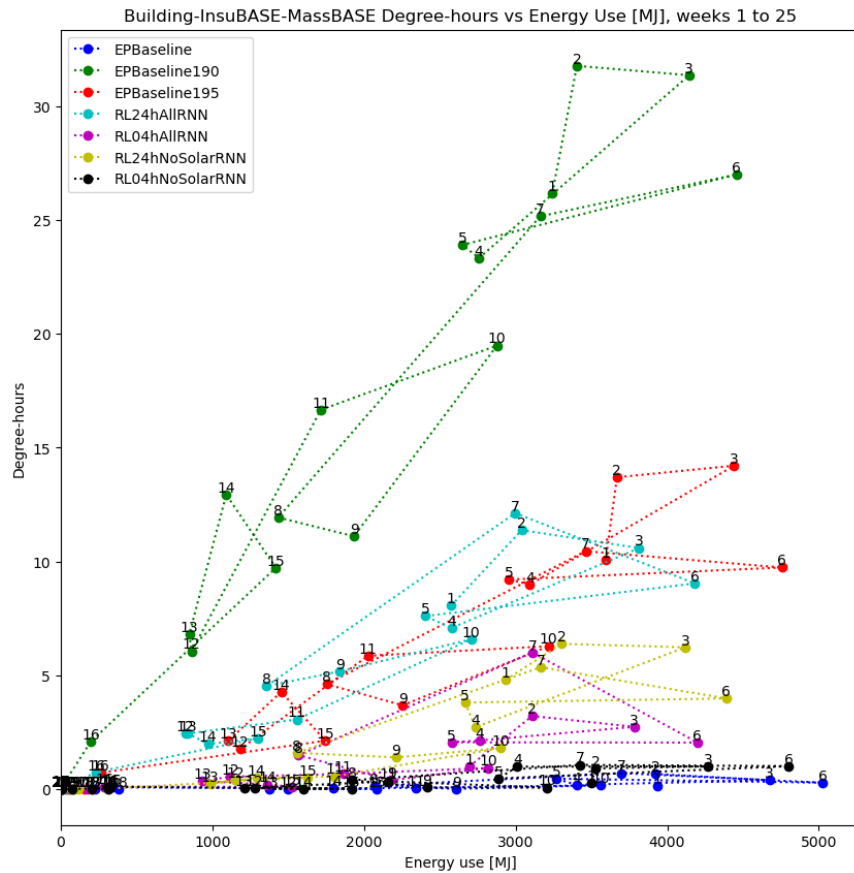


Figure 6.12.: Degree-hours plotted against energy use MJ for the baseline building for weeks 1 to 25

Figure 6.12 shows a plot of degree-hours over energy, but over a longer time span, covering the first 25 weeks of the year. Again this shows that the RL24hAllRNN uses less energy than EPBaseline190 while achieving better comfort than EPBaseline195. The RL04hAllRNN is not far off the energy saving but is considerably more comfortable every single week.

## 6.4. Alternative Baselines

When measuring improvement, a baseline is required and this baseline can have a large impact on the results. For this reason, the previous sections focused more on comparing the agents to each other to examine the trade-offs between energy saving and comfort. The original EPBaseline uses a simple set-point of 20 °C and is managed by the EnergyPlus algorithms. However, the agents display behaviour of decreasing the temperature at night and increasing it during the day. See figure 6.2 for reference. This can be mimicked using a rule-based controller with two temperature set-points; one for daytime and one for nighttime.

This is commonly found in electronic thermostats as well and is a good improvement over a static set-point.

The rule-based controllers used in this analysis have two set-points. The daytime set-point is always 20 °C and starts at 6 am in order to heat up the building before the 8 am school day starts. The nighttime set-point varies with each controller and is in the range of 17.8 to 19.5 °C. The nighttime set-point activates at 1 pm, as by this time the school is already heated up by the student activity and the sun, in addition, the school ends at 3 pm, hence the assumption is that no additional heating is required until the next day. In addition, on weekends the temperature set-point is the same as the nighttime set-point, leading to additional savings.

The rule-based controllers are referred to with the following naming abbreviations:

**RuleBased178, RuleBased180, ..., RuleBased195** The number at the end indicates the nighttime temperature in tenths of a degree. For example, RuleBased178 has a nighttime temperature set-point of 17.8 °C.

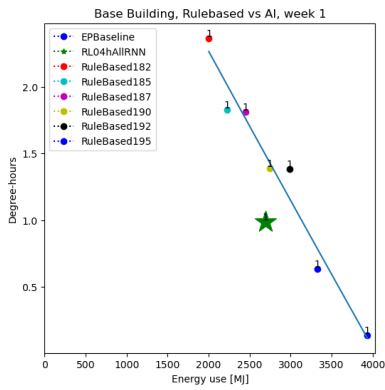
Simulating the multiple rule-based controllers it is possible to draw a trend line and compare the RL agent's performance against it. Figure 6.13 shows the performance week by week for the first 6 weeks of the year. These plots show that the agent RL04hAllRNN outperforms the rule-based controllers in 5 out of 6 weeks. Only in week 4 does RL04hAllRNN perform at the trend line of the rule-based controllers. Figure 6.14 compares them on a month-by-month basis, showing the performance through different seasons of the year. RL04hAllRNN outperforms the rule-based controllers by the largest margin in the winter when heating demand is at its highest. In the third month, March, the RL04hAllRNN performs slightly worse than the trend line, but overall for the whole year the RL04hAllRNN agent outperforms even the rule-based controllers.

### Savings Over Rule-Based Baseline

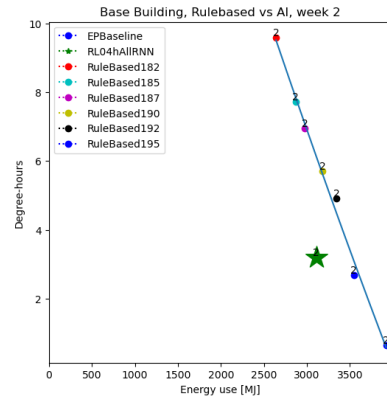
The yearly savings of the RL04hAllRNN agent over the initial EPBaseline was 22% energy saved, however, comfort was compromised and the EPBaseline's performance as a robust baseline was put into question when compared to rule-based controllers.

From figure 6.14 we can see that the RL04hAllRNN agent has a similar comfort level as somewhere between RuleBased192 and RuleBased195. The yearly energy savings of the RL04hAllRNN agent over the RuleBased192 is 3.45% and over the RuleBased195 is 11.17%.

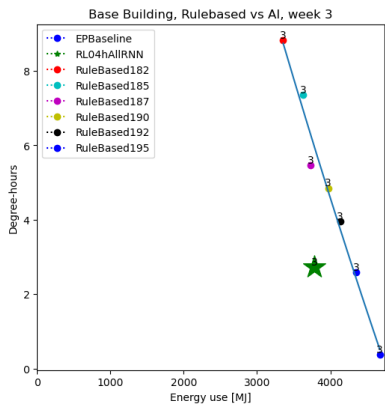
## 6. Climate AI Results



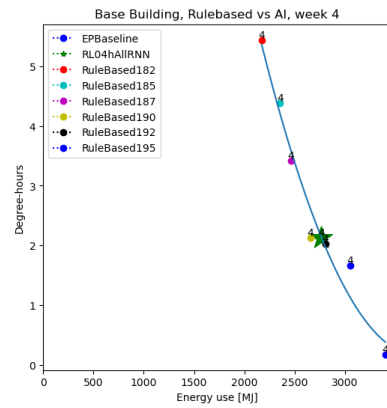
(a) RuleBased vs agent, week 1



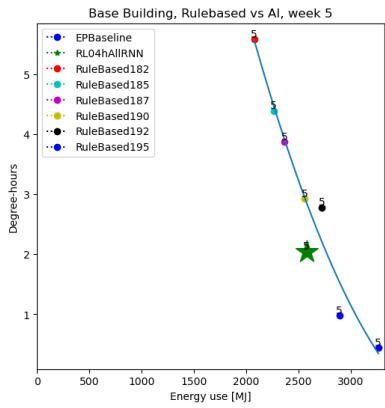
(b) RuleBased vs agent, week 2



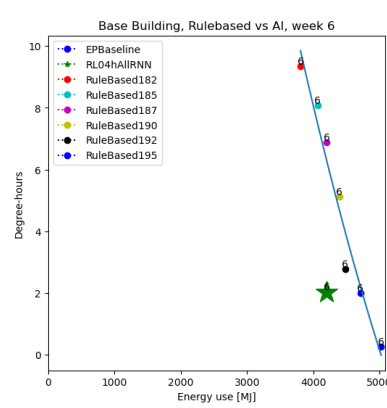
(c) RuleBased vs agent, week 3



(d) RuleBased vs agent, week 4



(e) RuleBased vs agent, week 5



(f) RuleBased vs agent, week 6

Figure 6.13.: Degree-hours plotted against energy use MJ for the rule-based controllers with trend line against the RL04hAIIRNN agent (green star) for weeks 1 to 6

