CARNEGIE MELLON UNIVERSITY

Machine Learning for Multi-Discipline Parametric Analysis in Architectural Practice

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR A DOCTOR OF DESIGN IN ARCHITECURE

THESIS COMMITTEE:

PROF RAMESH KRISHNAMURTI. PHD (WATERLOO). PROFESSOR, CARNEGIE MELLON UNIVERSITY.

DR JOHN HAYMAKER. PHD (STANFORD). DIRECTOR OF RESEARCH, PERKINS&WILL ARCHITECTS.

PROF DANIEL CARDOSO LLACH. PHD (MIT). PROFESSOR, CARNEGIE MELLON UNIVERSITY.

PROF AARTI SINGH. PHD (WISCONSIN-MADISON). PROFESSOR, CARNEGIE MELLON UNIVERSITY.

Submitted by Victor Okhoya
11-19-2019

Acknowledgements

I dedicate this work to my mother and father, Lois and Ainea, and to all my sisters Naututu, Naliaka, Nelima, Nanyolo and Nafu. I thank the School of Architecture at Carnegie Mellon for the opportunity and support through this process. In particular, I thank Prof Ramesh Krishnamurti for his guidance and for chairing my thesis committee. I also thank the other members of the committee Dr John Haymaker, Prof Daniel Cardoso Llach and Prof Aarti Singh. I particularly thank John Haymaker for being a great educator and for mentoring me both as a student and as a co-worker. In addition, I thank Susan Gushe AIBC and Perkins&Will Architects as a whole, for their support through the process.

I acknowledge and credit the contributions of all my research colleagues from the Perkins&Will Design Process Lab. In particular Dr Cheney Chen, Dr Marcelo Bernal, Tyrone Marshall AIA and Dr Mohamed Imam have all contributed to the data, the reference projects and the simulation processes discoursed upon in this work.

I would also like to mention the great teachers that have provided guidance over the years. Dr John Haymaker, Prof Harry Lewis, Prof Harvey Brown, Patrick Mwaniki, Raphael Ghortaza, Silvano Borusso and the women from my primary school days: Nzoi, Sisenda, Njaga, Pobunderwalla, Chakava, Walobwa and Kosgei.

Lastly, I thank Kelly for her patience.

Abstract

Parametric analysis performs building performance analysis by simulating multiple design alternatives compared to the single design alternative analysis of conventional methods. It is an emergent, data-driven approach to architectural design. It has the benefits of enabling the optimization of design spaces, reducing uncertainty in design decision making, and establishing the most sensitive input parameters of the design space. However, it also faces data analytical challenges in practical application. These challenges include the size of design spaces, the speed and accuracy of simulation, and the presence of complex design space conditions. This study investigates machine learning algorithms as a design space reduction strategy for overcoming these data analytical challenges and thereby facilitating the implementation of parametric analysis in architectural practice. It focuses primarily on residential projects designed by Perkins&Will Architects in British Columbia. The study asks if machine learning algorithms can be effective at design space reduction for multi-discipline parametric analysis. In addition, it asks which of the algorithms we study - artificial neural networks, support vector machines and random forests - is the most effective? Are machine learning algorithms robust under complex design space conditions? Further, what is the influence on algorithm performance of the impact factors sample size, sensitivity analysis, feature selection and hyperparameter tuning? The methodology for addressing these questions involves the use of real world case study projects, dynamic simulation of the case study design spaces, and machine learning experiments on the simulated design spaces. A methodological framework referred to as Design Space Construction is used for defining the design spaces, integrating the multi-disciplinary simulation infrastructure and evaluating the analysis outcomes. Machine Learning experiments are performed to assess the effectiveness of the algorithms and the influence of performance impact factors (sample size, sensitivity analysis, feature selection and hyperparameter tuning). The study finds that machine learning algorithms can indeed be effective for multi-discipline parametric analysis. In particular, artificial neural networks are the most effective algorithm. Likewise, hyperparameter tuning is the most influential impact factor affecting algorithm performance. Lastly, machine learning algorithms can be robust under complex design space conditions.

Keywords

Parametric Analysis, Machine Learning, Design Space

Published Papers

Bernal, M; Marshall, T; Okhoya, V; Chen, C; Haymaker, J (2019). Parametric Analysis versus Intuition. *eCAADe 37, SIGraDI 23.* 103-110.

Haymaker, J; Bernal, M; Marshall, T; Okhoya, V; Szilasi, A; Rezaee, R; Chen, C; Salveson, A; Brechtel, J; Deckinga, L; Hasan, H; Ewing, P; Welle, B. (2018). Design Space Construction: A framework to support collaborative, parametric decision making. *ITcon, 23*, 157-178.

Okhoya, V., Bernal, M., Chen, C., Marshall, T., & Haymaker , J. (2018). Constructing Design Spaces . *Perkins&Will Research Journal*.

Okhoya, V., Bernal, M., Chen, C., Marshall, T., & Haymaker , J. (2017). Design of Experiements for Multi-Discipline Parametric Analysis in Architecture. *(Unpublished).*

Contents

Acknowledgements ..... **vi**

Abstract ..... **vii**

Keywords ..... **vii**

Published Papers ..... **viii**

List of Figures **13**

List of Tables **18**

Abbreviations **21**

Glossary **23**

Chapter 1 Introduction **25**

1.1 Overview 25

1.2 Research Background 27

1.3 Outline 29

Chapter 2 Problem Formulation **31**

2.1 Limitations of Conventional Analysis 31

2.1.1 Project Example: Balcony Tower 34

2.1.2 Project Example: Hotel Residence 36

2.2 Importance of Parametric Analysis 37

2.2.1 Multi-Discipline Optimization 38

2.2.2 Uncertainty Analysis 39

2.2.3 Sensitivity Analysis 40

2.3 Challenges of Implementing Parametric Analysis 41

2.3.1 Design Space Size 43

2.3.2 Speed 47

2.3.3 Accuracy 48

2.3.4 Sampling Ratio 51

2.3.5 Design Space Complexity 56

2.4 Research Questions 58

Chapter 3 Literature Review **60**

3.1 Introduction 60

3.2 Machine Learning for Design Space Reduction 62

3.3 Machine Learning Review 63

Chapter 4 Machine Learning Background **69**

4.1 Types of Machine Learning 69
4.1.1 Supervised Learning 69
4.1.2 Unsupervised Learning 70
4.2 Supervised Machine Learning Algorithms 70
4.2.1 Support Vector Machines 70
4.2.2 Decision Trees 72
4.2.3 Artificial Neural Networks 75
4.3 Impact Factors 77
4.3.1 Sample Size 79
4.3.2 Sensitivity Analysis 79
4.3.3 Feature Selection 79
4.3.4 Hyperparameter Tuning 80
Chapter 5 Methodology 81
5.1 Reference Projects 81
5.2 Design Space Construction 83
5.2.1 The DSC Process 84
5.2.2 The Value Function 86
5.3 Simulation Methodology 87
5.3.1 Tools 87
5.3.2 Energy Analysis 88
5.3.3 Daylight Analysis 89
5.4 Experimental Methodology 91
5.4.1 Tools 91
5.4.2 Impact Studies 92
5.4.3 Evaluation Methods 93
5.5 General Experimental Method 95
5.5.1 Data Description 96
5.5.2 Design Space Definition 96
5.5.3 Calculate the Value Function 97
5.5.4 Algorithm Default Values 97
5.5.5 Impact of Sample Size 100
5.5.6 Impact of Sensitivity Analysis 101
5.5.7 Impact of Feature Selection 102
5.5.8 Impact of Hyperparameter Tuning 103
Chapter 6 Residential Tower Experiment 105
6.1 Introduction 105
6.2 DSC Process 106
6.2.1 Objective Definition 107
6.2.2 Alternative Generation 107
6.2.3 Impact Analysis 108
6.2.4 Value Assessment 109

6.2.5 Discussion 110
6.3 Method 111
6.3.1 Data Description 111
6.3.2 Design Space Definition 111
6.4 Results 112
6.4.1 Sample Size 112
6.4.2 Sensitivity Analysis 114
6.4.3 Feature Selection 117
6.4.4 Hyperparameter Tuning 118
6.5 Discussion 121
Chapter 7 Mid-Rise Towers Experiment 125
7.1 Introduction 125
7.2 DSC Process 126
7.2.1 Objective Definition 126
7.2.2 Alternative Generation 127
7.2.3 Impact Analysis 128
7.2.4 Value Assessment 130
7.2.5 Discussion 131
7.3 Method 133
7.3.1 Data Description 134
7.3.2 Design Space Definition 134
7.4 Results 134
7.4.1 Sample Size 134
7.4.2 Sensitivity Analysis 137
7.4.3 Feature Selection 140
7.4.4 Hyperparameter Tuning 141
7.5 Discussion 144
Chapter 8 Students' Residence Experiment 147
8.1 Introduction 147
8.2 DSC Process 149
8.2.1 Objective Definition 149
8.2.2 Alternative Generation 150
8.2.3 Impact Analysis 150
8.2.4 Value Assessment 151
8.2.5 Discussion 152
8.3 Method 152
8.3.1 Data Description 153
8.3.2 Design Space Definition 154
8.4 Result 154
8.4.1 Sample Size 154
8.4.2 Sensitivity Analysis 156

8.4.3 Feature Selection....159
8.4.4 Hyperparameter Tuning....160
8.5 Discussion....164
Chapter 9 Findings and Conclusion....167
9.1 Summary of Findings....167
9.1.1 Machine Learning Effectiveness....167
9.1.2 Best Performing Algorithm....168
9.1.3 Impact Factors....169
9.1.4 Design Space Complexity....171
9.2 Verification of Findings....173
9.3 Impact of Machine Learning on Design Space Reduction....176
9.4 Conclusion....178
9.5 Future Work....179
References....183

List of Figures

Figure 1:1 Diagram Summary of Chapter One. 25
Figure 1:2 Diagram of Research Background. 27
Figure 2:1 Diagram Summary of Chapter Two. 31
Figure 2:2 (Perkin&Will, 2019) Traditional Environmental Analysis diagram. 32
Figure 2:3 Conventional building performance analysis process. 32
Figure 2:4 (Perkins&Will, 2019) Balcony Tower project. 33
Figure 2:5 (Perkins&Will, 2019) Hotel Residence project. 34
Figure 2:6 (City of Surrey, 2019) Part 3 Step Code Targets. 34
Figure 2:7 (Perkins&Will, 2019) Balcony Tower, thermal bridging optimization PCP. 35
Figure 2:8 (Perkins&Will, 2019) Balcony Tower, HRV system optimization PCP. 36
Figure 2:9 (Perkins&Will, 2019) Balcony Tower, optimization of energy and daylight PCP. 36
Figure 2:10 (Perkins&Will, 2019) Hotel Residence, tornado graph showing sensitivity analysis. 37
Figure 2:11 (Perkins&Will, 2019) Sensitivity Analysis for the Mid-Rise Towers project. 41
Figure 2:12 Parametric Analysis process. 41
Figure 2:13 Design Space Reduction process. 42
Figure 2:14 The Parametric Analysis process incorporating Machine Learning. 43
Figure 2:15 (Perkins&Will, 2019) Residential Tower project. 44
Figure 2:16 (Perkins&Will, 2019) Mid-Rise Towers project. 45
Figure 2:17 (Perkins&Will, 2019) Students' Residence project. 45
Figure 2:18 (Perkins&Will, 2019) Rapid rate of design changes on three reference projects. 47
Figure 2:19 Average Sample Optimum accuracy. 50
Figure 2:20 Impact of Sampling Ratio on Energy. 51
Figure 2:21 Impact of Sampling Ratio for Illuminance. 52
Figure 2:22 (Perkins&Will, 2019) A design sketch of the Boys' School project. 53
Figure 2:23 (Perkins&Will, 2019) Panel combinations for the façade of the Boys' School project. 54
Figure 2:24 Boys' School design space reduction by designer judgement. 54
Figure 2:25 (Perkins&Will, 2019) Boys' School typical floor zoning. 54
Figure 2:26 Boys' School design space reduction through zone-by-zone analysis. 55
Figure 2:26 Boys' School design of experiments. 55
Figure 2:27 Boys' School Prediction Profiler. 55
Figure 2:29 Boys' School final design space definition of 288 instances. 56
Figure 2:30 Residential Tower, slice of typical level. 56
Figure 2:31 Hotel Residence, slices of typical levels. 57
Figure 2:32 Sources of Design Space Complexity. 58
Figure 3:1 Diagram Summary of Chapter Three. 60
Figure 3:2 Combinatorial Explosion. Each parameter has a value range of 10. 61
Figure 3:3 Approaches to solving high cardinality design spaces. 62
Figure 3:4 Liu et al. (2018) Luminance maps predicted by ANN algorithms. 63

Figure 3:5 (Wikipedia, 2019) Pareto Front 67
Figure 3:6 (Fang, 2017) Pareto front optimization for energy and daylighting analysis. 68
Figure 4:1 Diagram Summary of Chapter Four 69
Figure 4:2 Demonstration of the SVM concept. 71
Figure 4:3 Demonstration of the kernel concept. 71
Figure 4:4 Demonstration of the Support Vector Regression concept 72
Figure 4:5 A decision tree for the Boolean statement x1 ∧ x2. 72
Figure 4:6 (Machine Learning, 2019) Bias-variance trade-off 73
Figure 4:7 (Datacamp, 2019) Ensemble learning. 74
Figure 4:8 (Datacamp, 2019) Bagging. 74
Figure 4:9 (Datacamp, 2019) Random Forest. 75
Figure 4:10 (Towards Data Science, 2019) Perceptron. 76
Figure 4:11 (Towards Data Science, 2019) Multi-layer feed-forward neural network 76
Figure 4:12 Diagram of impact factor study. 77
Figure 4:13 Sensitivity analysis report from JMP. 78
Figure 5:1 Diagram Summary of Chapter Five 81
Figure 5:2 (Perkins&Will, 2019) Residential Tower, Vancouver. 82
Figure 5:3 (Perkins&Will, 2019) Mid-Rise Towers, Vancouver. 82
Figure 5:4 (Perkins&Will, 2019) Students' Residence, Vancouver. 83
Figure 5:5 Design Space Construction (DSC) framework. 83
Figure 5:6 DSC Alternative Generation. 84
Figure 5:7 DSC Impact Analysis. 85
Figure 5:8 DSC Value Assessment 85
Figure 5:9 Simulation tools. 87
Figure 5:10 (Perkins&Will, 2019) DSC Script in Rhinoceros + Grasshopper. 88
Figure 5:11 Energy analysis process. 89
Figure 5:12 Daylight analysis process. 89
Figure 5:13 (Perkins&Will, 2019) Daylight analysis graphic output. 90
Figure 5:14 Experimental process. 91
Figure 5:15 The Weka Experimenter interface. 92
Figure 5:16 The Weka Experimenter experiment results 92
Figure 5:17 Evaluation of sample size results 93
Figure 5:18 Evaluation of sensitivity analysis results. 93
Figure 5:19 Evaluation of feature selection results. 93
Figure 5:20 Evaluation of hyperparameter tuning results 94
Figure 5:21 Evaluation of impact factors for one algorithm on one experiment. 94
Figure 5:22 Evaluation of impact factors for one algorithm across experiments. 94
Figure 5:23 (Perkin&Will, 2019) Sprout Space example in Vancouver, British Columbia. 96
Figure 5:18 Weka Random Forest defaults. 98
Figure 5:19 Weka Multi-Layer Perceptron defaults. 99
Figure 5:20 Weka SMOreg defaults. 100