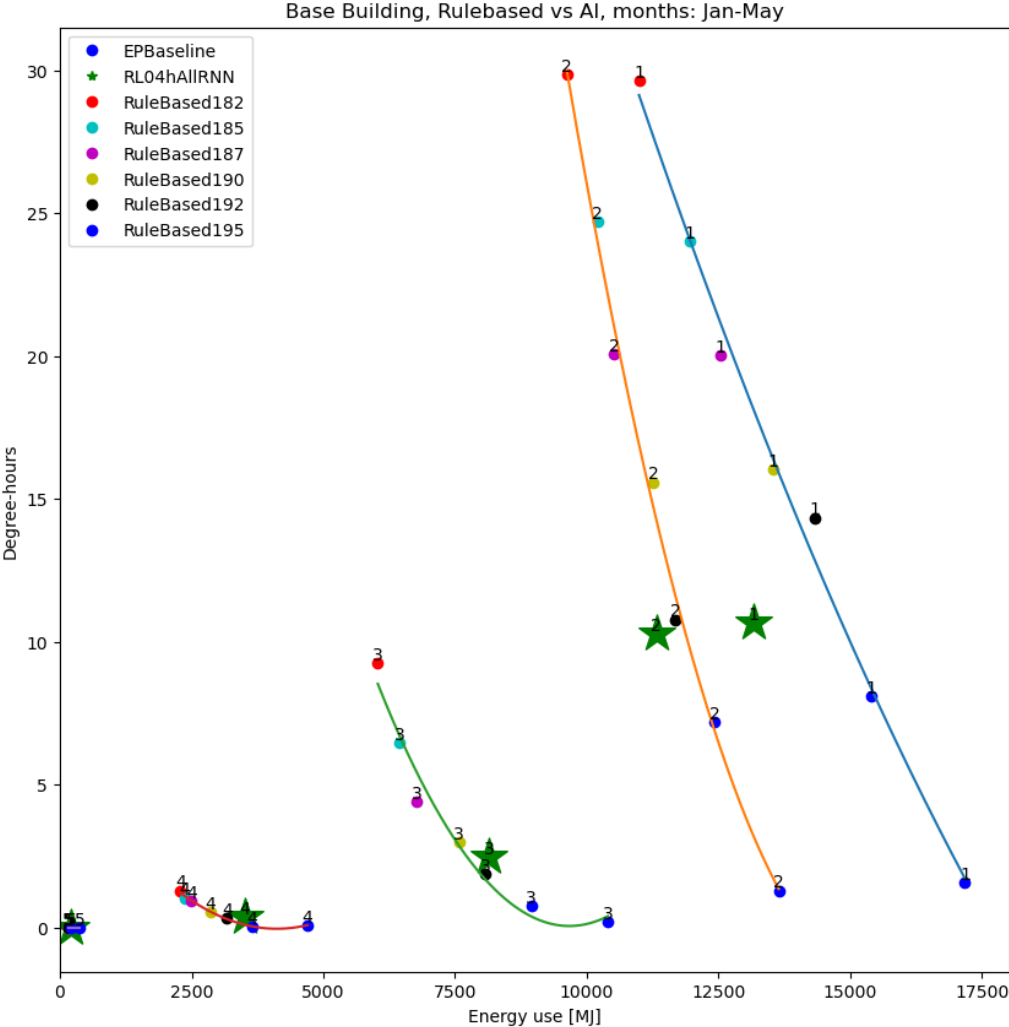


Figure 6.14.: Degree-hours plotted against energy use MJ for the rule-based controllers with trend line against the RL04hAllRNN agent (green star) for the months of January, February, March, April, and May.

## 6.5. Reinforcement Learning Metrics

Another strategy to compare the RL agents is to look at their training metrics - the data generated during the training phase. Looking at the rewards over the final 11 years of training in figure 6.15, we can see that RL04hNoSolarRNN performs better than both RL04hAllRNN and RL24hAllRNN, scoring a higher reward. This aligns with the overall reward scores in figure 6.3. It also highlights that the weighting between comfort and energy use in the reward is tilted more towards prioritising comfort as figure 6.10 shows that RL04hAllRNN is saving more energy but compromising comfort marginally more than RL04hNoSolarRNN.

Figure 6.16 shows the agents' loss over the 35 years of training data. The loss type used is the mean square error and measures the difference, loss, between the predicted best action and the truth. Using mean square error gives a larger weighting to outliers. As would be expected the loss starts off very high and gradually decreases over time. Again, RL04hNoSolarRNN has a lower loss than the other two algorithms, indicating it has learned to take better actions. However, this can also be due to RL04hNoSolarRNN being the network with the fewest inputs, therefore less data to learn from, resulting in faster learning as there may be less noise. There were also periods where RL24hAllRNN performed with a lower loss relative to the other two. This also comes down to some randomness, both in the initialisation of the network as well as the randomness in the batch picking used for experience replay. Additionally, figure 6.16 also indicates that the agents have not fully learned the optimal action as a loss of 0 would indicate it chooses the perfect action each time. Therefore, these agents would benefit from longer training run to decrease the loss further.

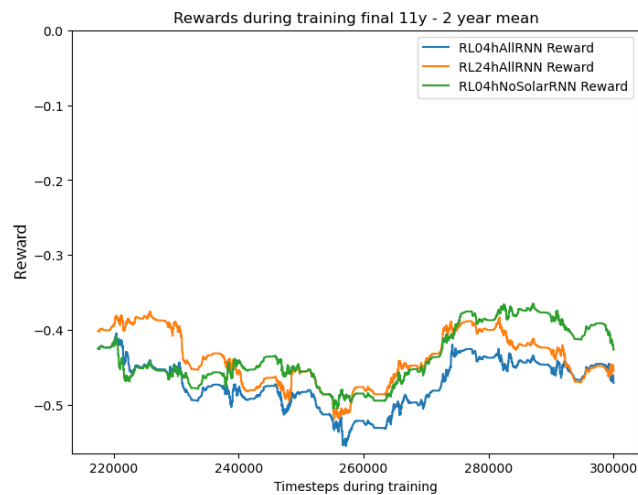


Figure 6.15.: Agents' reward with 2-year mean for the final 11 years of the total 35 years of training simulation

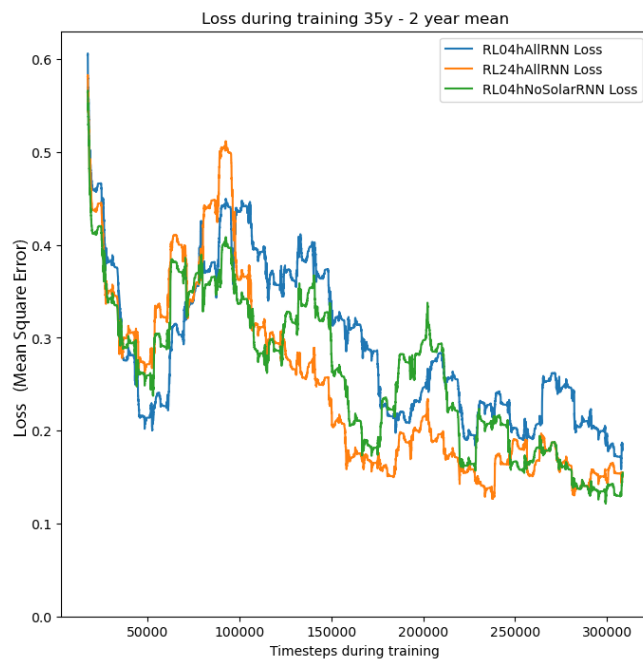


Figure 6.16.: Agents' mean square error loss for the whole training period of 35 simulated years.





## 7. Discussion

### 7.1. Results

#### 7.1.1. Achieving a 1% CO<sup>2</sup> Reduction

As mentioned in the Introduction, chapter 1, in the UK a 3% reduction in operational energy for buildings would result in a 1% reduction of UK CO<sup>2</sup> emissions. The RL04hAllRNN agent achieves this, even against the RuleBased192 which has a higher discomfort level than the DRL agent. This is despite the current agents being undertrained as indicated by the loss graph in figure 6.16 and more advanced DRL algorithms could improve this even further. Because the building is very simple, these savings do not come from advanced shading systems or modern heat pumps, but simply from anticipating future changes in outdoor weather and how that influences the indoor climate. Additional controls would further improve these results.

#### 7.1.2. Limitations

The building investigated in this thesis is timber-based and therefore has a low thermal mass compared to concrete or stone buildings. The results may, therefore, not fully reflect heavier buildings. However, thermal mass has a slow release time and is not controllable by the smart control system, and may therefore not make a significant difference overall.

The building investigated has very few controls and a similar building with movable shading may perform much better. In addition, the location of Wales also influences the temperatures and available solar energy.

For the tests and simulations, the user schedule was simplified and does not reflect the real variation in daily use of the building, especially considering its after-hours use for various community activities. This is a limitation for relating these theoretical savings to the real world where the use is more unpredictable.

Only a few types of RL algorithms were tested in a few combinations, and these do not represent the current state of the art, nor are they exhaustive of the full spectrum of available control algorithms.

### 7.2. Real-world Impact

The current low insulation and high thermal mass building variant, does not fully represent the far end of the spectrum of buildings. The baseline building is a timber building, but it would be interesting to see the savings for a low-insulated concrete or stone building.

## 7. Discussion

Especially since these represent a large part of the historic building stock and historically significant buildings, which cannot be easily retrofitted.

### 7.2.1. Stakeholder Decisions

**Building owners** may now consider whether smart control operation systems can benefit their buildings, either in terms of comfort, energy savings, or both. This must be weighed against the option of a traditional retrofitting or upgrading of the building fabric. The main decision drivers are cost, availability, and comfort.

**Costs** Overall a smart control system upgrade may cost less than a full retrofitting as the material costs may be lower, however, the expert consultancy costs would likely be higher for installing a smart control system. In addition, it is likely that the building operation system and mechanical equipment have not been inspected or adjusted for an extended period of time and a smart control upgrade would potentially find other areas of improvement.

**Availability** Smart control system providers already exist, but it is likely a smaller market than general contractors. Some large providers are Cisco, Bosch, Honeywell, Siemens, Johnson Controls, and newer startups include Scanalytics Inc., Phaidra, Viviot, Spectral, Brainbox AI, ClevAIR and many more. A retrofit or building fabric upgrade may also have a lower perceived risk for the building owner as they may have more familiarity with that route.

**Comfort** While the **DRL** agents shown in this thesis did improve comfort compared to some rule-based controllers at similar energy use, there is still no guarantee that a smart control system can provide both increased comfort and reduced energy expenditure. A building fabric upgrade on the other hand, such as wall insulation, improved windows, or even solar panels or energy storage can be guaranteed to provide improved comfort at the same or lower energy expenditure.

Overall, for a **Building owner** it comes down to cost and risk. A smart control system is potentially much cheaper but may have less guarantee of achieving the desired results and the results may not be as large as imagined. A building fabric retrofit on the other hand is more expensive, but also has a lower perceived risk due to its familiarity and can guarantee an increase in both comfort and energy saving.

A final consideration is to refer back to figure 2.17, which shows the potential energy savings of a building on the axes of potential energy and operational control. This can be used as a framework to assess and compare multiple buildings' potential savings from smart control. If the building lies toward the lower left, with low insulation, low operational control systems such as shading, low energy storage, or no variable energy pricing etc, a retrofit would be most beneficial. However, if the building is already well-insulated and has multiple control systems to regulate energy use a smart control system upgrade would be most beneficial as the marginal benefits from additional insulation diminish.

From a national or global perspective, the lower left quadrant buildings with low insulation and low energy control benefit the most from improvements as their absolute energy use is much higher than high-performance buildings. While a retrofit is the best option as outlined above, a cheaper smart control system may be the halfway option in the short term if funding is a major limitation.

**Historic buildings** have the additional limitation that a fabric retrofit may not be an option due to heritage restrictions that prevent major physical changes. In this case, improving the building operation may be the only available option. However, if this is combined with additional energy storage or variable energy pricing, perhaps due to variable renewable energy production, a smart control system can greatly improve the overall building operation, despite the limitations.

#### 7.2.2. Standardisation of Equipment

A limitation to the immediate implementation of smart control systems is the large number of different pieces of equipment that must be centrally controlled. Each piece of equipment may have its own control system or interface and some may be inaccessible without the manufacturer's assistance or permission. This is similar to how this thesis has to overcome the issue of exchanging data with EnergyPlus via the [API](#). Standardisation of interfaces and data formats would greatly improve this as well as the right to adjust such equipment without the manufacturer's restrictions.

### 7.3. Future Improvements and Excluded Complexities

With more time or further studies, the following improvements could be made and complexities currently excluded could be taken into account.

#### 7.3.1. Advanced Building Control

In addition to opening the ventilation louvres, the [MVHR](#) bypass can be used to disable the heat recovery of the unit, giving greater control of the ventilation system.

As well as increasing the number of actions from the current 11, higher set-points can also be used to provide a more rapid response by the system. However, moving away from set-points altogether and instead using heating power (kW) seems to be the best solution. The current 'Ideal Air' EnergyPlus object takes in a set-point, but the internal workings are unknown and likely non-linear. Meaning that if the set-point is 20 °C, the power input to the 'Ideal Air' component would be different depending on the building's current temperature, making the energy added less predictable by the agent. Using radiators or separate air heating sources in the digital twin could be one solution. However, the next limitation is the hot water tank of only 300 litres. This limits the total available energy in the short term and inputting the temperature of the hot water tank as one of the agent inputs would be beneficial as that correlates to how much the agent can raise the temperature in a short amount of time.

Holiday periods can also be reinstated as this would make the simulation more realistic, however, depending on the building and use cases it may be easier to simply have a low thermostat setting for those periods.

Electronically controlled internal shades could be added to prevent overheating or even reduce heat loss slightly at night. Additionally, the external shades are currently fixed overhangs but could be retractable. For the current building, without smart [RL](#) control the energy

## 7. Discussion

saving would be about  $1 \text{ kWh/m}^2/\text{year}$  or roughly 2,000 kWh to not have shading during the heating periods based on the original PHPP modelling. The EPBaseline uses just over 20,000 kWh per year, hence movable shading could be a saving of up to 10% compared to fixed shading. The shading should however be deployed during the summer to prevent overheating.

### 7.3.2. Building Physics Variations

The climate and weather have a large impact on the available solar energy, so the results and performances would differ depending on the location. This could be a future area of study.

The current definition for comfort as a temperature range between 20-25 °C can be replaced by something more complex such as predicted mean vote or a measure which also takes into account outdoor mean temperatures. Furthermore, the reward function could also be replaced by one of the performance measures developed in chapter 6.

The current comfort reward does not take into account the occupancy density. It only activates during the opening hours but does not adjust for the fact that the occupancy is not constant throughout the day. An added layer of complexity would be to include occupancy, which may give interesting results which prioritise comfort only when the building is sufficiently occupied. This is an underutilised metric for large buildings such as open offices and educational buildings, where the occupancy may be low at certain times, yet the building is still being operated to achieve the same level of comfort for the whole building, even if it is mostly empty.

Additional building controls can also be incorporated in the AI system such as lighting, however, with modern LED lights this is an area of lower potential energy savings.

### 7.3.3. Reinforcement Learning Improvements

The current agents use simple RNN units, which suffer from vanishing gradients and may be resulting in low performance. A simple upgrade would be to use GRUs instead or even LSTMs, which are used in several other studies. This should increase performance overall.

The RNNs are used to give the agent a concept of future events, but the current information describing the current time may be insufficient. In the *current readings* the agent receives *day of week* - an integer from 1 to 7, which is then normalised. However, close inspection of the results indicates that the models sometimes mistake Friday for being a weekend and therefore lowering the temperatures. The aim was that the *future work hours* boolean would help with this, but that does not seem to be the case as the agent without any weather foresight still sees the *future work hours*, but performs the same as the baseline. This could either be indicating that the time information is insufficient or that future weather data is required to make better decisions than the baseline.

The current reward or penalty for energy use does not scale across the building variants. This means that the less insulated building are being penalised more for using energy as their baseline use is higher. This would suggest that those buildings would compromise more on comfort, however, that is not easily visible in the results. This balance between the rewards across the different building types is another area for further study. In addition,

the performance metrics presented in chapter 6 can also be experimented with as a reward signal, though they are not always perfect metrics.

A major issue both in training and post-analysis is the relative weighting of energy use vs comfort. A more flexible approach would be to investigate the sub-field of multi-objective RL. This uses multiple objectives that can be defined separately, whereas the current implementation forces energy use and comfort to be combined into a single reward. With this method, it is potentially possible to alter the weighting of comfort and energy saving at will during operation.

The gamma or discount rate can be increased from 0.9 to 0.99 to prioritise future rewards more.

To overcome some of the current limitations of DRQN more advanced network architectures can be used such as actor-critic as seen in (Christodoulou, 2019), which uses a second network to critique and improve the performance of the first. Secondly, enabling concurrent actions, as is done in a complex game environment by Harmer et al. (2018), would greatly enhance the capability of operating more complex buildings with more control objects. This would directly address the problem of the exponentially growing number of actions.

To combat the problem of EnergyPlus's slow calculation speed, the digital twin model can be replaced by a NN-based surrogate model. This new model would also need validating, but the potential advantage is the much faster computation for each time step. Alternatively, multiple EnergyPlus models can be run in parallel with random actions chosen and saved. These multiple runs can then be used as a memory for experience replay or general training for a single DRL model. Running multiple simulations in parallel is possible and has been achieved in this thesis during the *genetic algorithm* calibration tests - see appendix B. Alternatively, multiple similar agents can be trained in parallel and then aggregated. This was not attempted for this thesis.

Another potential improvement in performance would come from simply using a larger Neural Network (NN), but this would increase training time and necessary training steps.

The current models and agents all use a 1-hour time-step, which is very coarse. This means the agents can only change the temperature once per hour. Another improvement would be to decrease the time-step to 15, 10 or 5-minute intervals. This would require larger changes to the code and state data, as the weather data would have to be interpolated and it is unknown whether the EnergyPlus API would provide weather data that is already interpolated or not.

## 7.4. Benchmarking

This thesis addresses the lack of openly comparing multiple algorithms as well as multiple buildings to uncover the relative influences of both on energy saving and comfort. However, an underlying issue is that, like most existing research presented here, the algorithms and buildings are different and potentially not comparable to the existing literature in a rigorous and empirical way without countless assumptions. One solution would be to establish benchmark data sets, as is common in other ML sub-fields. One such dataset could be the United States Department of Energy's *reference buildings*, which are already in EnergyPlus format. This is but one example and does not fully represent a European climate, other examples include BOPTEST (Arroyo et al., 2021) (Blum et al., 2021), SimApi (Pallonetto et al.,

## 7. Discussion

2019), and EnerGym (Scharnhorst et al., 2021). The point is, to truthfully argue that one smart control system is superior to others, it must be tested in a controllable and standardised way.

## 8. Conclusion

### 8.1. Key Findings

The RL agent RL04hAllRNN outperforms not only the simple EPBaseline but also the improved rule-based controller baselines, which automatically adjust the temperature at night and on weekends, showing that there is great potential for additional savings through smart control. This also achieves the necessary 3% savings required for a 1% reduction in  $CO_2$  in the UK.

The smart control algorithms differ depending on available information about the future as well as their internal network model. Overall more advanced models perform better, but a challenge is balancing energy savings with maintaining comfort. For specific improvements on the RL approach and algorithms, see the *Discussion*, chapter 7.

The results presented in chapter 6 are consistent with the findings from the literature review showing that increased energy availability and energy control lead to greater relative energy savings as shown in figure 2.17 based on literature and figure 6.1 based on the results from this thesis. This in itself is a useful framework for assessing which buildings are best suited for smart control.

It is, however, important to note that absolute savings are higher in lower-performing buildings, as shown in the diagrams in figure 6.1. Therefore, these would benefit the most, both in terms of cost savings and global  $CO_2$  emissions, from additional attention and smart control combined with other proven solutions such as variable energy tariffs or retrofitted energy storage.

### 8.2. Changes to Design and Retrofitting

Designers of new buildings should keep in mind their building's potential for additional savings through smart control, perhaps using the framework from figure 2.17, through increasing available energy input, storage, and control.

This change to a smart controlled building does also necessitate a change or upgrade to current building management systems and building equipment as their interconnection and centralised control become more important. This means providing accessible APIs to mechanical equipment, window shading and all other electronically controllable building equipment.

For retrofitting buildings that score low on energy storage and control, savings through smart control may not be realisable without significantly compromising comfort, if a second measure is not available. This could be installing heat storage on-site or taking advantage of a variable price energy supply.

## 8. Conclusion

Variable price energy may become more common for larger institutions or even private households as this works well with a variable supply of renewable energy and should be considered as a potential cost-saving for all building operations in combination with smart control.

### 8.3. Limitations

The building investigated in this thesis is a lightweight timber-based building and does not fully represent brick or concrete framed buildings.

The building investigated has very few controls.

The user schedules for the simulations were simplified and do not represent real-world unpredictability.

Only a limited number of *DRL* control algorithms were investigated.

### 8.4. Future Research

In addition to directly tackling the limitations of this thesis outlined above, future research may also be directed towards the following areas:

Smart control for low-insulated, high thermal mass buildings and whether variable price energy is sufficient to result in cost savings, considering their low potential to store energy.

More robust measures for assessing energy savings compared to changes in comfort, which were partially investigated in chapter 6.

Investigating the predictive algorithms' robustness to future weather uncertainty and sudden changes to predicted weather.

Investigating the use of surrogate models instead of the EnergyPlus energy model in to significantly speed up processing.

Benchmarking control algorithms on standardised buildings to create a standardised comparison of various smart control systems.