Figure 6:1 Diagram Summary of Chapter Six. 105
Figure 6:2 (Perkins&Will, 2019) RT design impression. Inset: proposed tower shift. 106
Figure 6:3 DSC process for the Residential Tower project. 107
Figure 6:4 (Perkins&Will, 2019) Residential Tower ranges of input variables. 108
Figure 6:5 (Perkins&Will, 2019) Residential Tower zoning on typical level. 109
Figure 6:6 Residential Tower Prediction Profiler from JMP. 109
Figure 6:7 Residential Tower partial PCP. 109
Figure 6:8 Residential Tower sensitivity analysis. 110
Figure 6:10 Residential Tower impact of Sample Size on the RAE of Site Energy. 112
Figure 6:11 Residential Tower impact of sample size on the RAE of Illuminance. 113
Figure 6:12 Residential Tower impact of Sample Size on the RAE of the Value Function. 113
Figure 6:13 Residential Tower Sensitivity Analysis, Energy. 114
Figure 6:14 Residential Tower Sensitivity Analysis, Illuminance. 115
Figure 6:15 Residential Tower Sensitivity Analysis, Value Function. 115
Figure 6:16 Residential Tower impact of Sensitivity Analysis on the RAE of Energy. 115
Figure 6:17 Residential Tower impact of Sensitivity Analysis on the RAE of Illuminance. 116
Figure 6:18 Residential Tower impact of Sensitivity Analysis on the RAE of the Value Function. 116
Figure 6:19 Residential Tower impact of Feature Selection. 117
Figure 6:20 Residential Tower impact of Hyperparameter Tuning on MLP. 119
Figure 6:21 Residential Tower impact of Hyperparameter Tuning on SMOReg. 120
Figure 6:22 Residential Tower impact of Hyperparameter Tuning on RF. 121
Figure 6:23 Residential Tower effect of impact factors on MLP. 122
Figure 6:24 Residential Tower effect of impact factors on SMOReg. 123
Figure 6:25 Residential Tower effect of impact factors on RF. 123
Figure 7:1 Diagram Summary of Chapter Seven. 125
Figure 7:2 (Perkins&Will, 2019) Mid-Rise Towers project. 126
Figure 7:3 (Perkins&Will, 2019) Mid-Rise Towers project slices. 127
Figure 7:4 (Perkins&Will, 2019) Mid-Rise Towers zoning approach 127
Figure 7:5 (Perkins&Will, 2019) Mid-Rise Towers whole building summary report 128
Figure 7:6 (Perkins&Will, 2019) Mid-Rise Towers performance matrix report 129
Figure 7:7 (Perkins&Will, 2019) Mid-Rise Towers detailed breakdown report 129
Figure 7:8 Mid-Rise Towers identification key. 130
Figure 7:9 (Perkins&Will, 2019) Mid-Rise Towers parallel coordinates plot. 130
Figure 7:10 (Perkins&Will, 2019) Mid-Rise Towers sensitivity analysis 131
Figure 7:11 (Perkins&Will, 2019) Mid-Rise Towers best performing strategies. 132
Figure 7:12 (Perkins&Will, 2019) Mid-Rise Towers optimum WWR ratios 132
Figure 7:13 (Perkins&Will, 2019) MRT best performing alternatives for Tower B typical levels 133
Figure 7:14 Mid-Rise Towers impact of Sample Size on the RAE of Site Energy. 135
Figure 7:15 Mid-Rise Towers impact of Sample Size on the RAE of Illuminance. 136
Figure 7:16 Mid-Rise Towers impact of Sample Size on the RAE of the Value Function. 137
Figure 7:17 Mid-Rise Towers Sensitivity Analysis, Energy. 138

Figure 7:18 Mid-Rise Towers Sensitivity Analysis, Illuminance..... 138
Figure 7:19 Mid Rise Tower Sensitivity Analysis, Value Function. 138
Figure 7:20 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of Energy. 139
Figure 7:21 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of Illuminance. 139
Figure 7:22 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of the Value Function..... 140
Figure 7:23 Mid Rise Towers impact of Feature Selection on the RAE..... 141
Figure 7:24 Mid-Rise Towers impact of Hyperparameter Tuning on MLP..... 142
Figure 7:25 Mid-Rise Towers impact of Hyperparameter Tuning on SMOReg. 143
Figure 7:26 Mid-Rise Towers impact of Hyperparameter Tuning on RF..... 144
Figure 7:27 Mid-Rise Towers impact factors for MLP algorithm. 145
Figure 7:28 Mid-Rise Towers impact factors for SMOReg algorithm. 146
Figure 7:29 Mid-Rise Towers impact factors for Random Forests algorithm..... 146
Figure 8:1 Diagram Summary of Chapter Eight. 147
Figure 8:2 (Perkins&Will, 2019) A design impression of the Students' Residence project. 148
Figure 8:3 (Perkins&Will, 2019) Students' Residence site and design principles. 148
Figure 8:4 (Perkins&Will, 2019) Students' Residence massing options for DSC analysis. 149
Figure 8:5 (Perkins&Will, 2019) Students' Residence DSC process. 149
Figure 8:6 (Perkins&Will, 2019) Students' Residence massing options and zoning diagram. 150
Figure 8:7 Students' Residence one-way MANOVA Wilk's Lambda test. 151
Figure 8:8 Students' Residence Test of Between-Subjects Effects..... 151
Figure 8:9 Students' Residence MANOVA, Tukey HSD test. 151
Figure 8:10 Students' Residence one-way ANOVA. 152
Figure 8:11 Students' Residence ANOVA, Tukey HSD test. 152
Figure 8:12 Students' Residence impact of Sample Size on the RAE of Site Energy. 154
Figure 8:13 Students' Residence impact of Sample Size on the RAE of Illuminance. 155
Figure 8:14 Students' Residence impact of Sample Size on the RAE of the Value Function 155
Figure 8:15 Students' Residence Sensitivity Analysis, Energy..... 156
Figure 8:16 Students' Residence Sensitivity Analysis, Illuminance. 157
Figure 8:17 Students' Residence Sensitivity Analysis, Value Function. 157
Figure 8:18 Students' Residence impact of Sensitivity Analysis on the RAE of Energy. 157
Figure 8:19 Students' Residence impact of Sensitivity Analysis on the RAE of Illuminance. 158
Figure 8:20 Students' Residence impact of Sensitivity Analysis on the RAE of Value Function 158
Figure 8:21 Students' Residence impact of Feature Selection on the RAE. 159
Figure 8:22 Students' Residence impact of Hyperparameter Tuning on MLP. 161
Figure 8:23 Students' Residence impact of Hyperparameter Tuning on SMOReg. 162
Figure 8:24 Students' Residence impact of Hyperparameter Tuning on RF. 163
Figure 8:25 Students' Residence impact factors for MLP algorithm. 165
Figure 8:26 Students' Residence impact factors for SMOReg algorithm. 165
Figure 8:27 Students' Residence impact factors for Random Forest algorithm. 166
Figure 9:1 Diagram Summary of Chapter Nine. 167
Figure 9:2 Effect of impact factors on MLP. 170

Figure 9:3 Effect of impact factors on SMOReg. 170
Figure 9:4 Effect of impact factors on RF. 171
Figure 9:5 Overall average performance of impact factors. 171
Figure 9:6 Energy scatterplots for the SR and MRT reference projects. 173
Figure 9:7 (Perkins&Will, 2019) Commercial Residential Towers project. 173
Figure 9:8 (Perkins&Will, 2019) CRT project, analysis slices. 175
Figure 9:9 (Perkins&Will, 2019) CRT project, best energy performing levels. 175
Figure 9:10 (Perkins&Will, 2019) CRT project, best daylight factor performing levels. 176
Figure 9:11 (Perkins&Will, 2019) CRT project, best balanced case. 176
Figure 9:12 Future work. Proposed zone-based machine learning process. 180
Figure 9:13 Future work. Transferable zone-based machine learning model. 181

List of Tables

Table 2:1 Summary of data analytical challenges associated with parametric analysis. 42
Table 2:2 Reference projects, initial design space sizes. 43
Table 2:3 Residential Tower project input parameter ranges. 44
Table 2:4 Mid-Rise Towers project input parameter ranges. 45
Table 2:5 Students' Residence project input parameter ranges. 46
Table 2:6 Estimated time to perform parametric analysis on reference project design spaces. 48
Table 2:6 Early stage design process for the Students' Residence project. 49
Table 2:8 Impact of Sampling Ratio on Energy. 52
Table 2:9 Impact of Sampling Ratio for Illuminance. 52
Table 3:1 Comparative summary of literature review sources. 64
Table 4:1 Supervised learning example. 69
Table 4:2 Decision tree attributes. 73
Table 5:1 DSC Objective Definition example. 84
Table 5:2 Structure of consolidated dataset. 95
Table 5:3 Design space for the Sprout Space project 95
Table 5:4 Sprout Space data description. 96
Table 5:5 Sprout Space training master files. 101
Table 5:6 Sprout Space training set files. 101
Table 5:7 Sprout Space sensitivity analysis master files 102
Table 5:8 Sprout Space feature selection master files. 103
Table 5:9 Sprout Space hyperparameter tuning master files. 103
Table 5:10 Hyperparameter ranges for the MLP algorithm. 104
Table 5:11 Hyperparameter ranges for the SMOReg algorithm. 104
Table 5:12 Hyperparameter ranges for the Random Forest algorithm. 104
Table 6:1 Residential Tower data description. 111
Table 6:2 Residential Tower design space description 111
Table 6:3 Residential Tower impact of Sample Size on the RAE of Site Energy. 113
Table 6:4 Residential Tower impact of sample size on the RAE of Illuminance. 113
Table 6:5 Residential Tower impact of sample size on the RAE of the Value Function. 114
Table 6:6 Residential Tower impact of Sensitivity Analysis on the RAE of Energy. 115
Table 6:7 Residential Tower impact of Sensitivity Analysis on the RAE of Illuminance. 116
. Table 6:8 Residential Tower impact of Sensitivity Analysis on the RAE of the Value Function. 117
Table 6:9 Residential Tower impact of Feature Selection. 118
Table 6:10 Residential Tower impact of Hyperparameter Tuning on MLP. 119
Table 6:11 Residential Tower comparative impact of Hyperparameter Tuning on MLP. 119
Table 6:12 Residential Tower impact of Hyperparameter Tuning on SMOReg 120
Table 6:13 Residential Tower comparative impact of Hyperparameter Tuning on SMOReg 120
Table 6:14 Residential Tower impact of Hyperparameter Tuning on RF. 121
Table 6:15 Residential Tower impact of Hyperparameter Tuning on RF. 121

Table 6:16 Residential Tower effect of impact factors on MLP. 123
Table 6:17 Residential Tower effect of impact factors on SMOReg. 123
Table 6:18 Residential Tower effect of impact factors on RF. 124
Table 7:1 Mid-Rise Tower summary of optimal values. 130
Table 7:2 Mid-Rise Tower data description. 134
Table 7:3 Mid-Rise Tower design space description. 134
Table 7:4 Mid-Rise Towers impact of sample size on the RAE of Site Energy. 136
Table 7:5 Mid-Rise Towers impact of sample size on the RAE of Illuminance. 136
Table 7:6 Mid-Rise Towers impact of sample size on the RAE of the Value Function. 137
Table 7:7 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of Energy. 139
Table 7:8 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of Illuminance. 140
Table 7:9 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of the Value Function. 140
Table 7:10 Mid-Rise Towers impact of Feature Selection on the RAE. 141
Table 7:11 Mid-Rise Towers impact of Hyperparameter Tuning on MLP. 142
Table 7:12 Mid-Rise Towers comparative impact of Hyperparameter Tuning on MLP. 142
Table 7:13 Mid-Rise Towers impact of Hyperparameter Tuning on SMOReg. 143
Table 7:14 Mid-Rise Towers comparative impact of Hyperparameter Tuning on SMOReg. 143
Table 7:15 Mid-Rise Towers impact of Hyperparameter Tuning on RF. 144
Table 7:16 Mid-Rise Towers comparative impact of Hyperparameter Tuning on RF. 144
Table 7:17 Mid-Rise Towers impact factors for MLP algorithm. 145
Table 7:18 Mid-Rise Towers impact factors for SMOReg algorithm. 146
Table 7:19 Mid-Rise Towers impact factors for Random Forests algorithm. 146
Table 8:1 Students' Residence data description. 153
Table 8:2 Students' Residence design space definition. 153
Table 8:3 Students' Residence impact of Sample Size on the RAE of Site Energy. 155
Table 8:4 Students' Residence impact of Sample Size on the RAE of Illuminance. 155
Table 8:5 Students' Residence impact of Sample Size on the RAE of the Value Function. 156
Table 8:6 Students' Residence impact of Sensitivity Analysis on the RAE of Energy. 157
Table 8:7 Students' Residence impact of Sensitivity Analysis on the RAE of Illuminance. 158
Table 8:8 Students' Residence impact of Sensitivity Analysis on the RAE of the Value Function. 158
Table 8:9 Students' Residence impact of Feature Selection on the RAE. 160
Table 8:10 Students' Residence impact of Hyperparameter Tuning on MLP. 161
Table 8:11 Students' Residence comparative impact of Hyperparameter Tuning on MLP. 161
Table 8:12 Students' Residence impact of Hyperparameter Tuning on SMOReg. 162
Table 8:13 Students' Residence comparative impact of Hyperparameter Tuning on SMOReg. 162
Table 8:14 Students' Residence impact of Hyperparameter Tuning on RF. 163
Table 8:15 Students' Residence comparative impact of Hyperparameter Tuning on RF. 163
Table 8:16 Students' Residence impact factors for MLP algorithm. 165
Table 8:17 Students' Residence impact factors for SMOReg algorithm. 166
Table 8:18 Students' Residence impact factors for Random Forest algorithm. 166
Table 9:1 Effectiveness of machine learning algorithms. 168

Table 9:2 Data format of consolidated dataset. 168
Table 9:3 Best performing algorithm one-way ANOVA..... 168
Table 9:4 Best performing algorithm Tukey HSD post processing. 169
Table 9:5 Effect of impact factors on MLP 170
Table 9:6 Effect of impact factors on SMOReg. 170
Table 9:7 Effect of impact factors on SMOReg. 171
Table 9:8 Design complexity one-way ANOVA. 172
Table 9:9 Design complexity Tukey HSD post processing..... 172
Table 9:10 Commercial Residence Towers data description. 174
Table 9:11 Commercial Residence Towers design space description. 175
Table 9:12 CRT project results of verification runs. 176
Table 9:12 RAE for the RT and MRT using the RT training set. 179

Abbreviations

ANN	Artificial Neural Network
ANOVA	Analysis of Variance
BC	British Columbia
BPA	Building Performance Analysis
CPU	Central Processing Unit
DOE	Design of Experiments
DSC	Design Space Construction
DSR	Design Space Reduction
DT	Decision Tree
EUI	Energy Use Intensity
GPU	Graphic Processing Unit
GHGI	Green House Gas Inventory
HRV	Heat Recovery Ventilation
HSD	Honestly Significant Difference
LEED	Leadership in Energy and Environmental Design
LEGB	Low Emission Green Building
MANOVA	Multi-variate Analysis of Variance
MLP	Multi-Layer Perceptron
MRT	Mid-Rise Towers project
NZEB	Net-Zero Energy Building
PCP	Parallel Coordinates Plot
PH	Passive House
PHPP	Passive House Planning Package
RAE	Relative Absolute Error
RF	Random Forest
RT	Residential Tower project
SHGC	Solar Heat Gain Coefficient
SMOReg	Sequential Minimal Optimization Regression algorithm

SR	Students' Residence project
SVM	Support Vector Machine
TEDI	Total Energy Demand Intensity
TEUI	Total Energy Use Intensity
WWR	Window to Wall Ratio

Glossary

Artificial Neural Network	A machine learning model comprised of nodes organized into layers and interconnections between the nodes. It learns by assigning weights to the interconnections.
Classification	The machine learning problem of identifying to which of a set of categories an output parameter belongs.
Conventional Analysis	Building performance analysis performed using single-run simulations as is common in current practice.
Decision Trees	A machine learning model comprised of nodes and branches. It learns by splitting data into branches and assigning the classification to the lowest nodes (leaves) of the tree.
Design Alternative	An individual row of a design space.
Design Preferences	Criteria used to evaluate the outputs of design space exploration.
Design Problem	A set of design parameters with specified ranges of values that can be used to represent a design analysis task such as energy analysis or daylight analysis.
Design Solution	Design alternatives that represent the best design preferences.
Design Space	The cross-product of all parameter value spaces of a design problem.
Design Space Complexity	A condition where the design space of a problem is to consist of several smaller design spaces because the project has varying design conditions.
Design Space Construction	A methodological framework for performing parametric analysis in architecture.
Design Space Exploration	A process that maps design spaces to design solutions.
Design Space Reduction	A process that creates a predictive data model by sampling a design space. Also referred to as *meta-modeling* or *surrogate modeling*.
Feature Selection	A process for improving the performance of a machine learning algorithm by running several subsets of input parameters to see if any perform better than the rest.
Hyperparameters	A machine learning algorithm parameter whose value is set before the training process begins and which can affect the performance of the algorithm.
Hyperparameter Tuning	The process of testing different values of an algorithm's hyperparameters to see which perform the best.
Input Parameters	Typically, variables used as inputs into simulation processes.
Output Parameters	Typically, variables used to represent the results of simulation processes.