Investigating physics based *RL* in order to improve learning rates of the agents and provide safe-guarding for extreme, undesirable behaviour.

Investigating multi-objective *RL* to give more control over the weighting of energy saving vs comfort after the training of the model.

*Note: see the Results, chapter 6, for detailed results and the Discussion, chapter 7, for more in-depth detail on the interpretation, real-world impact, potential improvements, and further research.*



## 9. Reflection

The graduation topic combines design informatics and building physics in order to investigate how Artificial Intelligence can be used to operate buildings and how that might impact both current buildings' energy consumption, but also change how we design and operate buildings for the future.

The outcomes of the thesis are satisfactory, but as always there is a desire to go deeper and approach the question from different angles. Equally, every new result and finding raises new and further questions, that desire to be answered.

The main objective of the project was to investigate the relative impact of *RL control algorithms* compared to *building fabric factors* for reducing energy use in buildings. This can lead to a shift in how buildings are operated and designed. Currently, building energy use and assumptions about performance during design are based on reactive control systems, which do not factor in weather forecasting. However, these paradigms, rules-of-thumb and assumptions about performance might change if all buildings are operated based on predictive control using *RL* or other predictive algorithms.

### 9.1. Research and Design

The research focuses on current building operation practices while questioning how these might change in light of future developments in *AI*. Buildings are designed to be used, but a change in how we use and operate them, will eventually also influence how they are designed. This graduation thesis focuses on the design of control systems and how that influences the energy use of buildings.

### 9.2. Ethics

Underneath the main research goal of reducing energy while maintaining user comfort lies a moral and ethical dilemma about what is user comfort for different users and how much should a user be willing to sacrifice in order to save energy. In addition, the current climate crisis or climate emergency puts even more pressure on reducing energy demand. However, for some user groups, a public building or semi-public building may be the only available place to seek refuge from extreme weather conditions.

The software also includes standard assumptions that by its nature of including some, will exclude others as no list can be exhaustive. In *Honeybee for Grasshopper*, the occupancy components assume only adults of a single metabolic rate. Not accounting for gender or age. In the setup of the energy model, a conversion had to be made from this standard adult to something more representative of a school child from the case study.

## 9. Reflection

The general impacts of saving energy and increasing comfort are positive, however, there may be negative side effects depending on the implementation as it may lead occupants to feel that they are being monitored by the building. As discussed in the thesis, the [AI](#) can be implemented in many different ways, some of which do not monitor users and others which do. This in itself is another area of research and some companies like [Ubiquisense](#) are trying to tackle this issue of balancing monitoring and privacy.

### 9.3. Societal Impact

The case study focuses on a publicly owned school building and any improvements made to such public infrastructure can have the potential to benefit the local community in the long term as reduced spending on building operations can be used on other public and social projects.

The project is directly investigating the potential of [AI](#) for saving energy in buildings. This can positively impact the planet by reducing  $\text{CO}_2$ , increase profits by decreasing costs of operation and even benefit people by making buildings more comfortable through predictive control to regulate the indoor environment.

The potential contribution to sustainable development and maintenance of the built environment is large, as a change to a building's control system may be far cheaper than retrofitting. However, the downside is the reluctance of the building owner to change operation systems, and many systems have not been altered since they were commissioned and there may also be a lack of expertise for each individual system setup. In reality, the adoption may be very low, as it still requires specialised knowledge of each building system and its endpoints. While this [RL](#) approach can save on the upfront modelling cost of the building, the [HVAC](#) system itself may need another layer of software on top before it is possible to interact with it depending on each individual system's firmware.

Furthermore, the research findings indicated that current low-performing buildings would also be the ones to benefit the least from such technology since their low levels of insulation and energy control makes them less suitable for predictive control based on weather forecasts. This means further research is required to find more suitable low-cost solutions.

## A. Glossary

This appendix contains description of a few select key terms. The section on Honeybee and EnergyPlus is especially useful for other researchers recreating a similar model.

### A.1. Building Physics

**Form factor** is a ratio of heat loss area (walls, roofs, floors) to the usable floor area. This is commonly used in Passivhaus as a rule of thumb to assess a building's compactness. Standalone residential houses have a high form factor, whereas terraced or multi-story buildings have a low form factor.

**Infiltration rate** is the rate at which uncontrolled ventilation occurs through small gaps in the construction. This is commonly measured either as air changes per hour,  $m^3$  per second,  $m^3$  per second per floor area, or  $m^3$  per second per surface area of the construction.

**Plug load power density** is the amount of unregulated electricity load used in the building per  $m^2$ . This includes personal electronic equipment and depending on the categorisation may also include large equipment such as fridges and freezers.

**Light power density** is the electric load of the lighting installation per  $m^2$  of the building.

**Occupant density** is the number of occupants per  $m^2$ .

**Heat Recovery** is used to express how much heat energy is recovered in a process. In the case of air handling units and indoor ventilation, heat recovery efficiency expresses how much energy is recovered from the outgoing air and transferred to the incoming air.

**Degree-hours** is an expression for degrees times hours. In this thesis, it refers to the magnitude of degrees outside a certain range multiplied by the length of time, in hours.

### A.2. Honeybee and EnergyPlus

Using Honeybee and EnergyPlus requires knowledge of both systems, how things are named and how things are translated between the two systems. Some familiarity with the tools is assumed, but in addition to the standard uses this thesis required knowledge and use of the following items. These were acquired by the author during the research period and are outlined here both to provide a glossary and to provide additional information to other researchers wishing to use the same tools and functions:

**application programming interface (API)** An **API** is a way to interact or communicate with a program. Similar to a graphical user interface (**GUI**), but without a visual/graphical component instead requiring a programming language as a means of information transfer.

**.idf / IDF** This is the main file for any EnergyPlus model. This contains all the information about the simulation model. That includes the physical 3D model of the building, described as a series of nonconvex 2D surface objects. The simulation parameters such as *Runperiod* as well as the simulation output variables and output files.

**EnergyPlus Runtime Language (ERL)** This is the language in which EnergyPlus IDF files are written. Most of the use-cases are creating model objects inside the IDF file. It can also be used for creating conditionals, logic statements, and loops, which is mostly relevant for advanced use of the energy management system. When adding objects to the IDF, often because Honeybee does not support them, they must be written in **ERL**.

**HB Set Identifier** This is a Honeybee component that changes the internal name of a Honeybee object. If this is not used, the translation between Honeybee and EnergyPlus will change the names of all the objects in an unpredictable manner. For example, a zone named Z1 First Floor will be changed to Z1-First-Floor-39052. This is to adhere to the EnergyPlus naming rules, which do not allow for spaces and must ensure that each name is unique, and hence the random number added to the end. Using *HB Set Identifier* forces the use of the original name, but the user must then make sure it adheres the naming rules of EnergyPlus. The new name is therefore 'Z1-First-Floor'. This can be seen in Figure 3.18. Predictable naming becomes important in later stages when referring to specific EnergyPlus objects in Python.

**EnergyPlus file outputs** EnergyPlus outputs multiple files after each simulation, some useful for investigating the potential information hidden in the model. As a rule of thumb, the file types ending in *dd* are dictionaries describing all possible or allowable output options and parameters. Examples are .idd, .rdd, .mdd etc. The .idd is specific to the EnergyPlus installation version and describes the possible EnergyPlus objects that can be created for any simulation model. The .rdd describes the output variables available in your unique model. These variables are also those which are outputted in the *eplusout.csv* results file, but must first be specified in the IDF file as an *output:variable* object. The .edd becomes essential when using the energy management system system as it contains all possible actuators and internal variables. To adjust the output files available, see Figure 3.21 for the additional strings written in EnergyPlus Runtime Language.

**energy management system (EMS)** The **EMS** is primarily used when you want to control the simulation during runtime in a manner more advanced than standard schedules. This can be done by writing control functions into the IDF file using EnergyPlus Runtime Language or by communicating with the runtime of EnergyPlus through the application programming interface.

**Actuators** These are objects that can change and be controlled during a simulation and include setpoints, flow rates, window openings, shading control as well as items that are not normally changed such as heat transfer characteristics of materials, inputs from weather files, etc. Depending on how the IDF is constructed, not all types of actuators are created. The actuators can be created manually as shown in Figures 3.19 and 3.20 as long as they adhere to the .idd standards and are written in EnergyPlus Runtime Language.

**EnergyPlus Calendar** Unlike Excel or Python Datetime, EnergyPlus does not have an advanced internal multi-year calendar. It does contain information about the lengths of months and the weekday for each day in a single year, however, when using it for multi-year simulations the calendar runs into some errors. In previous releases it was not possible to simulate more than one year, but this was solved by an addition to the *RunPeriod* object allowing the simulation period to be repeated X number of times. This has been updated again to now allow for a multi-year specification in the *RunPeriod* selecting start and end day, month and year. In setting the *RunPeriod*, the start weekday has to be specified correctly to match the real calendar date. However, when the simulation continues for the subsequent year, the internal calendar does not continue with the weekdays as a standard Gregorian calendar. Instead it adds 1 to the count of the initial weekday for the calendar year. This works fine for normal years, where each year starts one weekday later than the next, but not for leap-years. For example in 2008 the first day is a Tuesday, and the last day at the end of December is a Wednesday. Hence the first day of 2009 is a Thursday. However, in EnergyPlus if the *RunPeriod* is specified as starting Tuesday 1st January 2008, then when the simulation continues into 2009 it will simply increment the start weekday by 1. This means EnergyPlus would assume the first weekday in 2009 is Wednesday, not Thursday. This is a minor quirk that can impact the schedules governing the building operation when the weekdays suddenly skip. This is also important for the *eplusout.csv* file, as this does not contain year information, only day and month. Therefore, what may be assumed to be 2009 in the output *.csv* file, actually has the weekdays and dates of 2014, which starts with a Wednesday and is a non-leap-year.

**EDD Internal Variables** These include design factors such as occupancy, plug load, and ventilation air flow, but also include variables that shouldn't change such as the floor area. These may have been created in the transfer from Honeybee to EnergyPlus. New internal variable objects can be created based on the *.rdd* to give ems control of these.

## A.3. Machine Learning

**Genetic algorithm** is a type of optimisation algorithm inspired by nature's ability to solve optimisation problems through trial, error and mutation. These algorithms usually start with a large population of functions or variables, assess their performance, select, merge and mutate the best-performing ones and repeat.