Parameter	A variable that identifies a quantity of interest in a design problem.
Parameter Space	The set of input parameters of a design space.
Parameter Value	The quantity associated with a parameter in a design alternative.
Parameter Value Space	The range of quantities associated with a parameter in a design space.
Random Forest	A machine learning model that is a type of decision tree. It comprises of several decision trees working together, and it learns by taking votes from its constituent decision trees.
Regression	A machine learning process where the output parameter is a continuous value.
Sample Size	The number of instances in a design space.
Sensitivity Analysis	A process of improving machine learning algorithm performance by using only sensitive input parameters for training.
Supervised Learning	Machine learning where the output parameters in the training data have been assigned values.
Support Vector Machines	A machine learning model that learns by identify a divider (line or hypersurface) between the different values to be classified.
Unsupervised Learning	Machine learning where there are no output parameters, or the output parameters have not been assigned values.
Value Function	An output parameter that combines several other output parameters into one quantity to enable multi-discipline comparative analysis.

Chapter 1 Introduction

This chapter defines parametric analysis and machine learning, and the data analytical challenges of implementing parametric analysis in architectural practice are introduced. Machine learning algorithms are proposed as a solution to these challenges. We also discuss the research background, contextualizing the study in previous research into design space exploration. We then present an outline of the study as a roadmap for discourse.

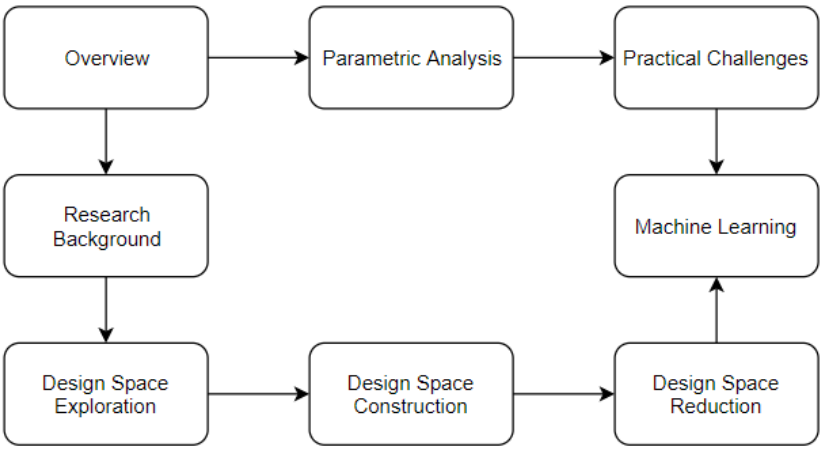


Figure 1:1 Diagram Summary of Chapter One.

1.1 Overview

Parametric analysis is emerging as an important data-driven approach to architectural design in the early stages. Similarly, machine learning is an emergent, data-driven and important discipline of data science. In general terms this study investigates the relationship between these two growing methods in the context of architectural practice. In particular, we investigate how machine learning methods can help to solve data analytical challenges that have been identified when performing parametric analysis in architectural practice.

In the context of this study parametric analysis is a broad and generic term to describe a computational approach to problem solving where a range of input parameters are related and varied, and the impact of such variations on a design problem is assessed. Evins (2013) presented a review of significant research applying parametric analysis to sustainable building design problems. Machairas et al. (2014) reviewed methods and tools for parametric building design optimization. Nguyen et al. (2014) reviewed simulation-based parametric analysis methods focusing on simulation programs, optimization tools, efficiency of optimization methods and industry trends.

With respect to machine learning, Norvig & Russell (1995) took the view of artificial intelligence as intelligent agents interacting with their environment through sensors that receive stimuli and effectors that perform actions. In this model, machine learning is the primary activity of a learning element which is responsible for making improvements to the actions of the agent. Mitchell (1997) characterized machine learning as the improvement by a machine at performing a task through experience. A machine is said to learn from experience with respect to some class of tasks and performance measures if its measured performance at the tasks improves with experience. Alpaydin (2004) described machine learning as programming computers to optimize a performance criterion using example data and past experience. This is done by optimizing the

parameters of a model that defines the problem. The model may be predictive, to make predictions of future observations, or it may be descriptive to gain deeper insight from the data and experience.

Current architectural practice recognizes the need to design high performance buildings and the need for building performance analysis in order to achieve this goal. However, conventional approaches to building performance analysis are based on single-run simulations coupled with the judgement and inference of a building performance expert. Parametric analysis, on the other hand, offers a more analytical approach. By running a large number of simulations data is gathered that can be used for more rigorous design exploration than single-run methods. This design exploration typically includes optimization, uncertainty analysis and sensitivity analysis.

In attempting to implement parametric analysis in architectural practice several data analytical challenges are encountered. These can be characterized as having to do with the size of design spaces, the speed of dynamic simulation, the accuracy of design space reduction models as well as the existence of complex design space conditions. Given the relentless pace of current architectural design processes it is desirable for all design related and design supporting activities to be executed as rapidly as possible. The data analytical challenges of parametric analysis, in general, undermine this requirement. Large design spaces take a long time to simulate. If simulation time is improved by reducing the size of the design space questions regarding accuracy arise. Complex design space conditions tend to increase the size and interdependencies of design spaces by increasing the number or range of analysis parameters involved.

In order to meet the project schedule demands of contemporary practice architects must either increase the speed at which they perform parametric analysis, or they must reduce the size of the design spaces they analyze. Both approaches have been attempted. Increasing the speed of analysis typically involves investment in high performance computational resources. These include high specification hardware as well as computational cloud frameworks like Microsoft Azure™ (Microsoft, 2019) and Amazon Web Services™ (Amazon, 2019). On the other hand, reducing the size of design spaces while maintaining acceptable levels of accuracy typically involves the use of data analytical methods. These include regression methods, design of experiments methods and machine learning methods. In this study we focus on machine learning methods of design space reduction.

As will be seen in Chapter Three, although machine learning methods have been used for parametric building performance analysis, their effectiveness for solving the challenges of parametric analysis on real world projects has not been fully explored. In addition, research into multi-discipline parametric analysis (such as analyzing both energy and daylighting simultaneously) in architectural practice still has room for further exploration. This study seeks to contribute to both these topics. We will ask the questions how effective are machine learning algorithms at resolving the data analytical challenges of multi-discipline parametric analysis? In particular, how effective are machine learning algorithms at resolving the questions of design space size, speed, accuracy, and design space complexity.

These questions will be investigated in the context of three reference projects undertaken at Perkins&Will Architects (Perkins&Will, 2019). The reference projects are a Residential Tower (RT) project, a Mid-Rise Towers (MRT) project and a university Students' Residence (SR) project. In order to effectively perform parametric analysis in practice we introduce a methodological framework called Design Space Construction (DSC). DSC facilitates multi-discipline parametric analysis by integrating a structured decision-making process involving the definition of objectives, generation of alternatives, analysis of alternatives and assessment of their comparative value. However, notwithstanding the DSC methodology, implementation of parametric analysis encounters data analytical challenges. We will perform a series of experiments to investigate how effectively machine learning algorithms respond to these challenges.

The key objectives of the multi-discipline analysis are energy and daylight together with a value function objective that combines the two. The key metric of our experiments will be the accuracy of the algorithms measured by the Relative Absolute Error or RAE (Cichosz, 2014). In addition, we will measure the effect of

four impact factors on the performance of the algorithms. These factors are sample size, sensitivity analysis, feature selection and hyperparameter tuning. The experiments will be performed using regression-based implementations of three common machine learning algorithms: SMOReg (Shevade et al., 1999) an implementation of support vector machines (SVM), MLP (Ware, 2019) an implementation of artificial neural networks (ANN), and RF (Breiman, 2001) an implementation of decision tree algorithms (DT).

1.2 Research Background

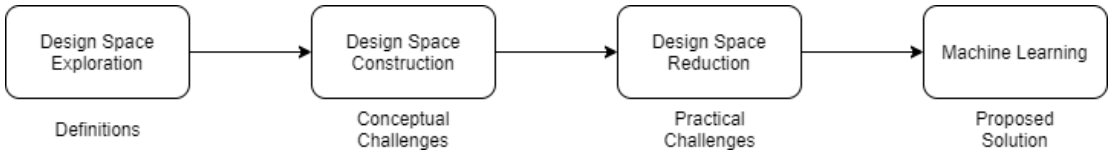


Figure 1:2 Diagram of Research Background.

The design space reduction discussed in this study stems from a line of research that identifies parametric analysis with the idea of design space exploration. Design space exploration has been introduced into computational design theory as a method to leverage computational search paradigms in order to enhance the architectural design process. However, the design theory is subject to conceptual challenges for which methodological solution frameworks have been developed. In attempting to implement these kinds of frameworks practical challenges arise. As mentioned in Section 1.1 key among these challenges is the problem of large design spaces that are not computable using readily available resources. There is need for systematic design space reduction. Therefore, this study investigates the use of machine learning algorithms as a design space reduction strategy.

Woodbury & Burrow (2006) define design space exploration as the idea that computers can depict design as the process of exploring design alternatives. The idea is based on three premises: first, it is a compelling model for designer behavior; second, it is an effective basis for computational support of the design process, and third, it is a feasible approach meaning there are tools and algorithms for its implementation. Woodbury & Burrow discuss design space exploration with reference to these three criteria.

Akin (2006) responded to Woodbury & Burrow and found that like much contemporary design research they did not sufficiently acknowledge non-graphical aspects of design space exploration. While he felt they did well in discriminating the noise from the substance of subject matter that was extensive in scope, he felt they did not take a sufficiently cognitive view of the subject matter. For example, they considered the re-use of design features as being like using copy-paste in a text editor. Akin argued that this is not necessarily the case; a more cognitively aware approach would recognize that a re-used design feature has a different impact arising from a potentially different design context the second time around. He recommended a stronger conceptual framework for the analysis of the broad range of concepts that the authors had attempted.

Krishnamurti (2006) also provided a commentary on the Woodbury & Burrow effort. He appeared to challenge the proposition that explicit representation of a design space along with explicit representation of search strategy is essential for successful design space exploration. For example, he suggested that design is subject to the sunk cost fallacy. Sunk cost is the time, effort or choices invested in by the designer which are non-refundable. In theory, sunk costs should not affect the outcome of a search strategy but, in reality, they do. A deep search in an explicit design space can become quite shallow if sunk costs are considered. Hence the behavior of human designers who iterate through a comparatively small number of alternatives can still be effective when seen through the lens of sunk costs. What we learn from what we discard can be as valuable as what we retain.

These definitions and conceptual refinements helped to guide research into design space exploration. As computation in general has advanced it has become more feasible to attempt actual implementations of

design space exploration in architectural practice. Such attempts, however, have given rise to theoretical and practical challenges that need to be addressed. The theoretical challenges arise from the realization that architectural design is a multi-disciplinary, multi-stakeholder decision making process. How can we effectively plan, control and communicate multi-disciplinary design processes? And how can we ensure that the quality of design spaces used for design exploration is as high as possible? Researchers have developed tools, methods and frameworks to try and answer these questions.

Haymaker (2006) recognized that building design must achieve increasingly complex goals and must therefore involve multi-disciplinary design and analysis. This requires designers to understand the interdependencies and trade-offs between disciplines. He conceptualized a narrative as a collection of information representations with explicit dependencies among them. He proposed a framework called Narratives for constructing and managing representations and dependencies between these representations during multi-disciplinary design and analysis processes.

If design space exploration is intended as a decision-making support strategy, then how can we evaluate the quality of design space exploration and the impact on decision making? Clevenger & Haymaker (2012) observed that performance-based design embodies complex multi-criteria problems that exceed the capabilities of unaided human cognition. This makes it difficult to elicit meaningful design guidance. To mitigate this, designers need the ability to evaluate design strategies. Clevenger & Haymaker proposed a framework for measuring the quality of distinct design strategies. The proposed framework developed design exploration as an iterative cycle of objective identification, alternative generation, impact analysis and value maximization. It synthesized a set of metrics to consistently measure the quality of design exploration as an important component of the design process.

Chachere & Haymaker (2011) proposed a framework for measuring the rationale clarity of design decisions and in this way provided means for evaluating the quality of design decisions. They defined design rationale as the reasons that collectively support a design decision. The components of rationale consist of several individual claims or assertions such as “The building can use a steel moment frame for its structure.” They argue that the clarity of a rationale depends on the clarity of its weakest assertion. They measure the clarity of a rationale using clarity conditions namely: coherence, connectedness, concreteness, consistency, certainty, correctness and credibility.

It appears therefore that the theoretical questions arising from design space exploration can be addressed by the introduction of methodological frameworks. In practice Perkins&Will Architects developed and implemented the DSC framework as a systematic process for performing building performance analysis using parametric methods (Haymaker, et al., 2018). However, when attempting design space exploration using the framework practical challenges arise. Real world projects are complex to define and analyze. They require several parameters spanning broad ranges of values to adequately represent the design problem. This in turn gives rise to large and often intractable design spaces. The most advanced building performance analysis platforms, such as EnergyPlus™ (US Department of Energy, 2019) for energy analysis and Radiance™ (Ward, 2019) for daylight analysis, use dynamic simulation methods that are computationally intensive and comparatively slow. Moreover, data analysis methods such as data visualization and sensitivity analysis are performed on simulated data for effective reporting. However, if we attempt to reduce the size of design spaces in order to improve the simulation time, we risk losing accuracy in our data analytical methods. Lastly, real world projects often exhibit design space complexity for example arising from geometric variations in the design. Parametric Analysis should remain robust in the face of this design space complexity.

In sum, the research background is based on a tradition of research into design space exploration. Early pioneers defined the concepts and sought to situate the research within the broader epistemology of design theory. This early work gave rise to a second wave of research that asked critical theoretical questions of design space exploration in relation to the broader domain of design theory. This in turn gave rise to solution frameworks in answer to many of the emerging research questions. The implementation of these frameworks

on real world projects has exposed practical challenges some of which are associated with the high cardinality of real-world design spaces. We ask can machine learning algorithms perform effectively in helping to resolve these practical challenges?

1.3 Outline

Following this introduction, in *Chapter Two* we will formulate the problem statement of the study. We begin by demonstrating the limitations of conventional methods of building performance analysis. We then discuss the importance of parametric analysis in overcoming these limitations. We identify several data analytical challenges associated with implementing parametric analysis in architectural practice. The problem statement seeks to investigate how effectively machine learning algorithms can perform as a design space reduction strategy to help resolve these challenges.

In *Chapter Three* we review literature relating to machine learning for design space reduction. We begin by discussing the cause of high cardinality design spaces and the need for design space reduction. We then make the case for machine learning as a design space reduction strategy. We review literature related to parametric analysis in architecture and argue it has not thoroughly investigated machine learning for multi-discipline building performance analysis.

In *Chapter Four* we discuss the theoretical background to the machine learning algorithms included in this study. We introduce two types of machine learning algorithms – supervised learning and unsupervised learning and identify parametric analysis as a supervised learning problem. We then discuss the three machine learning algorithms to be used in the study (ANNs, SVMs & DTs). We describe the theoretical basis of the algorithms and identify the key hyperparameters that influence their performance. We also identify specific implementations of these algorithms that we will run – SMOReg, MLP & RF.

In *Chapter Five* we describe the methodology of the research. The research methods we apply are simulation methods and experimental methods. We begin by describing the DSC methodological framework that is needed to effectively integrate multi-discipline parametric analysis. The framework is used in each of three reference projects. The reference projects were designed by Perkins&Will Architects namely a mixed-use commercial residential tower, a mid-rise commercial residential tower and a university students' residence.

We describe the process, tools and metrics for both energy analysis and daylight analysis. We also present a general experimental method for the machine learning experiments. We argue that presenting a general method in one place will avoid repetition. The general method involves a description of the data types used in the experiments, the definition of the parametric design space and the calculation of the value function for multi-discipline analysis. In addition, the general method involves examining the impacts of important factors that affect the performance of the machine learning algorithms. These factors are the sample size of the training dataset, sensitivity analysis, feature selection and hyperparameter tuning.

In *Chapter Six* we discuss our first reference project – Residential Tower. We begin by describing the DSC process that defines the design space for parametric analysis and the process of simulation by which data is generated. We then use the general experimental method to investigate the performance of machine learning for parametric analysis. We investigate the performance of three common machine learning algorithms (ANN, SVM, DT) on three objectives (Energy, Illuminance, Value Function). We also consider the impact of sample size, sensitivity analysis, feature selection and hyperparameter tuning on algorithm performance. We then present the results using comparative graphs and discuss the findings.

In *Chapter Seven* we discuss our second reference project – Mid-Rise Towers. We describe the DSC process that included creating ten slices of the towers for analysis corresponding to the different geometric conditions of the design. As such, in addition to investigating algorithm performance on energy and daylighting objectives, we also investigate the impact of design space complexity on algorithm performance.

On design problems that need several slices for design exploration, can we use fewer simulations per slice to train machine learning algorithms but still achieve acceptable levels of accuracy?

In *Chapter Eight* we discuss our third reference project – Students' Residence. We again use the DSC process to define the design space and the simulation process. In this reference project three different design alternatives are examined to understand if there are statistically significant differences in their performance. As such three geometrically distinct analytical models, with identical input parameters, are combined into a single machine learning model. We seek to understand if this geometrical variation has a significant impact on machine learning algorithm performance.

In *Chapter Nine* we present the findings and conclusions. We first summarize and verify the findings of the study. The summary of findings will answer the question whether machine learning algorithms can perform effectively on multi-discipline parametric analysis. We will also indicate which algorithms perform most accurately and which impact factors most influence algorithm performance. We will test whether machine learning algorithms can be robust under complex design space conditions. Verification of the findings will be done by performing an independent machine learning experiment on a comparable project. We will then describe the impact of machine learning algorithms on design space reduction and draw our conclusions from the study. Suggestions for future work propose a zone-based machine learning data model to enable data reuse by enabling machine learning models to be transferred across different projects.

Chapter 2 Problem Formulation

This chapter formulates the problem statement of the research. We begin by discussing the limitations of conventional methods of building performance analysis and providing examples of questions that conventional approaches cannot easily answer. We describe the importance of parametric analysis and show how it answers these questions better than conventional approaches. We then discuss the data analytical challenges of implementing parametric analysis in architectural practice. These include the size of design spaces, the speed of simulation, the accuracy of predictions, the sampling ratio and the design space complexity problem. Lastly, we state the research question which seeks to investigate machine learning as a design space reduction strategy for resolving the data analytical challenges of parametric analysis in architectural practice.

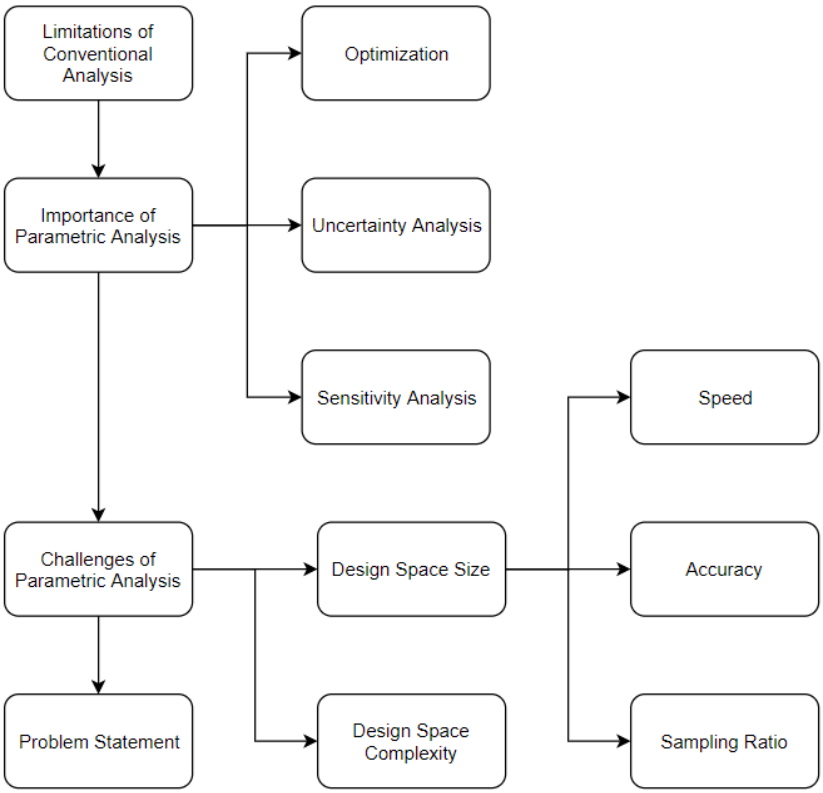


Figure 2:1 Diagram Summary of Chapter Two.

2.1 Limitations of Conventional Analysis

This study seeks to investigate whether machine learning algorithms can be effective at resolving design space reduction challenges related to parametric analysis. To understand these challenges and why it is important to resolve them it is useful to understand the benefits of performing parametric analysis in the first place. These benefits are in turn best understood by discussing some of the limitations of conventional building performance analysis.

What do we refer to as conventional building performance analysis? Traditionally, architects evaluated the environmental context and performance of their designs using analytical diagrams (Figure 2:2). While these diagrams were useful visualization and communication aides, the decision making they supported was not scientific. As computational methods developed environmental analysis moved more and more to using computational tools that performed a myriad of analyses in more rigorous fashion. These analyses include

energy analysis, daylight analysis, solar radiation analysis, acoustic analysis, view access analysis among others.

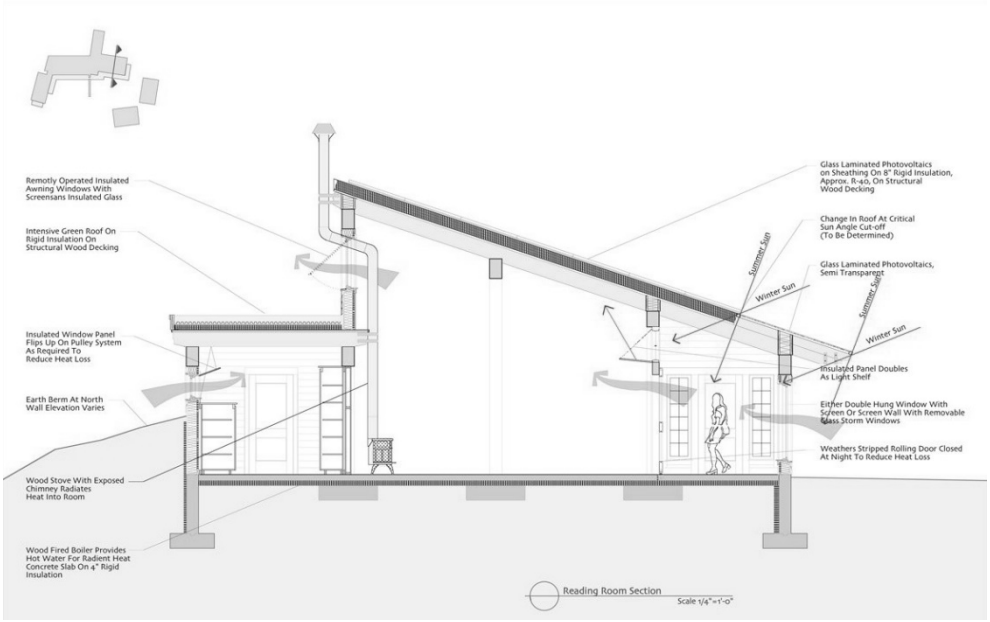


Figure 2:2 (Perkin&Will, 2019) Traditional Environmental Analysis diagram.

The typical process of performing these analyses involves a number of steps (Figure 2:3). First, a simplified analytical model is developed from the detailed design model. Second, analytical data is input to the model. This analytical data includes input parameter values as well as background data such as location and weather information and applicable design standards. Third, simulation is performed using an appropriate simulation platform. Fourth, simulation results are reported in numerical and visual formats. Fifth, results are interpreted, and recommendations proposed. This process is what we refer to as conventional building performance analysis.

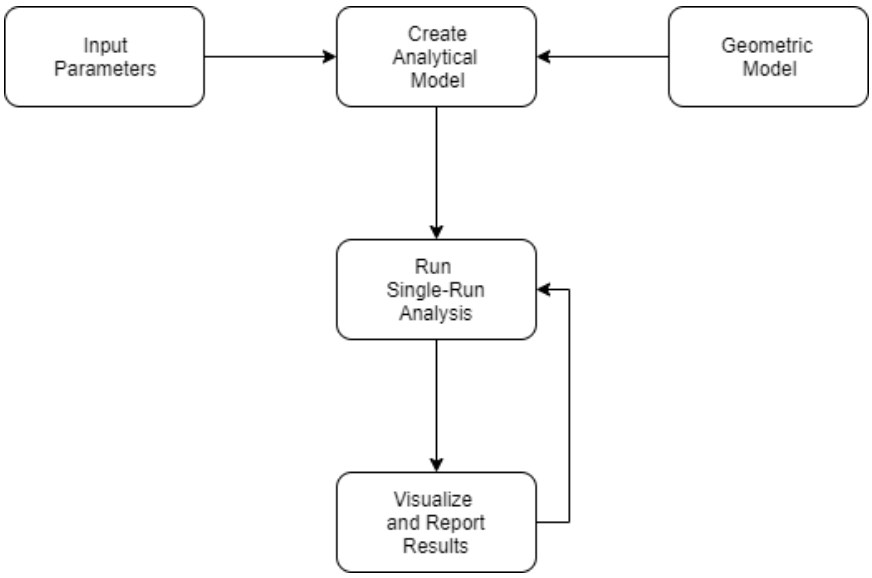


Figure 2:3 Conventional building performance analysis process.

We note that the conventional analysis process does not have to be linear: it can be iterative. After a first iteration the designer may decide to test an alternative to the design by modifying the analytical model or input data. A second iteration is executed and if this produces a better result it gives the designer a sense of

direction. We note further that conventional analysis is single-run based. Although a number of iterations may be executed each iteration is performed and evaluated in isolation. In addition, we note that the conventional approach is typically a single discipline analysis. For example each analysis deals with just one of energy, daylighting, thermal comfort, view quality, etc. Although a number of different disciplines may be analyzed through the course of design, each discipline tends to be analyzed in isolation and the correlation between disciplines is assessed intuitively.

Conventional analysis relies on single-run analysis to provide a general indication of building performance and then uses professional judgement, precedent and intuition to make inferences and draw conclusions. This approach is subjective and qualitative. Parametric analysis, on the other hand, is a quantitative and analytical approach. Research, for example Nguyen et al., (2014), indicates that quantitative approaches lead to superior results compared to conventional analysis. At Perkins&Will we have anecdotally observed that parametric analysis improves building performance design targets by 15 – 20% compared to designer intuition.

To provide examples of questions that are not easily answered by conventional analysis we introduce two projects designed at Perkins&Will. *Balcony Tower* is a commercial-residential tower project located in Surrey, British Columbia (Figure 2:4). *Hotel Residence* is a mixed-use commercial-residential project consisting of a mid-rise residential tower and hotel. It is located in Vancouver, British Columbia (Figure 2:5).



Figure 2:4 (Perkins&Will, 2019) Balcony Tower project.



Figure 2:5 (Perkins&Will, 2019) Hotel Residence project.

2.1.1 Project Example: Balcony Tower

Balcony Tower (Figure 2:4) is a 118m tall residential building, part of a proposed four-tower cluster. The design rationale of the project supports the urban design guidelines of the City of Surrey in seven ways: establishing an identifiable downtown skyline; integrating walkability and transit orientation; creating a high quality, attractive architectural built form; establishing an image of a green city center; providing a high quality, vibrant public realm; using west coast contemporary lines and natural materials, and fostering unique identity in a historical building neighborhood.

In engaging a building performance analysis process the design team sought to answer some specific questions. First, what is the effect of exposed balconies on building energy performance? Second, how can we optimize energy performance if we invest in a high efficiency Heat Recover Ventilation (HRV) system. Third, how can we maximize daylighting and view access while meeting energy performance requirements?

As part of the building permitting process the City of Surrey requires buildings to meet provincial energy requirements referred to as the Step Code (2019). The energy performance requirements for residential occupancies prescribe a range of thermal energy demand intensity (TEDI) values (Figure 2:6). The design team and the client felt that only two of these TEDI values were feasible for the project: 30 kwh/ m^2/year (Step 3) or 45 kwh/m^2/year (Step 2).

Step	Hotels and Motels	Other Group C Occupancies	Hotels and Motels	Other Group C Occupancies
	Equipment and Systems - Maximum Total Energy Use Intensity kwh/m2/year		Building Envelope - Maximum Thermal Energy Demand Intensity kwh/m2/year	
1	Conform to Part 8 of the NECB			
2	170	130	30	45
3	140	120	20	30
4	120	100	15	15

Figure 2:6 (City of Surrey, 2019) Part 3 Step Code Targets.

The project design team proposed extensive use of exposed balconies which create a thermal bridging problem in heating dominated climates like British Columbia. They therefore sought to understand if they could meet the Step Code requirements with partially thermally broken balconies or if they needed fully thermally broken balconies. In other words, what is the minimum effective building envelope R-value that will meet the Step Code requirements?

British Columbia compliance modeling standards typically require the services of a building envelope consultant because there are qualification requirements for approved energy modelers that most architects do not possess. In the case of the Balcony Tower project the building envelope consultant proposed the use of an 80% HRV system which is more efficient than other Perkins&Will projects. The design team therefore sought to understand the minimum envelope R-values that can meet the Step Code given an 80% HRV system. In other words, could they attempt to meet the Step 3 standard rather than the Step 2 standard given the investment in high efficiency HRV system?

In addition to these energy related questions the design team also wanted to maximize their window to wall ratio (WWR) on strategically located facades in order to maximize natural lighting and view access. Vancouver and surrounding cities like Surrey are scenic locations with potential views to green highlands, mountains and the ocean. View access is therefore an important aspect of designing residential projects, improving the value of residential units. Access to daylighting can reduce energy costs and improve the quality of living spaces but conflicts with energy performance. The design team therefore sought to know what is the highest WWR they could achieve while still meeting the energy requirements of the Step Code.

These are multi-disciplinary optimization questions that conventional analysis cannot easily or accurately answer. In the case of thermally broken balconies and efficient HRV systems what is needed is an optimization of the effective R-value of the building envelope. A single-run analysis, or even a few single-run analyses, can at best give the designer an approximate understanding of where this optimum might lie. On the other hand, parametric analysis will run a large number of simulations (hundreds or thousands). The results can be visualized using a PCP or parallel coordinates plot (Figure 2:7). This can be used to perform more accurate optimization.

Using a PCP for the balcony thermal bridging question, the output parameter representing the building TEDI is set to the target energy requirement (30 or 45 kwh/m^2/year). This will filter input parameter combinations that can meet this target. From these input parameter values we can select the minimum effective R-value. Similarly, for the high efficiency HRV system, we can set the TEDI output parameter to meet the Step 3 target (30 kwh/m^2/year). The HRV input parameter is then set to 80% efficiency. This will filter input parameter combinations (if any) that meet these conditions. We can then select the minimum effective R-value.

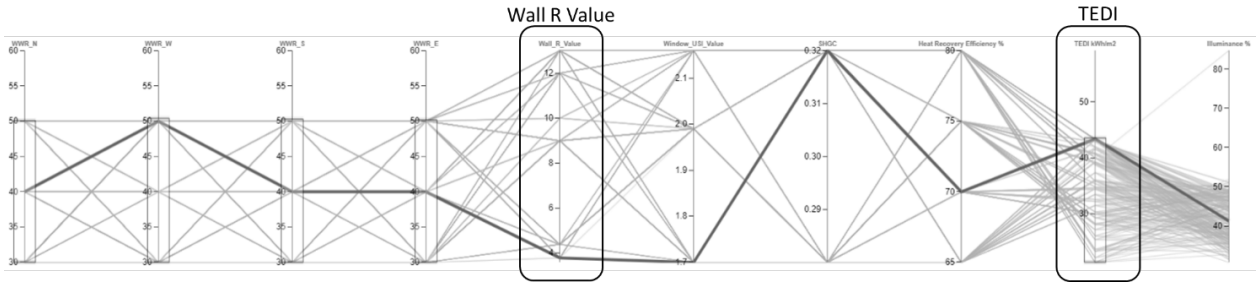


Figure 2:7 (Perkins&Will, 2019) Balcony Tower, thermal bridging optimization PCP.