**Slime mould optimization** is a type of optimisation algorithm inspired by slime moulds found in nature, because of their ability to solve certain problems better than traditional computer algorithms.

**Transfer learning in Deep Learning (DL)** is when re-using a pre-trained model on a new, but similar problem. This could be a pre-trained building operation model, deployed on a new building. This greatly reduces the required amount of training to adjust to the new problem, as long as the problems are similar enough.

**Supervised learning** is a type of ML requiring pre-labelled data to train a prediction model.

**Unsupervised learning** is a type of ML that uses unlabelled data to analyse and discover patterns in the data.

## A. Glossary

**Model-based RL** is a type of **RL** with 2 functions; a transition function and a reward function. The transition function tries to predict the next state. The reward function tries to predict the next reward.

**Model-free RL** is a type of **RL** with only 1 function, namely the reward function. The system does not predict the next state, but only the next reward. An example of this is Q-learning.

**Neuron** is a connection point in the neural network. This takes in the weighted inputs from the connected neuron and applies an activation function before sending the information to the subsequent neurons.

**Node** - synonym for neuron. See neuron.

**Weight** is a multiplication factor applied to the inputs of neurons in a neural network.

**Bias** is an addition factor applied to the inputs of neurons in a neural network.

**Activation function** is applied in the neuron and provide non-linearity to the network. Common activation functions include **ReLU**, **GELU**, Sigmoid, and tanh.

**Backpropogation** is the process by which the prediction error in the **NN** is propagated backwards to adjust the weights and biases of the neural network to improve the prediction.

**Parameters** are internal variables of the neural network, such as the weights.

**Hyperparameters** are the settings for the learning process, adjusting how the algorithm learns and optimises.

## B. Alternative Calibration Methods

### B.1. Calibration Methods

Different calibration methods were investigated, but not used in the final product as the model achieved sufficient accuracy and effort was instead focused on the [RL](#) parts of the thesis.

#### **Methods for adjusting energy model in order to meet validation criterion:**

Eye-balling inputs until they match well enough. Often used in the beginning or if there is a good understanding of the model's behaviour.

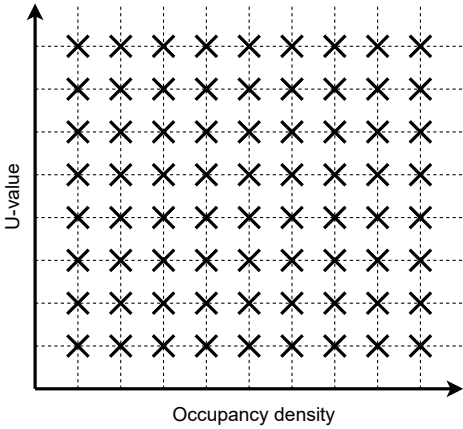
Grid method - exhaustive search. For each input value generate a regular range of values and then run every single combination across all dimensions.

Random method - similar to grid, but only running a smaller random subset of combinations. Then plot them with range for each value on the x axis and accuracy on the y axis, then pick the x-value with the highest accuracy as the correct value for that feature.

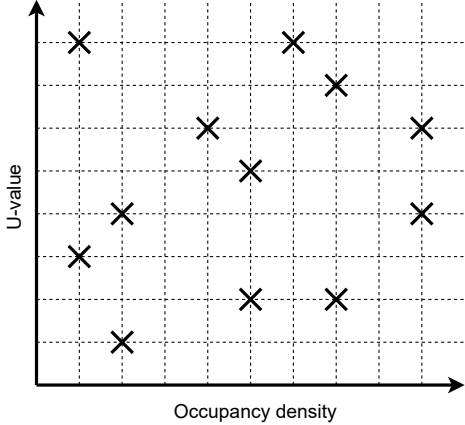
Genetic algorithm to minimize mean square error between the model results and the validation data.

A genetic algorithm was used in the earlier stages of the thesis before changing course to use the [ASHRAE](#) validation metrics with the on-site temperature data. One major flaw in the genetic algorithm approach was that the fitness function only fitted to a single value rather than fitting to a curve, making it non-convergent and the results useless. However, the method, algorithm and results are presented in this chapter.

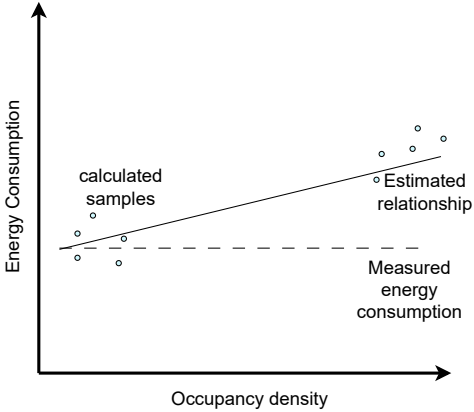
B. Alternative Calibration Methods



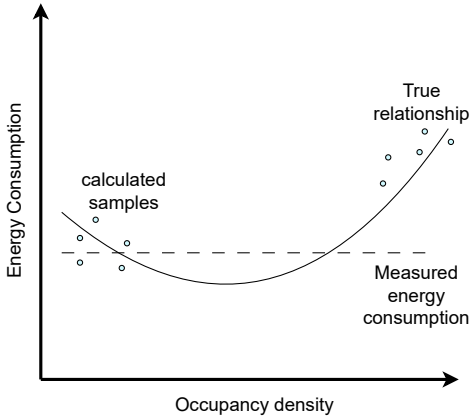
(a) Grid method for exhaustive search. Run all combinations.



(b) Random sampling from grid



(a) Straight line interpretation of samples



(b) Curve line interpretation of samples may be more correct, but requires sufficient number of samples to be revealed.



## B.2. Genetic Algorithm for Calibration

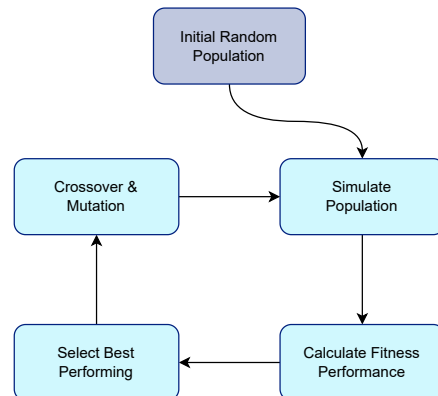


Figure B.3.: Genetic Algorithm concept overview

---

### Algorithm B.1: Genetic Algorithm

---

**Input:** An optimization based on a genetic algorithm using an energy simulation  $S$ , fitness function  $F$ , selection criterion  $P$ , mutation rate  $m$

**Output:** Dataframe of genes and performances

```

1 Generate random population  $R$ 
2 while running do
3   for each gene in population  $R$  do
4     simulate each gene with function  $S$ 
5     calculate fitness performance with  $F$ 
6     append data to dataframe
7     select genes for crossover based on  $P$ 
8     crossover genes to generate new population  $R$ 
9     randomly mutate some genes based on  $m$ 
10 return Dataframe
  
```

---

Fitness objective Target was  $18.2 \text{ kWh/m}^2$  (gross area) for the period

$$\text{Fitness Value} = (\text{Target kWh} - \text{simulated kWh})^2$$

B. Alternative Calibration Methods

Parameter	Range	Unit
People per area	0 - 0.5	$people\ m^{-2}$
Watts per area	0 - 15	$W\ m^{-2}$
Infiltration rate	0 - 0.0006	$m^3\ s^{-1}\ m^{-2}$
Windows u-value	0.4 - 3.0	$W\ m^{-2}\ K^{-1}$
Windows g-value	0.1 - 0.95	<i>coefficient of</i> $W\ m^{-2}$
Walls thermal resistance	0 - 30	$K\ m^2\ W^{-1}$
Thermal mass of int walls (specific heat capacity)	500 - 10,000	$J\ kg^{-1}\ K^{-1}$
Heat recovery efficiency	0 - 0.95	<i>coefficient</i>

Table B.1.: Parameters used for digital twin calibration in the genetic algorithm setup.

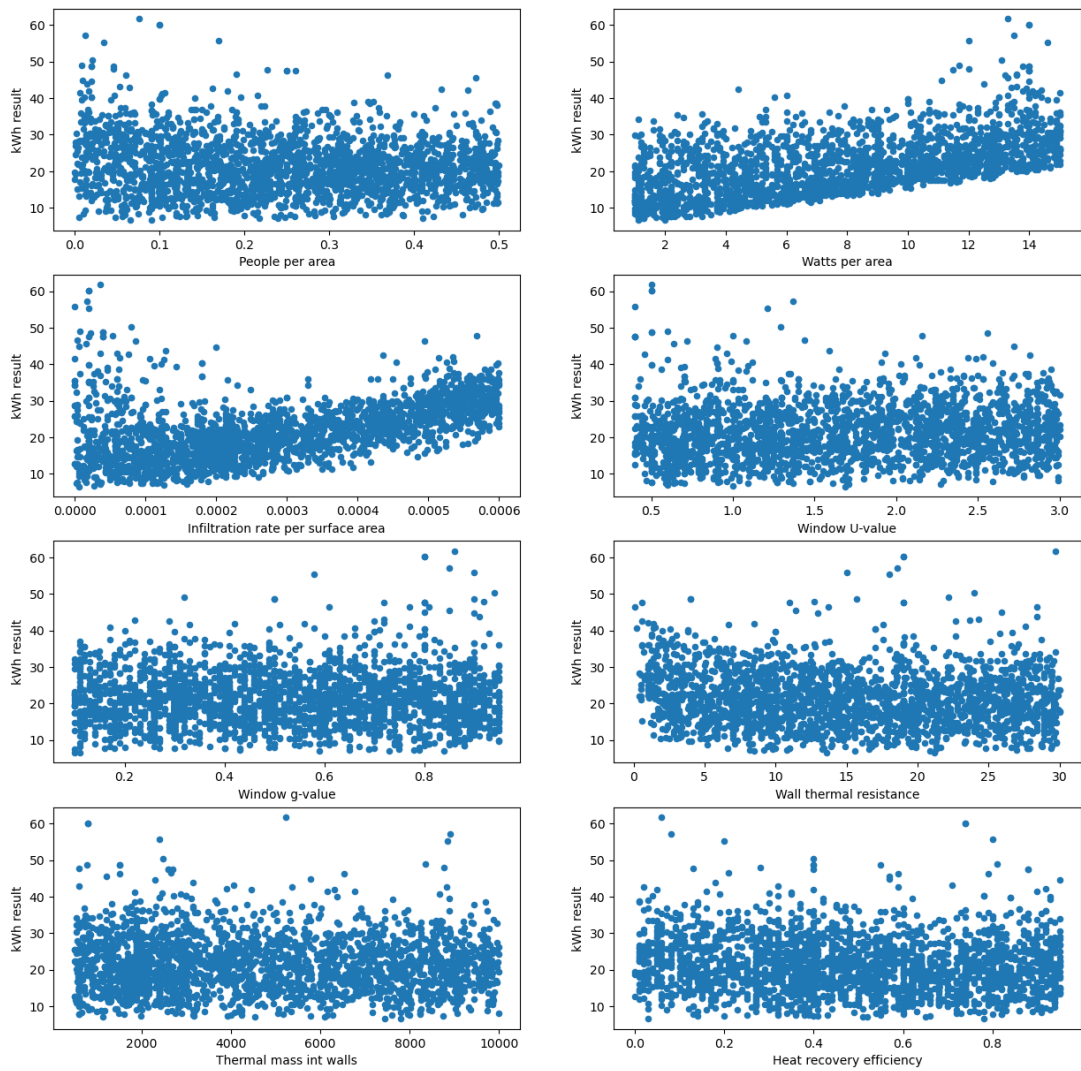


Figure B.4.: Results from multiple runs of the genetic algorithm

# C. Additional Results from Analyses

## C.1. Metrics Performance Scores

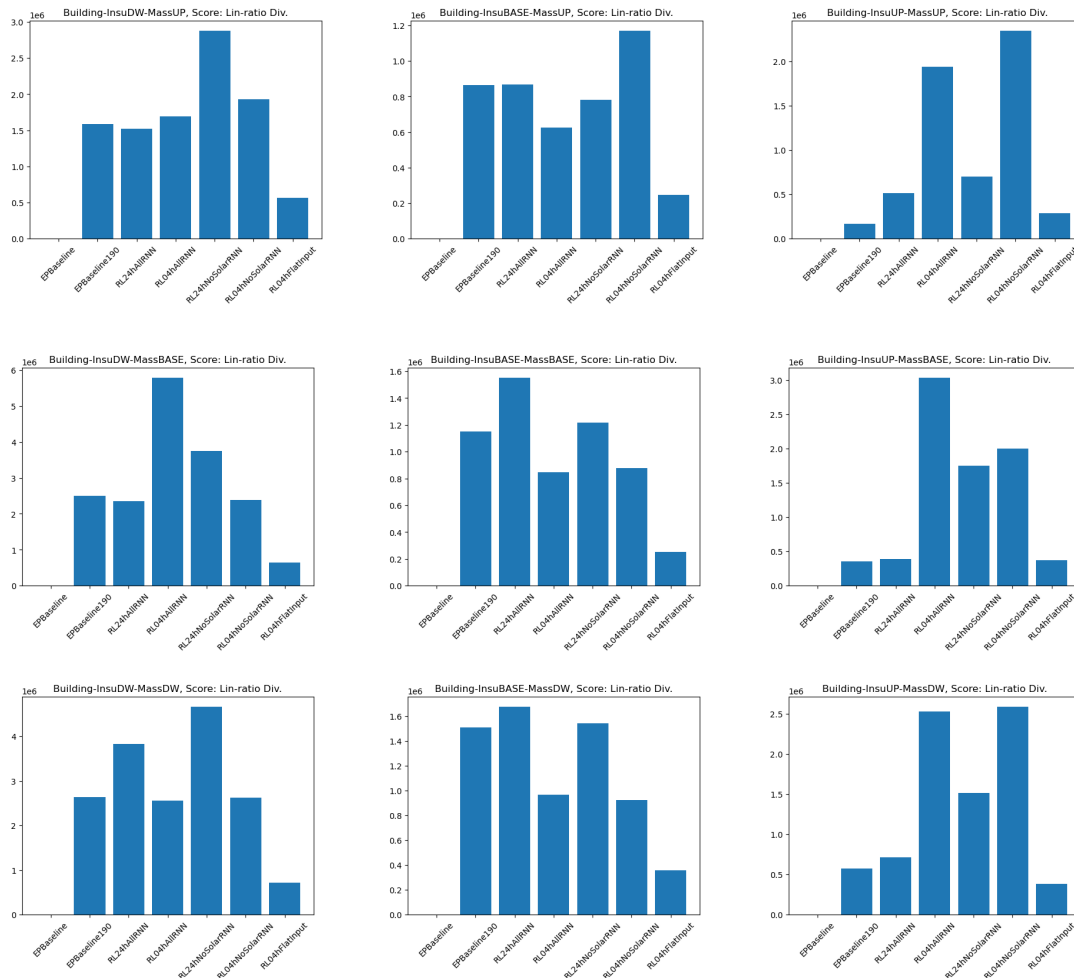


Figure C.1.: Bar chart of the accumulated score of the algorithms using the linear ratio division metric for all 9 building variants.

## C.2. Degree-hours vs Energy use for all building types

C. Additional Results from Analyses

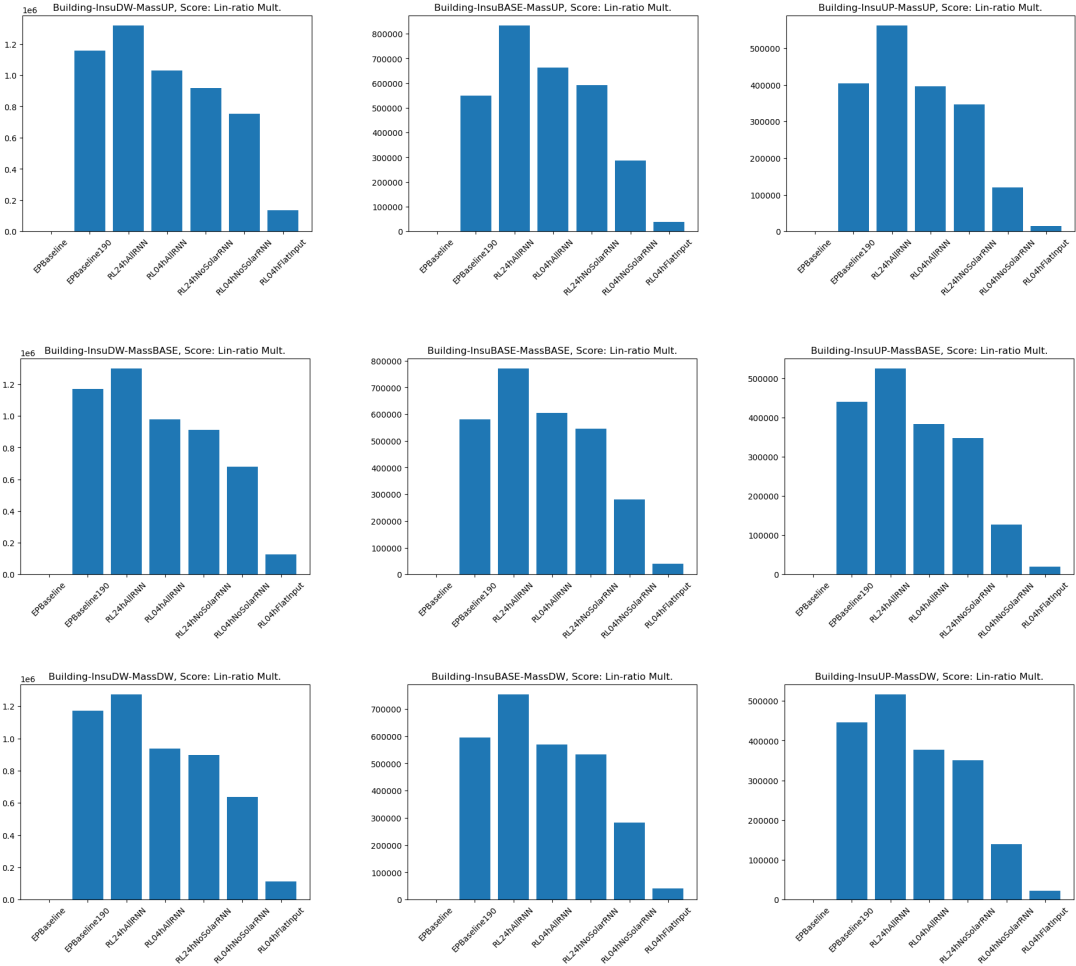


Figure C.2.: Bar chart of the accumulated score of the algorithms using the linear ratio multiplication metric for all 9 building variants.