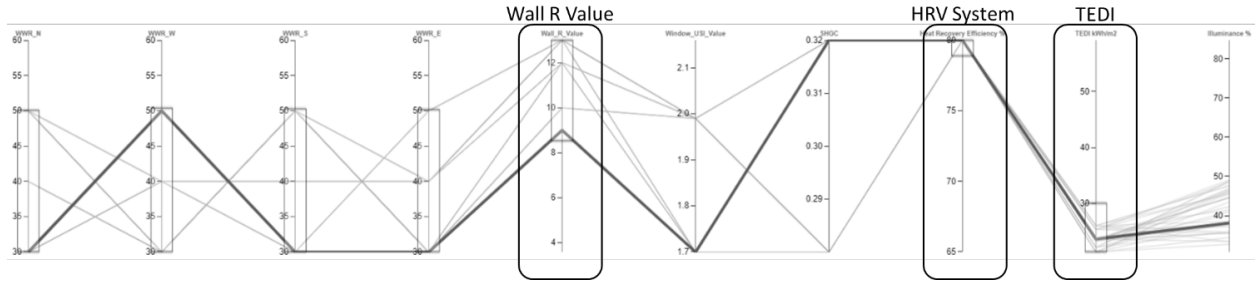


Figure 2:8 (Perkins&Will, 2019) Balcony Tower, HRV system optimization PCP.

In the case of maximizing WWR while achieving energy targets, this is a multi-objective (and multi-discipline) optimization problem. It requires the balancing of conflicting metrics: TEDI for energy and Illuminance for daylight. Performing a single-run energy analysis and a separate single-run daylight analysis does not accurately find this balance. Not only is it not clear how any given input parameter affects any one of the output parameters (TEDI or Illuminance), it is also not clear how improving one output parameter affects the other output parameter. In other words, how does meeting a more aggressive energy target with poor daylighting compare to meeting a less aggressive energy target with better daylighting and vice-versa?

Using a PCP for multi-discipline parametric analysis of the WWR question, we can set the energy output parameter to the target value. This will filter all WWR input parameter combinations that meet the energy requirement. We can then select the highest Illuminance value corresponding to the energy target. Alternatively, the problem can be optimized by introducing a value function that combines normalized output parameter values into a single number. We can then optimize by seeking the input combinations that return the highest value function.

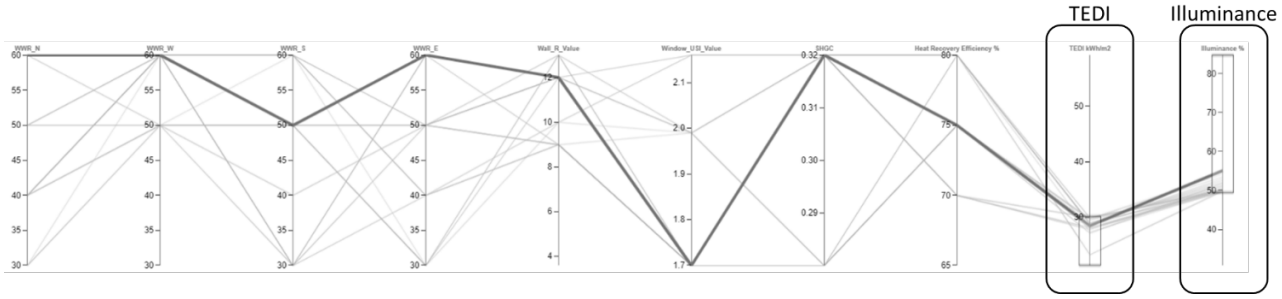


Figure 2:9 (Perkins&Will, 2019) Balcony Tower, optimization of energy and daylight PCP.

2.1.2 Project Example: Hotel Residence

Hotel Residence (Figure 2:5) is a mixed-use development including a 29-story residential tower, a 15-story hotel building and a 4-story podium. Taking advantage of the site's prime location and potential for concentrated growth and high-density forms, the development seeks to cultivate the concept of sustainable urbanism. This is achieved by creating a diverse development to generate and promote a transit-oriented neighborhood for the future. The development will offer ample public spaces at grade, amenities, artist studios, a hotel, and a residential tower with diverse housing options. It will also provide wide sidewalks, streetscapes, and continuous weather protection to enhance walkability in the future transit-oriented and highly urbanized neighborhood.

The Hotel Residence retained a building envelope consultant to facilitate energy compliance requirements. After preliminary analysis the consultant proposed that the energy performance of the project be computed using an area weighted average of the hotel and residential occupancies. They also proposed an air infiltration target of 0.1 liters/sec/m^2. These proposals raised two questions for the design team. First, given that there are two distinct building occupancies which input parameters have the most impact on energy performance? This would allow the designers to focus effort where it was most likely to have a significant

impact. Second, what is the impact on energy performance if the 0.1 liters/sec/m^2 air infiltration (a very aggressive target by industry standards) is not achieved in reality? The design team felt that even if they detailed the building with the intent of meeting this target, variability during construction posed a risk that it would not be achieved.

The first question is a sensitivity analysis problem. It requires multiple instances of the input parameters with corresponding output energy values. From this we can use statistical methods such as the correlation coefficient to compute the correlations between the input parameters and the output parameter. The input parameters with the largest correlation coefficient values are the most sensitive or impactful. Such an analysis cannot be performed through conventional single-run analysis.

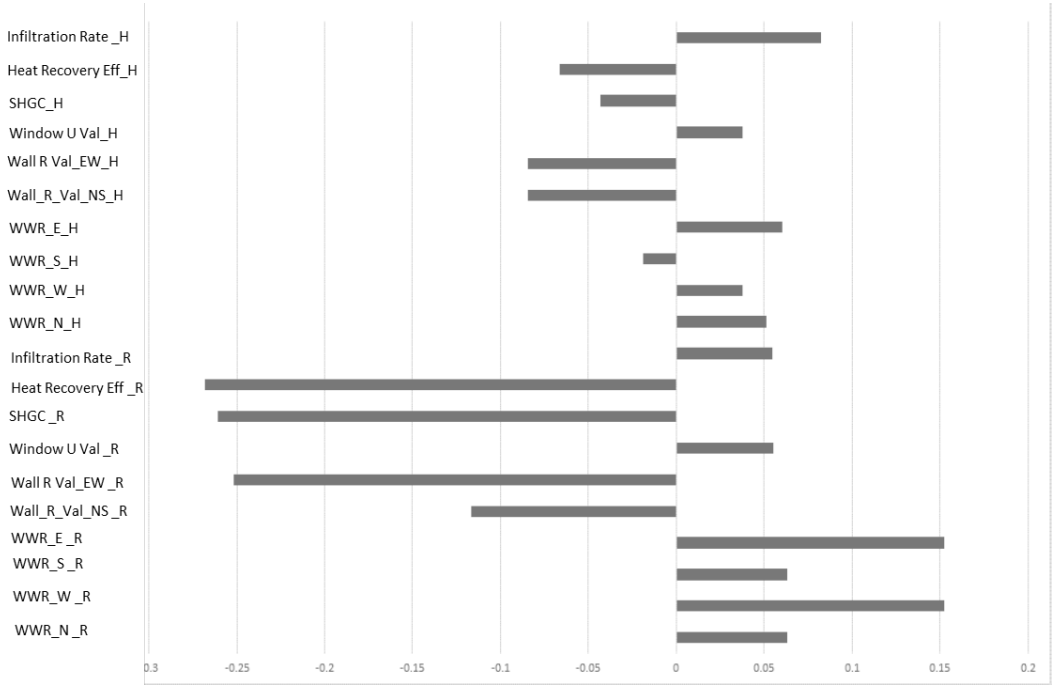


Figure 2:10 (Perkins&Will, 2019) Hotel Residence, tornado graph showing sensitivity analysis.

The second question is an uncertainty problem. The designers wished to understand what happens to the energy output parameter under conditions where the air infiltration value is uncertain. While it would be possible to perform a second single-run simulation using a different air infiltration value to see the effect, this would give an incomplete picture. For example, increasing an input parameter by a given percentage may result in a reduction of the output parameter value; however, if the relationship is non-linear, then a further increase in the input parameter could result in an increase in the output parameter. Conventional analysis cannot detect this kind of complex relationship.

Using parametric analysis, we can solve the first question using statistical methods such as the correlation coefficient method and plotting the results on a tornado graph (Figure 2:10). The second question can also be solved using statistical methods or by inspecting a PCP. We can, for example, perform an independent sample student's t-test on output values at 0.1 liters/sec/m^2 and at 0.2 liters/sec/m^2. If no significant difference is detected, we can confidently infer that performance at the 0.2 value will be as robust as performance at the 0.1 value.

2.2 Importance of Parametric Analysis

In the previous section we discussed some limitations of conventional analysis and indicated how parametric analysis improves on these limitations. In this section we focus on the importance of parametric analysis. The key features of parametric methods discussed are optimization, including multi-objective

optimization, uncertainty analysis and sensitivity analysis. Optimization seeks to find the design alternative that maximizes or minimizes the output parameter values. Multi-objective optimization seeks to maximize or minimize the value of multiple output parameter values simultaneously. Uncertainty analysis seeks to understand the impact of variations of the input parameter values of a design space on the output parameters. Sensitivity analysis seeks to understand the impact of variations of each individual input parameter on the output parameter.

2.2.1 Multi-Discipline Optimization

Machairas et al. (2014) define optimization as the procedure of finding the minimum or maximum value of an output parameter by choosing input parameter values subject to constraints. Optimization can be single-objective or multi-objective when there is more than one output parameter to optimize. If a multi-objective optimization involves output parameters from different disciplines, such as energy and daylighting, then it is a multi-discipline optimization.

Nguyen et al. (2014) classified building optimization into seven classes including one-dimensional vs multi-dimensional optimizations, continuous vs discrete optimizations, and constrained vs unconstrained optimizations. They identified eight general classes of commonly used optimization algorithms including direct search, gradient-based and stochastic-based. Typical algorithms include hill climbing methods, simulated annealing, genetic algorithms and swarm optimization.

They reported the efficiency of optimization in improving building performance for a range of climate types. In cold temperate climates they found reductions of 23 – 49% compared to conventional analysis in space heating energy consumption could be achieved. In the United States they reported that, on average, a reduction of 20 – 30% of building energy consumption is an achievable target. In warmer, cooling dominated climates like Florida the reduction of total energy consumption was 7.1%.

They described several challenges related to building performance optimization. These included challenges with handling discontinuous parameters or discrete parameters; challenges with multi-objective optimization, such as how to deal with the weights of the different output parameters in the objective function, and challenges related to the number of input parameters which they recommended should be constrained to about ten. Other issues included the optimization of computationally expensive models which can be resolved through high performance computing or meta-modelling (what we are referring to as design space reduction), the presence of uncertainty in the analysis and the challenge of integrating building performance analysis tools with building optimization tools.

Shi & Yang (2013) observed that research into multi-objective optimization for building performance was mainly driven by engineers. They proposed that this process should be undertaken by architects as well – but this requires tools and workflows compatible with architectural practice. They proposed a multi-discipline optimization workflow using Rhinoceros™ (Robert McNeel, 2019) and Grasshopper™ (Grasshopper, 2019) for analytical modeling, Radiance for daylight analysis and EnergyPlus for energy analysis. Lartigue et al. (2014) developed a methodology for optimizing the envelope of a building with respect to heating, cooling and daylighting. They recognized that many optimization studies were concerned with energy consumption without considering thermal comfort and daylighting. They used a Pareto optimized approach for retrofitting a dormitory room and found that their method quickly and accurately provided optimal solutions. Flager et al. (2009) performed a multi-disciplinary optimization of a classroom for structural integrity, energy consumption, daylighting as well as first cost and lifecycle cost. They used a genetic algorithm for structural optimization, design of experiments and a gradient based optimizer for energy optimization, and Darwin (2004) for the multi-discipline optimization.

2.2.2 Uncertainty Analysis

De Wit & Augenbroe (2002) observed that building performance design decisions, whether they are based on simulation, design guidelines or expert judgement, tend to be deterministic. Coakley et al. (2014) argued that there are significant differences between building performance analysis predictions and actual measured building performance. Part of this difference can be attributed to uncertainty and this should be reflected in design decision making. This requires the application of effective uncertainty analysis methods.

Tian et al. (2018) identified four sources of uncertainty in building performance analysis: weather data, building envelope, mechanical systems and occupant behavior. Weather, particularly at this time of global climate change, is highly uncertain. Since weather data is a key input of building simulation algorithms it follows that simulation outputs will have weather related variations. Moreover, weather data is typically based on past meteorological data which does not necessarily correlate strongly with future weather conditions. With respect to building envelope performance influencing parameters can be thermal parameters or surface parameters. Thermal parameters include conductivity, density and specific heat capacity of materials. The main surface parameters are solar absorption or reflectance as well as thermal emissivity. Other parameters influencing building envelope performance are thermal bridging, convective heat transfer coefficient and thickness of building materials.

Building performance analysis assumes that mechanical systems operate in ideal conditions whereas the performance of mechanical systems is affected by factors such as oversizing, maintenance and wear and tear. Wang et al. (2012) examined the uncertainty of energy consumption arising from building operation approaches such as lighting control and HVAC operation schedules. Yan et al. (2017) examined the impact of outdoor air control on energy consumption and indoor air quality.

Occupant behavior can account for up to 30% variation in building performance (Eguaras-Martinez, Vidaurre-Arbizu, & Martin-Gomez, 2014). This variance arises because building performance analysis uses deterministic operational schedules to simulate demand fluctuations. However, occupant behavior is complex and certainly non-deterministic. This means that new approaches for modeling occupant behavior as a critical building performance input are required. These new methods can be classified as implicit, where occupant behavior is predicted based on interaction with building controls, or as explicit, where occupant behavior is predicted based on the occupant's state such as whether they feel too cold or too hot.

Rezaee (2016) argues that uncertainty analysis is essential for design space exploration and design decision making. She distinguishes between aleatory uncertainty and epistemic uncertainty. Aleatory uncertainty refers to an inherent randomness in the behavior of a system. Epistemic uncertainty refers to a lack of knowledge about the behavior of a system. Aleatory uncertainty is irreducible while epistemic uncertainty can be eliminated by thorough examination.

Understanding the uncertainties in building performance analysis can help determine what to account for in early design. Following De Wit, Rezaee categorizes uncertainty in building performance analysis tools as numerical, modeling, scenario and specification. Numerical uncertainties arise from computational and numerical imperfection; modeling uncertainties are associated with an analytical model's inherent imperfection as an abstraction of reality; scenario uncertainties are associated with adjacency conditions such as occupancy patterns and external weather conditions. Specification uncertainty (or design parameter uncertainty), has two types: decided parameter uncertainty encodes a lack of knowledge about known parameters whose values may be uncertain while undecided parameter uncertainty encodes a lack of knowledge of the values of unknown parameters. In other words, decided parameter uncertainty relates to uncertainty in the parameter value space while undecided parameter uncertainty relates to uncertainty in the parameter space itself.

Uncertainty in the values of input parameters means that there will be a range of possible values for output parameters. Understanding this range is important for design because it allows for design decision making that is robust. Hoes et al. (2011) define robustness as the ability of a building to handle changes (or disturbances) in the building environment and maintain the required performance. Jurecka (2007) identifies three types of variables in parametric analysis that influence robustness: control variables, noise variables and fixed variables. Control variables can change values but are under the control of the designer (such as window U-values on a building façade); noise parameters represent the source of variation (such as quality of workmanship when installing the façade glazing) while fixed variables are relevant to the analysis but do not change value (such as the area of the building façade on a renovation project). The goal of robust design is to determine values of the control variables such that the resulting design is insensitive to variations arising from the noise variables.

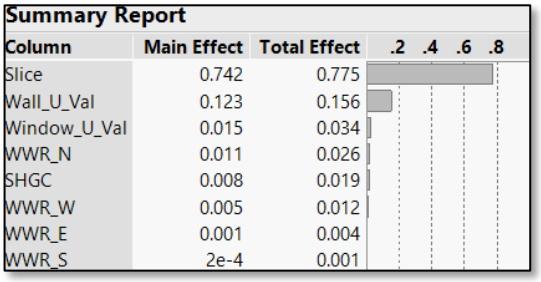
2.2.3 Sensitivity Analysis

Nguyen et al. (2014) have defined sensitivity analysis as the effect of a given input on the output of a parametric analysis. If a change of an input parameter *X* produces a change in an output parameter *Y*, and if these changes can be measured, then we can determine the sensitivity of *Y* with respect to *X*. Architects, faced with a decision on how to optimize the performance of a design, cannot easily recognize which parameters have the most influence. This means they often spend effort and resources on design strategies that do not much improve the desired outcomes. Sensitivity Analysis can help architects prioritize their design efforts towards the most impactful design inputs.