C.2. Degree-hours vs Energy use for all building types

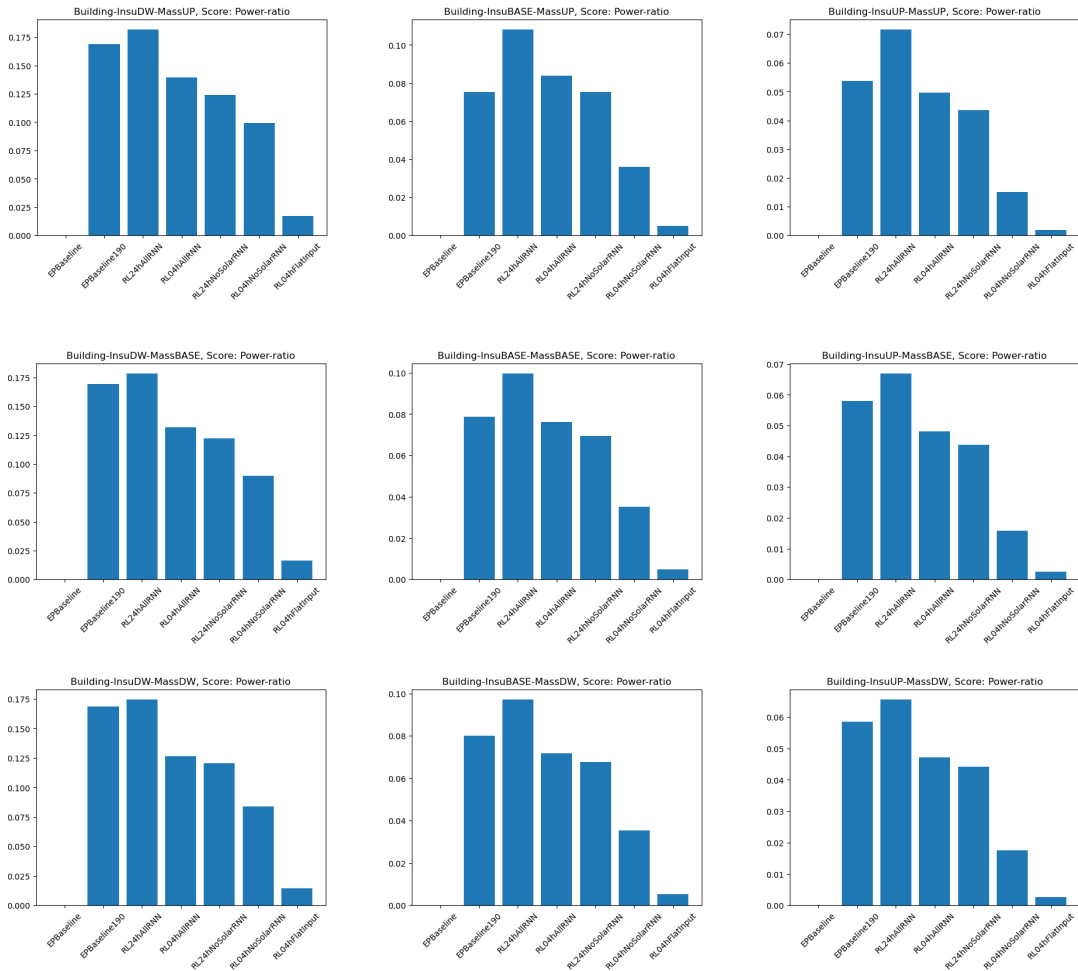


Figure C.3.: Bar chart of the accumulated score of the algorithms using the power ratio metric for all 9 building variants.

C. Additional Results from Analyses

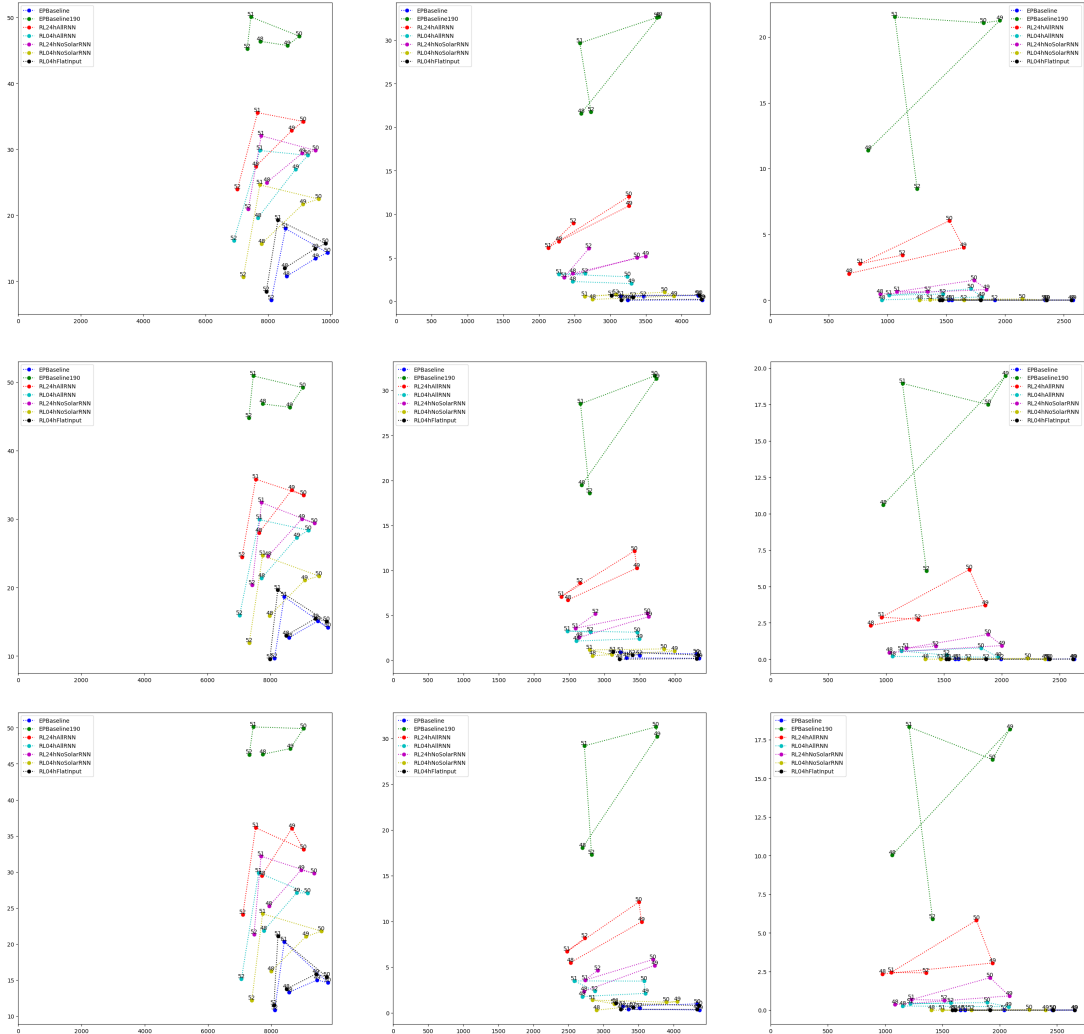


Figure C.4.: Degree-hours plotted against energy use MJ for all 9 buildings for weeks 48 to 52.

# Bibliography

- Arroyo, J., Manna, C., Spiessens, F. and Helsen, L. (2021), 'An Open-AI gym environment for the Building Optimization Testing (BOPTTEST) framework'.
- Arroyo, J., Manna, C., Spiessens, F. and Helsen, L. (2022), 'Reinforced model predictive control (RL-MPC) for building energy management', *Applied Energy* **309**, 118346.
- ASHRAE (2002), 'ASHRAE Guideline 14-2002'.
- Attar, R., Chronis, A., Hanna, S. and Turrin, M. (2016), '2016 Proceedings of the Symposium on Simulation for Architecture and Urban Design'.
- B4B (2023), 'About – Brains4buildings'. Available at <https://brains4buildings.org/about/>.
- Bannister, P., Thomas, P. and Lowndes, P. (2011), 'REPRESENTATION OF HVAC CONTROL IN COMMON SIMULATION PACKAGES'.
- Blanco, J. L., Engel, H., Imhorst, F., Ribeirinho, M. J. and Sjödin, E. (2021), 'Seizing the decarbonization opportunity in construction — McKinsey', <https://www.mckinsey.com/industries/engineering-construction-and-building-materials/our-insights/call-for-action-seizing-the-decarbonization-opportunity-in-construction>.
- Blum, D., Arroyo, J., Huang, S., Drgoňa, J., Jorissen, F., Walnum, H. T., Chen, Y., Benne, K., Vrabie, D., Wetter, M. and Helsen, L. (2021), 'Building optimization testing framework (BOPTTEST) for simulation-based benchmarking of control strategies in buildings', *Journal of Building Performance Simulation* **14**(5), 586–610.
- Bokel, R. (2019), 'Validation and Calibration of Design Builder Temperature Simulations\_v2'.
- Chakrabarty, A., Maddalena, E., Qiao, H. and Laughman, C. (2021), 'Scalable Bayesian optimization for model calibration: Case study on coupled building and HVAC dynamics', *Energy and Buildings* **253**, 111460.
- Chitkara, S., Thamban, A. and Zeiler, W. (1/july/2022), 'Approaches to ML & system-based diagnose and conclusions on the beta design of a software module'.
- Christodoulou, P. (2019), 'Soft Actor-Critic for Discrete Action Settings'.
- Cígler, J., Gyalistras, D., Široky, J., Tiet, V. and Ferkl, L. (2013), Beyond theory: The challenge of implementing model predictive control in buildings, in 'Proceedings of 11th Rehva World Congress, Clima', Vol. 250.
- de Wilde, P. (2014), 'The gap between predicted and measured energy performance of buildings: A framework for investigation', *Automation in Construction* **41**, 40–49.

## Bibliography

- Dr. Wolfgang Feist (2023), 'PHPP Passive House Design Package', [https://passiv.de/former\\_conferences/Passive\\_House\\_E/PHPP.html](https://passiv.de/former_conferences/Passive_House_E/PHPP.html).
- Dulac-Arnold, G., Mankowitz, D. and Hester, T. (2019), 'Challenges of Real-World Reinforcement Learning'.
- Ekici, B., Kazanasmaz, T., Turrin, M., Tasgetiren, M. F. and Sariyildiz, I. S. (2019), 'A Methodology for daylight optimisation of high-rise buildings in the dense urban district using overhang length and glazing type variables with surrogate modelling', *Journal of Physics: Conference Series* **1343**(1), 012133.
- Eubel, C. (2023), 'RL - EmsPy (work in progress...)'. Available at <https://github.com/mechyai/RL-EmsPy>.
- European Commission (2020), 'In focus: Energy efficiency in buildings', [https://commission.europa.eu/news/focus-energy-efficiency-buildings-2020-02-17\\_en](https://commission.europa.eu/news/focus-energy-efficiency-buildings-2020-02-17_en).
- Gao, Y., Ruan, Y., Fang, C. and Yin, S. (2020), 'Deep learning and transfer learning models of energy consumption forecasting for a building with poor information data', *Energy and Buildings* **223**, 110156.
- Goodfellow, I., Bengio, Y. and Courville, A. (2016), *Deep Learning*, Adaptive Computation and Machine Learning, The MIT Press, Cambridge, Massachusetts.
- Guo, J., Liu, R., Xia, T. and Pouramini, S. (2021), 'Energy model calibration in an office building by an optimization-based method', *Energy Reports* **7**, 4397–4411.
- Harmer, J., Gisslén, L., del Val, J., Holst, H., Bergdahl, J., Olsson, T., Sjöo, K. and Nordin, M. (2018), 'Imitation Learning with Concurrent Actions in 3D Games'.
- Hasankhani, A., Tang, Y., VanZwieten, J. and Sultan, C. (2021), Comparison of Deep Reinforcement Learning and Model Predictive Control for Real-Time Depth Optimization of a Lifting Surface Controlled Ocean Current Turbine, in '2021 IEEE Conference on Control Technology and Applications (CCTA)', IEEE, San Diego, CA, USA, pp. 301–308.
- HEC (2019), 'Forty Percent of "AI Startups" in Europe Don't Actually Use AI', <https://www.hec.edu/en/news-room/forty-percent-ai-startups-europe-don-t-actually-use-ai>.
- Hendrycks, D. and Gimpel, K. (2020), 'Gaussian Error Linear Units (GELUs)'.
- Hesketh, R. (2022), 'Energy Efficiency Ratings Aren't Actually Predicting Energy Efficiency', *Bloomberg.com*.
- Ji, Y. and Xu, P. (2015), 'A bottom-up and procedural calibration method for building energy simulation models based on hourly electricity submetering data', *Energy* **93**, 2337–2350.
- Kinsley, H. and Kukiela, D. (2020), *Neural Networks from Scratch in Python*.
- Kon, P. (2022), 'Is GELU, the ReLU successor? – Towards AI'.
- Lamberti, G. (2020), Thermal comfort in the built environment: Current solutions and future expectations, in '2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)', pp. 1–6.