Methods of performing sensitivity analysis can be classified as local or global. Heiselberg et al. (2009) define local sensitivity analysis as a one-factor-at-a-time method where evaluation of output variability depends on the variation of one input parameter at a time while all other inputs are held constant. The input to output relationship is assumed to be linear and any interaction between input parameters is not considered. Global sensitivity analysis evaluates output variability due to one input parameter by varying all input parameters together.

Local sensitivity analysis can be performed by establishing a base case value for each parameter and then evaluating its impact on the output when it has the maximum and minimum values of its range (Tian W. , 2013). Yang et al. (2016) discuss four global sensitivity analysis methods: regression based, Morris design, variance-based and meta-modelling. Regression based methods standardize the coefficients of a regression equation thereby making them directly comparable. Morris method computes the absolute differences between simulated values S and changed values $S + \Delta S$ for each input variable. The mean μ of these differences is an indicator of how important a variable is in determining the output and the standard deviation σ indicates the amount of non-linearity and interaction with other variables. Fourier Amplitude Sensitivity Test (FAST) is a common variance-based method. It uses Fourier series to help compute main effects and total effects of an input variable on the output. Main effects do not include interactions with other variables while total effects combine main effects and interactions. The meta-modeling approach involves creating a meta-model and then using the computationally cheap calculated outputs from the meta-model to perform sensitivity analysis.

Delgarm et al. (2018) used one-factor-at-a-time and variance based sensitivity analysis to assist in building performance analysis on the early stage design of a multi-story building located in Iran. They established that window size was the most dominant variable for the annual cooling, heating and total building energy need assessment. Heiselberg et al. (2009) demonstrated sensitivity analysis on a seven-story office building. They used the Morris method and found that lighting control and the amount of ventilation during winter were the most important input parameters. Corrado & Mechri (2009) used the Morris method to perform a sensitivity analysis on a two-story single-family home located in Turin. They found that only 10 out of 105 input parameters significantly impacted the performance and the others could be neglected.



Summary Report

Column	Main Effect	Total Effect	.2 .4 .6 .8
Slice	0.742	0.775	
Wall_U_Val	0.123	0.156	
Window_U_Val	0.015	0.034	
WWR_N	0.011	0.026	
SHGC	0.008	0.019	
WWR_W	0.005	0.012	
WWR_E	0.001	0.004	
WWR_S	2e-4	0.001	

Figure 2:11 (Perkins&Will, 2019) Sensitivity Analysis for the Mid-Rise Towers project.

2.3 Challenges of Implementing Parametric Analysis

Implementing parametric analysis in architectural practice faces different types of challenges. These include poor attitudes by architects towards quantitative methods as part of the design process; potential conflicts with the role of other consultants and sub-consultants; compliance standards that do not support computational approaches, and data analytical challenges related to design space reduction. In this study we focus exclusively on challenges related to design space reduction. We argue that machine learning algorithms can help to resolve these types of challenges.

The parametric analysis process improves on single-run conventional analysis by generating large amounts of data that can be analyzed more rigorously. Figure 2:12 shows how the parametric analysis process differs from the conventional analysis process shown in Figure 2:3. However, if the design space size is too large then it cannot be simulated in a reasonable amount of time. This means the design space must be systematically reduced (Alternatively, an investment can be made in hardware-based methods to improve simulation speeds). The process of making this reduction is what we refer to as design space reduction.

The design space reduction process involves three steps (Figure 2:13). First, the design space is sampled. Second, a data model is created based on the sample. In this study the data models we create are machine learning models. Third, predictions are made on the design space using the data model. Since prediction is much faster than simulation this makes the parametric analysis process more feasible. It should be noted that in our usage design space reduction is not the same as sampling. Sampling is, in fact, a step in the design space reduction process. In this study we will use sampling based on design of experiments methods, however we will develop data models based on machine learning.

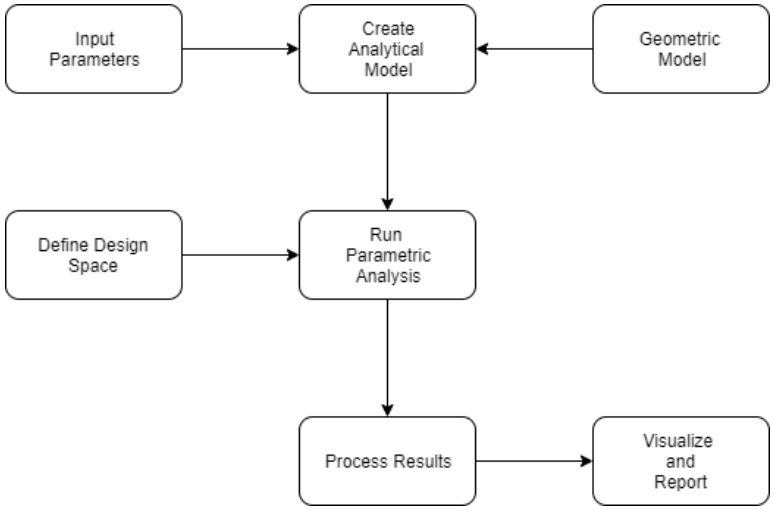


Figure 2:12 Parametric Analysis process.

The design space reduction process raises questions. Understanding that the accuracy of predictions depends on the sample size from which the data model is created, what is the appropriate sample size that gives acceptable prediction accuracy? This means we have to define what we mean by acceptable accuracy in the first place. How quickly can we simulate sample sizes associated with acceptable accuracy levels? This means we have to establish a threshold on the amount of time we can spend simulating the sample. Does the size of the sample in relation to the size of the design space make a difference? Furthermore, for a robust process, we need to understand how variations in design space complexity affect the prediction accuracy. Lastly, we need to understand what techniques are effective in improving the accuracy of our machine learning models. We do this by studying the effect of impact factors (sample size, sensitivity analysis, feature selection, hyperparameter tuning) on the accuracy of machine learning algorithms.

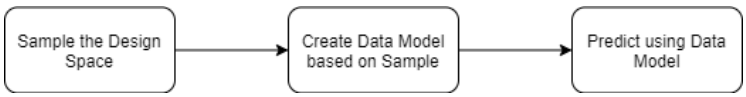


Figure 2:13 Design Space Reduction process.

The questions raised above and summarized in Table 2:1 are what we refer to as data analytical challenges associated with parametric analysis in architectural practice. In this section we will discuss these questions leading up to the problem statement in Section 2.4. In addition, it is instructive to demonstrate (Figure 2:14) how machine learning modifies the parametric analysis process depicted in Figure 2:12.

CHALLENGE	DESCRIPTION	PROPOSED SOLUTION	QUESTION
Design Space Size	Large design spaces take too long to simulate	Reduce the size of the design space.	How does this affect accuracy of predictions?
Accuracy	Literature and practice indicate that acceptable accuracy is between 90 - 95%	Use a sample size that guarantees acceptable accuracy.	How does this affect simulation speed?
Speed	Design process proceeds rapidly compared to simulation speed	Use sample size that can be simulated in less than 33hrs (1000 instances)	How does this affect accuracy of predictions?
Sampling Ratio	Large design spaces need large sample sizes	Use a sampling ratio between 1:10 and 1:100	What if this gives a sample size > 1000?
Design Space Complexity	Real projects have complex geometry and adjacencies	Create a machine learning model that encompasses complexity	Can machine learning algorithms be robust under complex design space conditions?

Table 2:1 Summary of data analytical challenges associated with parametric analysis.

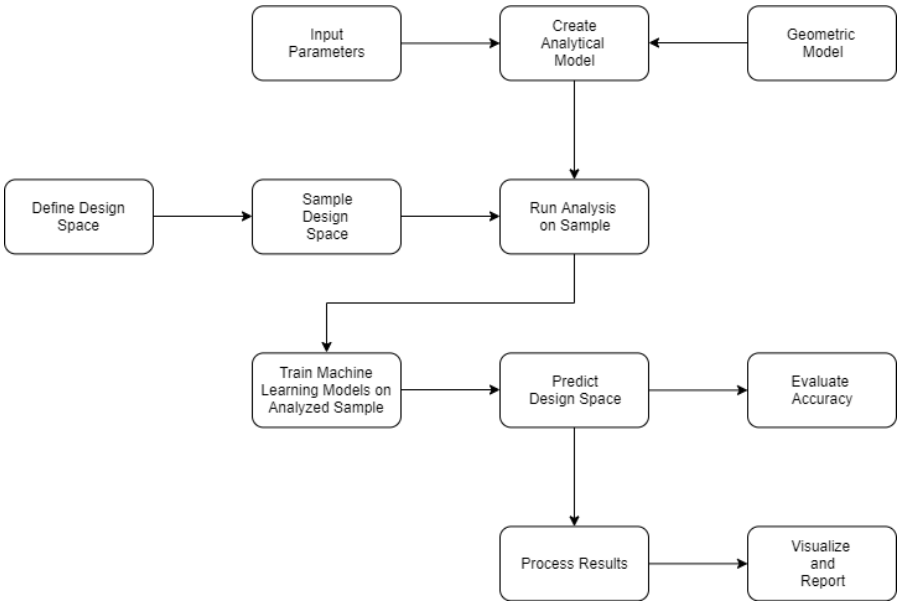


Figure 2:14 The Parametric Analysis process incorporating Machine Learning.

2.3.1 Design Space Size

Project	Size of Design Space
Residential Tower	2,733,750
Mid-Rise Towers	51,840
Students' Residence	354,294

Table 2:2 Reference projects, initial design space sizes.

Real world design spaces tend to be large in order to adequately represent the complexity of design problems. The actual size depends on the goals of the parametric analysis (this can in turn depend on project type, project stage, project size, etc.) and how much resolution the designers require. This can be demonstrated by considering the initial design spaces of three reference projects investigated in this study (Table 2:2).

The specification for each of the design spaces is made by a building performance expert (such as an energy modeler) working with the project designer. For the Residential Tower (Figure 2:15) the designers sought to understand the performance effect of glazing ratios on different orientations, the shading effect of balconies on different orientations as well as the effects of thermal bridging calculated using the Chi Value parameter. They specified input parameter ranges shown in Table 2:3.



Figure 2:15 (Perkins&Will, 2019) Residential Tower project.

Parameter	Parameter Value Space	Number
WWR North	30, 40, 50, 60, 70	5
WWR West	30, 40, 50, 60, 70	5
WWR South	30, 40, 50, 60, 70	5
WWR East	30, 40, 50, 60, 70	5
Balcony Depth North	0, 5, 10	3
Balcony Depth West	0, 5, 10	3
Balcony Depth South	0, 5, 10	3
Balcony Depth East	0, 5, 10	3
Wall U Value	0.284, 0.189, 0.141	3
Chi Value	1.19, 0.48	3
Window U Value	0.85, 1.4	2
SHGC	0.3, 0.5, 0.7	3
	Total	**2,733,750**

Table 2:3 Residential Tower project input parameter ranges.

The Mid-Rise Towers project (Figure 2:16) sought to understand the effect of geometric variation on different facades of the building and the overall impact on project performance. The project had three towers and two podiums with different program requirements on different levels and at different orientations. There were also potential self-shading effects from shadows cast on each tower by the other towers at different times. The conventional approach to building performance analysis, as carried out by an engineering consultant, recommended a single optimized value for each of the input parameters. The design team, however, felt that this was not a sufficiently nuanced approach. They sought to better understand the optimum inputs at the different levels and orientations of the design. They specified input parameter ranges shown in Table 2:4.



Figure 2:16 (Perkins&Will, 2019) Mid-Rise Towers project.

Parameter	Parameter Value Space	Number
WWR North	25,35	2
WWR West	45,55,65	3
WWR South	25,35	2
WWR East	45,55,65	3
Roof R Value	20,30,40,50	4
Wall R Value	5,7,10,15	4
Window U Value	0.25,0.3,0.35	3
SHGC	0.3, 0.35,0.4	3
Number of Slices		10
	Total	**51,840**

Table 2:4 Mid-Rise Towers project input parameter ranges.

The Students' Residence (Figure 2:17) analysis involved analyzing three different design alternatives, or massing types, to understand which one had the best performance for further design development. Parametric analysis was performed on all three design alternatives using identical input parameters (Table 2:5). This allowed for statistical testing to understand if there were significant differences in the three output datasets. One-way analysis of variance (ANOVA) identified that there were indeed significant performance differences between the three design alternatives, and further testing (Tukey's HSD) identified a clearly superior alternative.



Figure 2:17 (Perkins&Will, 2019) Students' Residence project.

Parameter	Parameter Value Space	Number
Massing Type	A, B, C	3
WWR North	30, 40, 50	3
WWR West	30, 40, 50	3
WWR South	30, 40, 50	3
WWR East	30, 40, 50	3
Shading Depth North	0, 1', 2'	3
Shading Depth West	0, 1', 2'	3
Shading Depth South	0, 1', 2'	3
Roof Effective U value	0.08, 0.1, 0.15	3
Wall Effective U Value	0.1, 0.15, 0.2	3
Window U Value	0.85, 1.1	2
SHGC	0.3, 0.5, 0.7	3
	Total	**354,394**

Table 2:5 Students' Residence project input parameter ranges.

Large design spaces imply long simulation times or slow overall simulation speeds. For practical purposes large design spaces must be reduced. This reduction typically involves sampling a subset of the design space, developing a data analytical model based on the sample, and then using the model to make predictions on the full design space. Prediction is much faster than simulation.

However drastic design space reductions, while resulting in dramatic improvements in overall simulation speed also result in decreased prediction accuracy. We must therefore answer the question of what the appropriate prediction accuracy is for parametric analysis. As a follow on to this question we need to understand the sample sizes that assure us of this level of accuracy.

Design spaces of the magnitude of our reference projects would take several days or weeks to simulate depending on the equipment being used. When measured against the speed at which design processes evolve this can be a bottleneck hampering the implementation of parametric analysis. Design space reduction provides the means to sample the design space in order to reduce the number of raw simulations.

We note that in Chapters 6 – 9 we will introduce design spaces for each reference project that are substantially reduced from the initial design spaces shown in this section. This is because stratified design space reduction (Section 2.3.4) was used to reduce the design space size and achieve sampling ratios suitable for machine learning. For practical purposes we recommend a sample size of 1000. This represents a sample that can be simulated in about 33 hours (Assume simulation is performed using high specification hardware as indicated in Section 2.3.2 and using the dynamic simulation tools indicated in Chapter 5).

We choose 1000 simulations (33 hours, assuming 2 minutes per simulation) as a benchmark considering the worst-case scenario where designers need to respond to design questions overnight. This happens frequently in practice as shown in Figure 2:18. We have listed the formal design presentations that were required of the design teams within two months of project kick-off for three of our reference projects. We see that there are a number of occasions where the design team had a single day between presentations. If parametric analysis is to be robust to the needs of architectural practice, then it should be responsive to aggressive project schedules.

PROJECT	PRESENTATION TYPE	DATE	DURATION DAYS
Balcony Tower	Kick Off	20-Sep	
	Internal	12-Oct	22
	Client	13-Oct	1
	Internal	18-Oct	5
	Internal	24-Oct	6
	Internal	1-Nov	8
	Internal	3-Nov	2
	Client	16-Nov	13
Hotel Residence	Kick Off	16-Apr	
	Internal	19-Apr	3
	Internal	1-May	12
	Client	23-May	22
	Internal	28-May	5
	Internal	25-Jun	28
Mid Rise Towers	Kick Off	24-Oct	
	Client	25-Oct	1
	Client	16-Nov	22
	Planning	26-Nov	10
	Client	27-Nov	1
	Internal	28-Nov	1
	Client	18-Dec	20

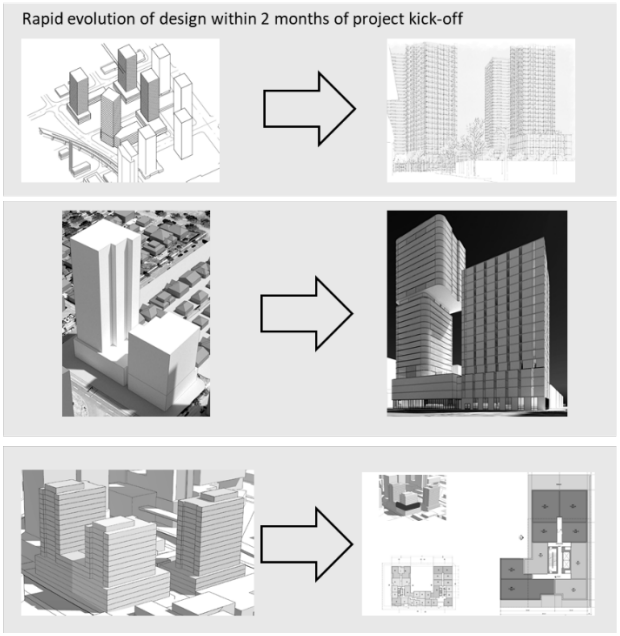


Figure 2:18 (Perkins&Will, 2019) Rapid rate of design changes on three reference projects.

2.3.2 Speed

The overall simulation speed of a parametric analysis exercise depends on the simulating equipment and simulation settings such as the resolution of the analysis grid during daylight simulation. As mentioned in Section 2.3.1 very slow overall simulation speeds will prevent parametric analysis from keeping pace with design decision making. Strategies for dealing with slow overall simulation speeds include cloud computing, parallel computing and GPU computing. These are hardware-based strategies. While hardware-based strategies have their merits, they have also been criticized.

Garg et al. (2011) described how to perform parallel processing for annual energy simulation. Zhang (2010) developed a parametric analysis tool for parallel processing of EnergyPlus simulations. Mazzarella et al. (2014) described the advantages of using cloud computing for building performance simulation and optimization. However, Richman et al. (2014) identified potential issues with these approaches. They include the costs of cloud services on frameworks like Amazon and Microsoft Azure, the challenge of setting up these simulations, the need to monitor the process carefully and the fact that for some design problems, even with cloud computing, the design spaces are so large that computation time can still run into months.

Alternatively, design space reduction strategies using statistical or algorithmic methods can be used to sample the design space and thereby improve the overall simulation speed. In this context the importance of simulation speed can be seen by looking at Table 2:6. We present time estimates for simulating the design spaces of the three reference projects on generally accessible high specification hardware (For example this study uses *Micro-Star International GS65 Stealth Thin 8RF; Intel® Core™ i7-8750H CPU @ 2.20GHz (12 CPUs); 32GB RAM; NVIDIA GeForce GTX 1070 with Max-Q Design, 8082MB VRAM).* This hardware executes at approximately 2 minutes per simulation (30 seconds for energy simulation and 90 seconds for daylight simulation). This time will depend on analysis settings like the resolution of daylight analysis grids and the number of light ray bounces for raytracing. We see that the time taken would range from 2 months to 10 years. Parallel processing can reduce these times substantially, but this requires an investment in hardware. Additionally, using the Students' Residence as an example, we can demonstrate the rate at which a typical design decision making process evolves (Table 2:7). Without a serious investment in hardware or design space reduction, the parametric analysis simulation speed will be too slow to inform the design process.

Project	Size of Design Space	Duration (mins)	Duration (days)
Residential Tower	2,733,750	5,467,500	3,797
Mid-Rise Towers	51,840	103,680	72
Students' Residence	354,294	708,588	492

Table 2:6 Estimated time to perform parametric analysis on reference project design spaces.

2.3.3 Accuracy

The accuracy question of parametric analysis is three different questions: simulation accuracy, prediction accuracy and sample accuracy. The first question is how reliable the actual results of the simulation are. In this case the sources of inaccuracy are user error or the simulation engine itself. Simulation engine developers perform thorough benchmarking and testing of their tools for example: DesignBuilder Space Heating Equipment Performance Tests (2017); DesignBuilder Space-Cooling Equipment Performance Comparative Tests (2017); DesignBuilder Building Thermal Envelope and Fabric Load Tests (2017), and DesignBuilder Space-Cooling Equipment Performance Analytical Verification Tests (2017). In addition, tools that have been in use for a while will have been user tested by a community of experts who would report and publicize any glaring inaccuracies. We therefore take the view that industry standard tools are likely to be reasonably accurate for our purposes.

The second question related to accuracy has to do with prediction accuracy. Once we have created a data model and used it to make predictions, we must understand how different the predicted values are from what we actually get by simulation. This difference, or prediction error, is a measure of the accuracy of the data model for given sample sizes. The metric we will use to calculate accuracy is the Relative Absolute Error (RAE) computed as follows:

$$\mathrm{RAE} = \frac{100\%}{n} \sum_{1}^{n} \frac{|x - \widehat{x}|}{|\bar{x} - \widehat{x}|}$$

where $\hat{x}$ is the predicted value and $\bar{x}$ is the mean value of the output parameter, and *n* is the number of instances (sample size).

The third question has to do with sample accuracy. As we will see there are occasions where design space reduction is itself reduced to a sampling exercise. No predictive data models are developed, and estimates are made based on nothing more than the sample. Such sampling obviously increases simulation speed, but it also decreases the accuracy of sample statistics.

To demonstrate the effect of sample size on accuracy we use data from the Residential Tower reference project. We perform an experiment that involves:

1. Run 1,500 daylighting simulations to generate an experimental dataset. We will use the daylighting simulation metric *Illuminance* for this experiment.

2. Sort the daylighting results in descending order. Calculate the average of the top 100 results to obtain an average optimal result *(Average Optimum).*

3. Create 8 random samples from the daylighting result set of sample size 100, 300, 500, 700, 900, 1100, 1300, 1500.

4. Sort each sample in descending order. For each sample calculate the average of the top 100 results to obtain an average optimal result *(Average Sample Optimum).*

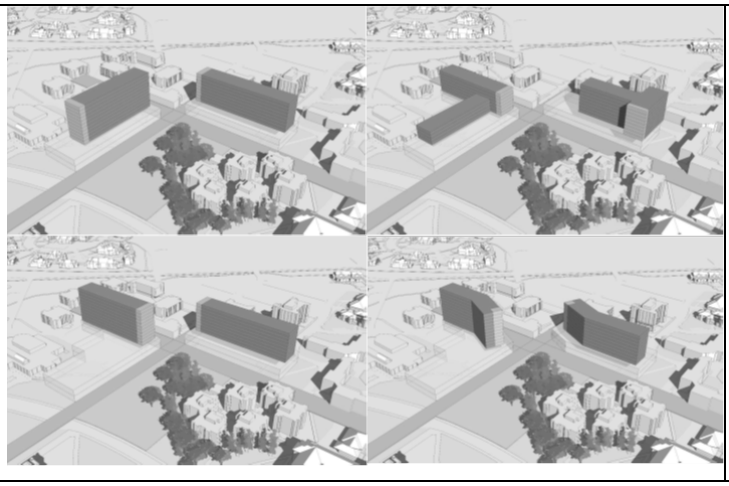
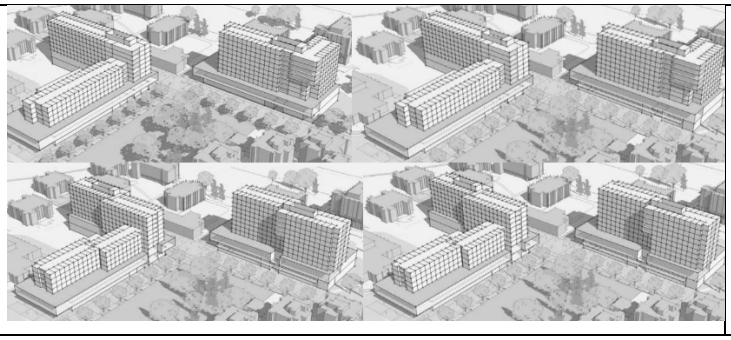
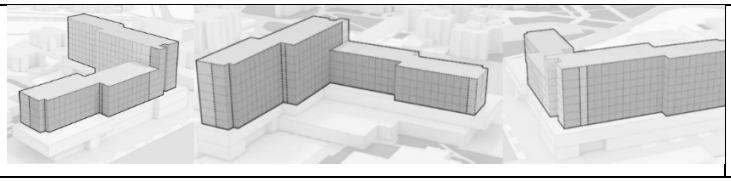
DURATION	MASSING DEVELOPMENT	DESIGN FACTORS	PARAMETRIC ANALYSIS
	INITIAL MASSING		
		• Massing and Orientation • Height • Setbacks • Identity and presence • User experience • Neighborhood	Provide guidance: • Form factor • Orientation
	DETAILED MASSING		
21 days		• Homeliness • Relationship to context • Financial Sustainability • Passive Design Principles	Provide guidance: • Form factor • Glazing percentage
	INITIAL ENVELOPE		
14 days		• Program requirements • Performance • Aesthetics	Provide guidance: • Glazing percentage • Glazing performance
	DETAILED ENVELOPE		
13 days		• Focused exploration • Shading strategy • Overhangs • Articulation • Materials • Scale	Provide guidance: • Sun shading • Glazing • Material Selection

Table 2:7 Early stage design process for the Students' Residence project.

5. Calculate the percentage accuracy of each sample using:

6. $\frac{Average\ Sample\ Optimum}{Average\ Optimum} \times 100$

7. Plot the results (Figure 2:19).

Sample Size	1500	1300	1100	900	700	500	300	100
% Accuracy	100.00	99.29	98.97	98.37	97.13	97.00	94.18	84.84

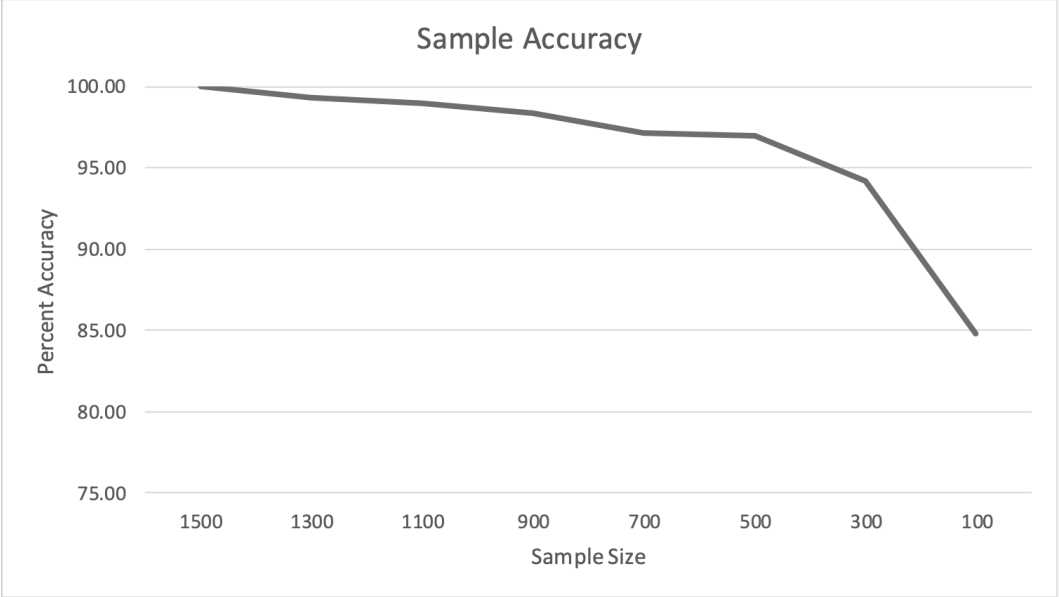


Figure 2:19 Average Sample Optimum accuracy.

There is a decrease in accuracy as the sample size decreases. From a sample size of 1500 to 100 the average sample optimum accuracy drops 15.16% from 100% to 84.84%. This decline in accuracy as sample size decreases is true for both prediction accuracy and sample accuracy. This raises the question of what the minimum acceptable accuracy is for parametric analysis in architecture. We examine this question below.

Liu et al. (2018) used deep artificial neural networks to predict long term luminance maps from High Dynamic Range (HDR) images of existing environments. They reported a prediction accuracy of 93% using 200 input images. Zhou & Liu (2015) used machine learning to predict the daylight and thermal metrics for window glazing transmittance, weather condition and blind reflectance values. For daylighting metrics, they reported accuracies ranging from 83.65% to 96.35% using artificial neural networks and 62.15% to 73.88% using support vector machines.

Ostergard et al. (2018) investigated six metamodeling techniques for building performance analysis including machine learning methods (RF, SVM and ANN). Using the coefficient of determination (R^{2}) as their accuracy metric they reported accuracies ranging from 60% with sample size of 32 to 99% with sample size above 2048. Amasyali & Gohary (2018) reviewed data-driven building energy consumption prediction models. They reported several different accuracy metrics in their study. Looking in particular at the Mean Absolute Percentage Error (MAPE) as a reference, they reported accuracies ranging from about 82% to 99% for both SVMs and ANNs.

In view of the literature above this study proposes an accuracy scale ranging from 90% (10% RAE) on the low end of the scale to 95% (5% RAE) on the high end. The research question of this study asks can machine learning algorithms be *effective* at resolving the data analytical challenges of parametric analysis. The idea of effectiveness combines speed and accuracy considerations. We define a machine learning algorithm as

effective for performing parametric analysis in architectural practice if it returns acceptable prediction accuracy (5 – 10% RAE) on a reduced design space (sample size of 1000 instances or less) that guarantees acceptable speed (33 hours or less of simulation).

2.3.4 Sampling Ratio

If a design space reduction strategy based on sampling is used, then the size of the sample will affect the prediction accuracy of the reduction. For a given design space size the larger the sample size the more accurate the design space reduction. Moreover, it is also the case that for a given sample size the smaller the design space size the more accurate the design space reduction. It is therefore not simply the design space size or the sample size that affect the accuracy of the design space reduction strategy. Rather the *sampling ratio* of the sample size to the design space size is also an important consideration.

To see the importance of the sampling ratio we consider two datasets based on the Residential Tower reference project. One of the datasets has a sample size of 7,5128,750 instances while the other has a design space size of 10,369 instances. In addition, two samples are taken from each dataset with sample sizes of 100 instances and 1,000 instances respectively. All four samples are used to train machine learning algorithms with Site Energy and Illuminance as the training classes. Figure 2:20 and Figure 2:21 as well as Table 2:8 and Table 2:9 show the results for different machine learning algorithms and different objectives.

Both the design space size and the sample size contribute to the RAE. Looking at the best results we observe that a 1:10 sampling can achieve a result with less than 10% error (MLP, Energy, 10k design space size, 1k sample size) while a 1:100 sampling is just outside this threshold using the default calibration of the machine learning algorithms (MLP, Energy, 10k design space size, 100 sample size). This suggests that with some model tuning the 1:100 ratio may achieve less than 10% error. Sampling ratios of 1:1000 and greater are likely to give greater than 20% error. We therefore recommend sampling ratios between 1:10 and 1:100 for this study.

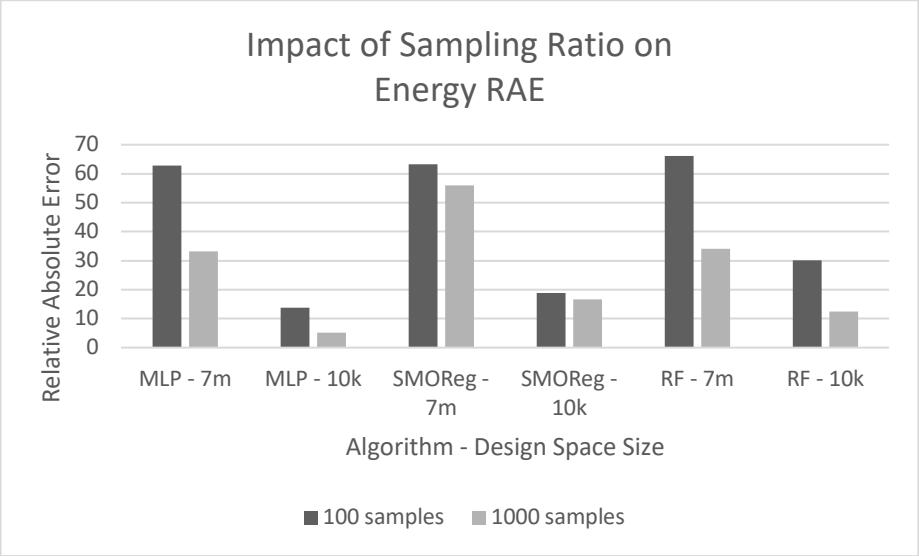


Figure 2:20 Impact of Sampling Ratio on Energy.