- Lamberti, G., Fantozzi, F. and Salvadori, G. (2020), Thermal comfort in educational buildings: Future directions regarding the impact of environmental conditions on students' health and performance, in '2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC / I&CPS Europe)', pp. 1–6.
- LETI (2020), 'LETI Climate Emergency Design Guide'.
- Li, Y., Bae, Y. and Im, P. (2021), Surrogate Model of Flexible Research Platform EnergyPlus Models to Enable Sensitivity Analysis, Technical Report ORNL/LTR-2021/1923, 1817464.
- Lin, Y., McPhee, J. and Azad, N. L. (2021), 'Comparison of Deep Reinforcement Learning and Model Predictive Control for Adaptive Cruise Control', *IEEE Transactions on Intelligent Vehicles* 6(2), 221–231.
- Liu, S. and Henze, G. P. (2006), 'Experimental analysis of simulated reinforcement learning control for active and passive building thermal storage inventory: Part 2: Results and analysis', *Energy and Buildings* 38(2), 148–161.
- Lotus Labs (2020), 'Clarifying AI, Machine Learning, Deep Learning, Data Science with Venn Diagrams'.
- Luo, J. T., Joybari, M. M., Panchabikesan, K., Sun, Y., Haghghat, F., Moreau, A. and Robichaud, M. (2020), 'Performance of a self-learning predictive controller for peak shifting in a building integrated with energy storage', *Sustainable Cities and Society* 60, 102285.
- Ma, J., Qin, J., Salsbury, T. and Xu, P. (2012), 'Demand reduction in building energy systems based on economic model predictive control', *Chemical Engineering Science* 67(1), 92–100.
- Martin, C. H., Peng, T. S. and Mahoney, M. W. (2021), 'Predicting trends in the quality of state-of-the-art neural networks without access to training or testing data', *Nature Communications* 12(1), 4122.
- Martinez-Viol, V., Urbano, E. M., Delgado-Prieto, M. and Romeral, L. (2022), 'Automatic model calibration for coupled HVAC and building dynamics using Modelica and Bayesian optimization', *Building and Environment* 226, 109693.
- McLeod, D. R., Jaggs, M., Cheeseman, B., Tilford, A. and Mead, K. (n.d.), 'Passivhaus primer: Airtightness Guide - Airtightness and air pressure testing in accordance with the Passivhaus standard - A guide for the design team and contractors'.
- Merzkirch, A., Hoos, T., Maas, S., Scholzen, F. and Waldmann, D. (2014), 'Accuracy of energy performance certificates - Comparison of the calculated and measured final energy demand for 230 residential buildings in Luxembourg', *Bauphysik* 36, 40–43.
- Mitchell, T. M. (1997), *Machine Learning*, McGraw-Hill Series in Computer Science, McGraw-Hill, New York.
- Mozer, M. C. (1998), 'The Neural Network House: An Environment that Adapts to its Inhabitants', p. 5.
- Muehleisen, R. T. and Bergerson, J. (2016), 'Bayesian Calibration - What, Why And How', p. 10.

## Bibliography

- Nguyen, T., Raghu, M. and Kornblith, S. (2021), 'Do Wide and Deep Networks Learn the Same Things? Uncovering How Neural Network Representations Vary with Width and Depth'.
- Olah, C. (2015), 'Understanding LSTM Networks'.
- OneBuilding (2023), 'Climate Weather Files'. Available at [https://climate.onebuilding.org/WMO\\_Region\\_6\\_Europe/GBR\\_United\\_Kingdom/index.html](https://climate.onebuilding.org/WMO_Region_6_Europe/GBR_United_Kingdom/index.html).
- OrbitalStack (2023), 'Orbital Stack AI', OrbitalStack.
- Pallonetto, F., Finn, D., Milano, F. and Mangina, E. (2019), 'SimApi, a smartgrid co-simulation software platform for benchmarking building control algorithms', *SoftwareX* **9**, 271–281.
- Pascal Poupart (2018), 'CS885 Lecture 12: Deep Recurrent Q-Networks'.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S. (2019), PyTorch: An imperative style, high-performance deep learning library, *in* 'Advances in Neural Information Processing Systems 32', Curran Associates, Inc., pp. 8024–8035.
- Perolat, J., Vyllder, B. D., Hennes, D., Tarassov, E., Strub, F. and Tuyls, K. (2022), 'Mastering Stratego, the classic game of imperfect information', <https://www.deepmind.com/blog/mastering-stratego-the-classic-game-of-imperfect-information>.
- PHPP 9 (*Passive House Planning Package*) (2016), Passivhaus Institute.
- Ramos Ruiz, G., Fernández Bandera, C., Gómez-Acebo Temes, T. and Sánchez-Ostiz Gutierrez, A. (2016), 'Genetic algorithm for building envelope calibration', *Applied Energy* **168**, 691–705.
- Russell, S. J., Norvig, P. and Davis, E. (2010), *Artificial Intelligence: A Modern Approach*, Prentice Hall Series in Artificial Intelligence, 3rd ed edn, Prentice Hall, Upper Saddle River.
- Satake, A., Ikegami, H. and Mitani, Y. (2016), 'Energy-saving Operation and Optimization of Thermal Comfort in Thermal Radiative Cooling/Heating System', *Energy Procedia* **100**, 452–458.
- Scharnhorst, P., Schubnel, B., Fernández Bandera, C., Salom, J., Taddeo, P., Boegli, M., Gorecki, T., Stauffer, Y., Peppas, A. and Politi, C. (2021), 'Energym: A Building Model Library for Controller Benchmarking', *Applied Sciences* **11**(8), 3518.
- Sculley, D. (2022), 'D. Sculley — Technical Debt, Trade-offs, and Kaggle'.
- Simonini, T. and Sanseviero, O. (2022), 'The hugging face deep reinforcement learning class'.
- Široký, J., Oldewurtel, F., Cigler, J. and Prívvara, S. (2011), 'Experimental analysis of model predictive control for an energy efficient building heating system', *Applied Energy* **88**(9), 3079–3087.
- Suryana, L. (2020), 'Which one is better: Reinforcement Learning or Model Predictive Control? Inverted Pendulum — Case\*'

- Sutton, R. S. (1988), 'Learning to predict by the methods of temporal differences', *Machine Learning* 3(1), 9–44.
- Sutton, R. S. and Barto, A. G. (2018), *Reinforcement Learning: An Introduction*, Adaptive Computation and Machine Learning Series, second edition edn, The MIT Press, Cambridge, Massachusetts.
- The Economist (2022), 'Another game falls to an AI player', *The Economist* .
- Thieblemont, H., Haghghat, F., Moreau, A. and Lacroix, G. (2018), 'Control of electrically heated floor for building load management: A simplified self-learning predictive control approach', *Energy and Buildings* 172, 442–458.
- Thieblemont, H., Haghghat, F., Ooka, R. and Moreau, A. (2017), 'Predictive control strategies based on weather forecast in buildings with energy storage system: A review of the state-of-the art', *Energy and Buildings* 153, 485–500.
- van der Linden, A. C., Erdtsieck, P., Kuijpers-van Gaalen, I. M. and Zeegers, A. (2013), *Building Physics*, 1st ed edn, ThiemeMeulenhoff, Amersfoort.
- van Dronkelaar, C., Dowson, M., Spataru, C. and Mumovic, D. (2016), 'A Review of the Regulatory Energy Performance Gap and Its Underlying Causes in Non-domestic Buildings', *Frontiers in Mechanical Engineering* 1.
- Vázquez-Canteli, J. R. and Nagy, Z. (2019), 'Reinforcement learning for demand response: A review of algorithms and modeling techniques', *Applied Energy* 235, 1072–1089.
- Wang, L. (2009), *Model Predictive Control System Design and Implementation Using MATLAB®*, Advances in Industrial Control, Springer London, London.
- Williams, T. (2022), 'Reinforcement Learning Vs. Deep Reinforcement Learning', <https://www.techopedia.com/reinforcement-learning-vs-deep-reinforcement-learning-whats-the-difference/2/34039>.
- Yanatma, S. (2022), 'Ranked: The European countries with the best and worst home insulation', <https://www.euronews.com/green/2022/12/09/europes-energy-crisis-in-data-which-countries-have-the-best-and-worst-insulated-homes>.
- Yang, L., Nagy, Z., Goffin, P. and Schlueter, A. (2015), 'Reinforcement learning for optimal control of low exergy buildings', *Applied Energy* 156, 577–586.
- Yang, Z. and Becerik-Gerber, B. (2015), 'A model calibration framework for simultaneous multi-level building energy simulation', *Applied Energy* 149, 415–431.
- Yiu, T. (2021), 'The Curse of Dimensionality', <https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e>.
- Yu, Z. J., Huang, G., Haghghat, F., Li, H. and Zhang, G. (2015), 'Control strategies for integration of thermal energy storage into buildings: State-of-the-art review', *Energy and Buildings* 106, 203–215.
- Zagoruyko, S. and Komodakis, N. (2017), 'Wide Residual Networks'.
- Zhang, Y. (2010), *New Advances in Machine Learning*, BoD – Books on Demand.

## **Colophon**

This document was typeset using  $\text{\LaTeX}$ , using the KOMA-Script class `scrbook`. The main font is Palatino.