DESIGN SPACE SIZE	SAMPLE SIZE	ALGORITHM	RAE	CORREL. COEFF.
7,518,750	100	MLP	62.91	0.81
		SMOReg	63.19	0.72
		RF	66.16	0.8
	1000	MLP	33.12	0.94
		SMOReg	55.96	0.79
		RF	34	0.94
10,369	100	MLP	13.84	0.97
		SMOReg	18.78	0.98
		RF	30.14	0.96
	1000	MLP	5.26	0.99
		SMOReg	16.6	0.98
		RF	12.34	0.99

Table 2:8 Impact of Sampling Ratio on Energy.

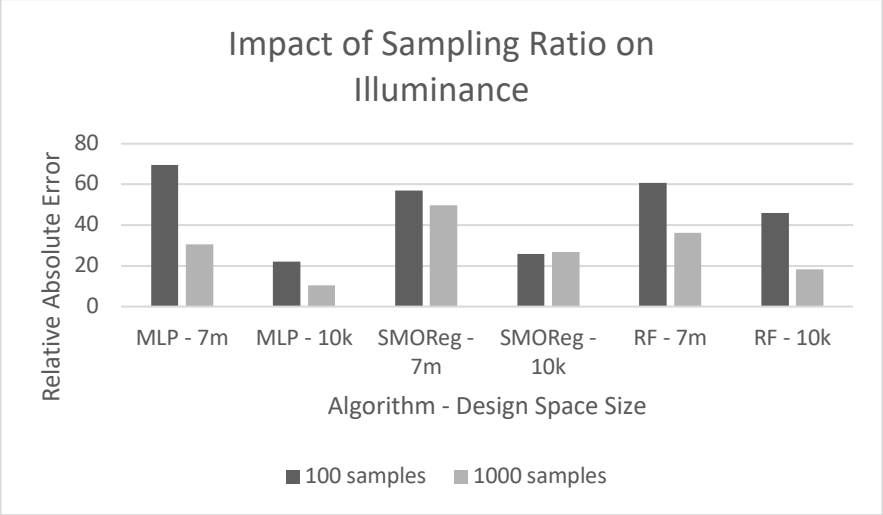


Figure 2:21 Impact of Sampling Ratio for Illuminance.

DESIGN SPACE SIZE	SAMPLE SIZE	ALGORITHM	RAE	CORREL. COEFF.
7,518,750	100	MLP	62.91	0.81
		SMOReg	63.19	0.72
		RF	66.16	0.8
	1000	MLP	33.12	0.94
		SMOReg	55.96	0.79
		RF	34	0.94
10,369	100	MLP	13.84	0.97
		SMOReg	18.78	0.98
		RF	30.14	0.96
	1000	MLP	5.26	0.99
		SMOReg	16.6	0.98
		RF	12.34	0.99

Table 2:9 Impact of Sampling Ratio for Illuminance.

The use of sampling ratio as a guiding metric raises a question. What happens if a design space needs to be larger than the recommended sampling ratio can allow? We cannot simply increase the sample size because at some point it will become impractical to simulate all the instances. For example, if a design space has 10,000,000 instances and the recommended sampling ratio is 1:100 then this would ask for a sample size of

100,000 instances. This would take more than four months to simulate using generally accessible high specification hardware.

To deal with this we explored a stratified approach to design space reduction. If the initial design space is very large low fidelity heuristics should be used to reduce the design space size to within range of the sampling ratio before high fidelity methods like machine learning are introduced. To see this process in action we describe the design space reduction for *Boys' School* (Figure 2:22) a Perkins&Will project where stratified design space reduction was used.



Figure 2:22 (Perkins&Will, 2019) A design sketch of the Boys' School project.

The Boys' School was a design proposal for a high school campus of 775 students in grades 8 through 12. The design team sought to apply the DSC process to optimize the envelope of one of the buildings on the proposed campus – the Arts and Sciences Building. The designers proposed a prefabricated panel façade concept and identified ten different possible panels for the design (Figure 2:23). The building had 3 levels and about 200 possible panel locations giving 10^{200} possible façade panel combinations. This number is not computable.

Due to the extreme size of the design space a stratified design space reduction process was needed. First, designer judgement was used to reduce the design space to about 288 million instances. Second, a zone-by-zone analysis process was used to reduce the design space to 13,824 instances. Third, a DOE was used to further reduce the design space to 288 instances which were then comfortably simulated.

As an example of designer judgement, it was recognized that any room (like a lab) that needed benches fixed to its walls could not have full length glazed panels if the exterior walls were to be useful. Therefore, only partial length glazed panels were selected at those façade locations. Using these kinds of judgements reduced the design space to 288,000,000 instances as shown in Figure 2:24.

The project sought to determine whether it was possible to achieve Passive House (Passive House Institute, 2019) standards (specifically 15 kWh/m^2/year for heating and cooling loads). As such, energy analysis was more important to this stage of the project than daylight analysis. The energy modeler recommended a zone-by-zone energy only analysis as a filtering mechanism for the overall design space. A 14-zone slice of the project was created (Figure 2:25) and 32 energy simulations per zone were executed. Since energy simulations last for about 30 seconds the overall duration was a little under 4 hours. The zone-by-zone analysis was used to identify a reduced number of panel types on each façade for multi-discipline analysis (Figure 2:26). This reduced the design space to 13,824 instances.

Lastly, a DOE was used to further reduce the design space to a manageable 288 simulations. Comparatively, this is the stage at which machine learning algorithms can be introduced into the process. An experiment

was designed in JMP software using the design space identified through zone-by-zone analysis (Figure 2:27). The design randomly sampled 32 instances from the design space, and these were simulated for both energy and daylighting and a value function calculated. The value function results were fed back into JMP to activate a regression process. From the results of the regression the Prediction Profiler feature was used to identify estimated optimum values for all input parameters (Figure 2:28). Using these estimates a final design space of 288 instances was established and simulated (Figure 2:29).

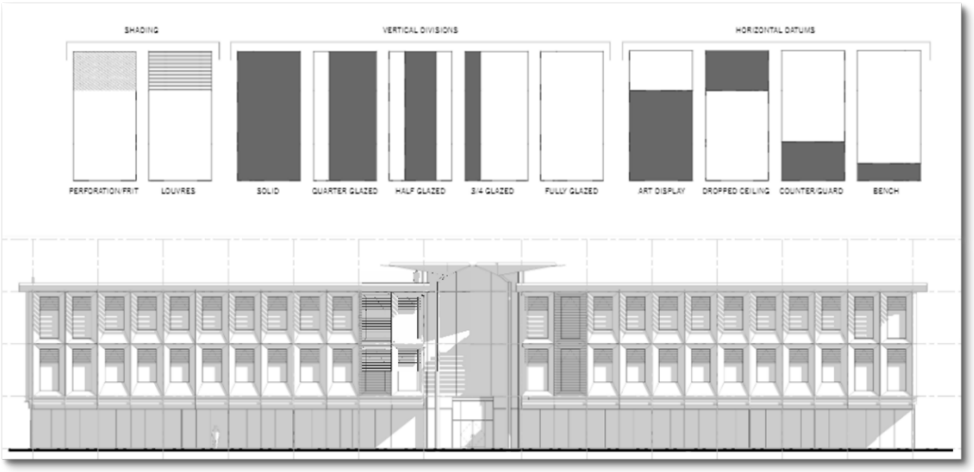


Figure 2:23 (Perkins&Will, 2019) Panel combinations for the façade of the Boys' School project.

AREAS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Description	Science Labs	Social/ learning	Atrium (S)	Circulation	Art (S)	Atrium (N)	Mechanical	Learning	Learning	Faculty	Stairs	Art (N)	Admin	End	GF End	Atrium (E/W)
Target Glazing Range	0.24-0.54	0.24-0.87	0.7-0.9	0.75-1	0.75-1	0.4-0.87	0-0.25	0.24-0.54	0.24-0.54	0.24-0.54	0-0.41	0.14-0.41	0.24-0.54	0-0.25	0.54-0.87	0.54-0.87
Panel Types	B, C,E,G	A,B,C,D,G	A, B	A,B,C,D,G	A,B,C,D,G	A,B,C,D,	E,F,H	C,D,E,F,G	C,D,E,F,G	C,D,E,F,G	F,G,H	F,G,	C,E,F,G	F,H	A,B	A, B
Shading	Y (1,2,3)	Y (1,2,3)	Y (1,2,3)	Y (1,2,3)	Y (1,2,3)	N	N	N	N	N	N	N	N	N	Y (1,4,5)	Y (1,4,5)
SHGC	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3

Figure 2:24 Boys' School design space reduction by designer judgement.

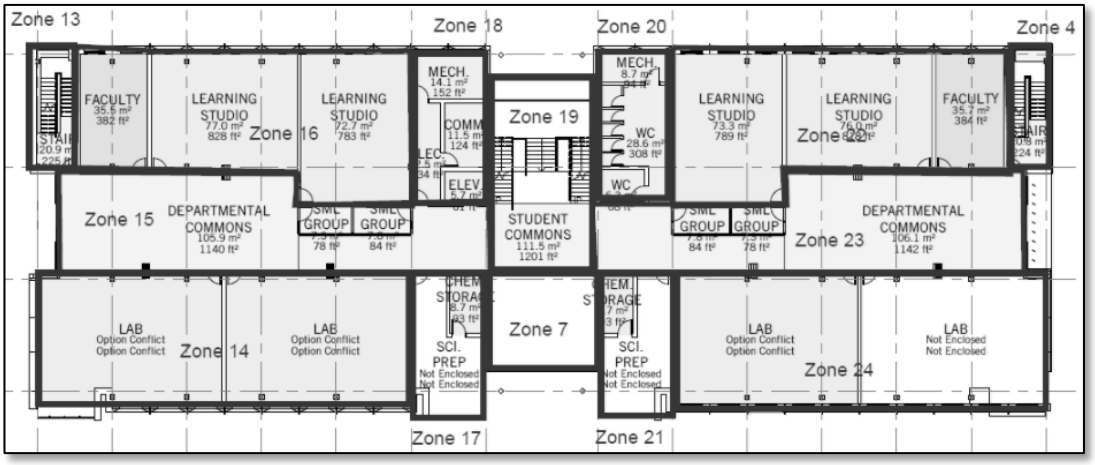


Figure 2:25 (Perkins&Will, 2019) Boys' School typical floor zoning.

	South W	Panel	West W	Panel	East W	Panel	North W	Panel	Wall R	Window U	SHGC	Shading
Zone 14	40%	CE	20%	F	0%		0%		6,7	0.65	0.4	1
Zone 17	40%	CD	0%		0%		0%		6	0.7	0.3	1
Zone 7	60%	B	0%		0%		0%		7	0.65	0.3	1
Zonn 21	40%	CD	0%		0%		0%		7	0.65	0.3	1
Zone24	40%	CE	0%		20%	F	0%		6,7	0.65-0.7	0.4	1
Zone 23	0%		0%		60%	B	0%		6,7	0.65,0.7	0.3	1
Zone 4	0%	`	0%		`		20%	G	7	0.65,0.7	0.5	0
Zone 22	0%		0%		0%		40%	CE	6,7	0.65,0.7	0.3	0
Zone 20	0%		0%		0%		20%	E	7	0.65	0.4	0
Zone 19	0%		0%		0%		40%	C,D	6,7	0.65,0.7	0.3	0
Zone 18	0%		0%		0%		20%	E	7	0.65,0.7	0.5	0
Zone 16	0%		0%		0%		40%	CE	6,7	0.65,0.7	0.3	0
Zone 13	0%		0%		0%		20%	G	7	0.65	0.5	0
Zone 15	0%		0%		60%	B	0%		7	0.7	0.3	0, 0.5, 1

Figure 2:26 Boys' School design space reduction through zone-by-zone analysis.

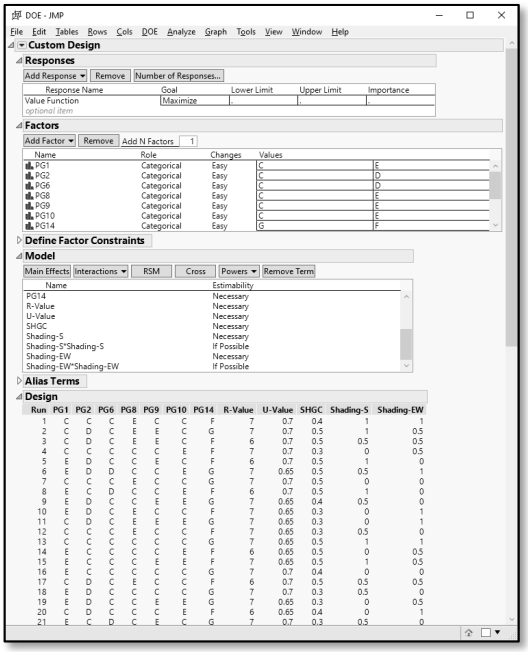


Figure 2:27 Boys' School design of experiments.

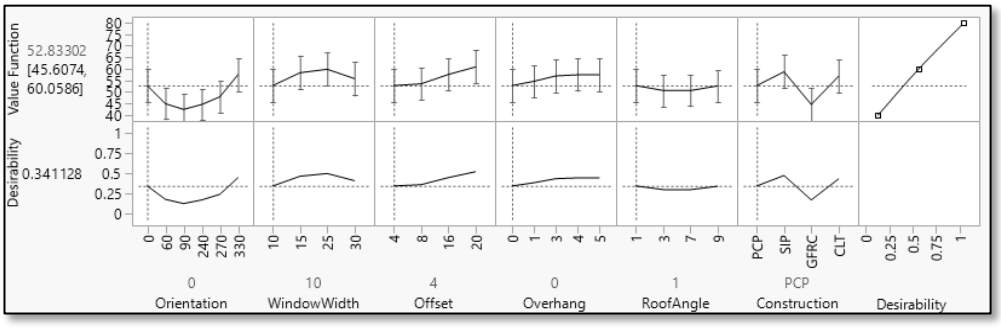


Figure 2:28 Boys' School Prediction Profiler.

Inputs	PG1	PG2	PG3	PG6	PG7	PG8	PG9	PG10	PG11	PG14	PG16	R-Value	U-Value	SHGC	Shading-N	Shading-S	Shading -EW
Options	C,E	C,D	B	C,D	E	C,E	C,E	C,E	G	G,F	B	6,7	0.65,0.7	0.3,0.4,0.5	0	0,0.5,1 - H	0,0.5,1 - V
Number	2	2	1	2	1	2	2	2	1	2	1	2	2	3	1	3	3
Full Factorial	C	C,D	B	C,D	E	C,E	C,E	C,E	G	F	B	7	0.65	0.3,0.4,0.5	0	1	0,0.5,1 - V
Number	1	2	1	2	1	2	2	2	1	1	1	1	1	3	1	1	3

Figure 2:29 Boys' School final design space definition of 288 instances.

2.3.5 Design Space Complexity

When performing conventional energy analysis on real world projects many analysis tools perform full building energy analysis. This means they simulate energy exchanges between all the zones defined in the analytical model. This can be a time-consuming exercise if there are a large number of zones. When performing parametric analysis, it is therefore desirable to simplify the scope of simulation. One way to do this is to use the idea of slices. Many buildings are repetitive from level to level. This means that rather than simulate the full building with all its zones we can instead simulate a typical level, or slice, and then obtain the full building performance by multiplying the typical level performance by the number of levels.

For example, on the Residential Tower project, many of the tower levels were typical. Therefore, in order to perform the parametric analysis a typical level was taken as a slice and analyzed (Figure 2:30). Developing the machine learning training model for such a slice is straightforward. The model consists of the input parameters of the slice's analytical model with the output parameter as the training label.

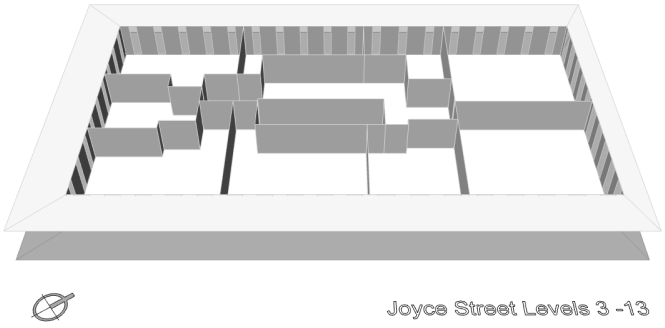


Figure 2:30 Residential Tower, slice of typical level.

Many projects, however, cannot make direct use of the slice simplification. For example, the Hotel Residence project has two towers with different occupancies. This means, at minimum, that parametric analysis for the Hotel Residence needs two separate simulations corresponding to one slice from each tower. In reality, due to different occupancies within the hotel itself (residential and podium) three slices were chosen (Figure 2:31).

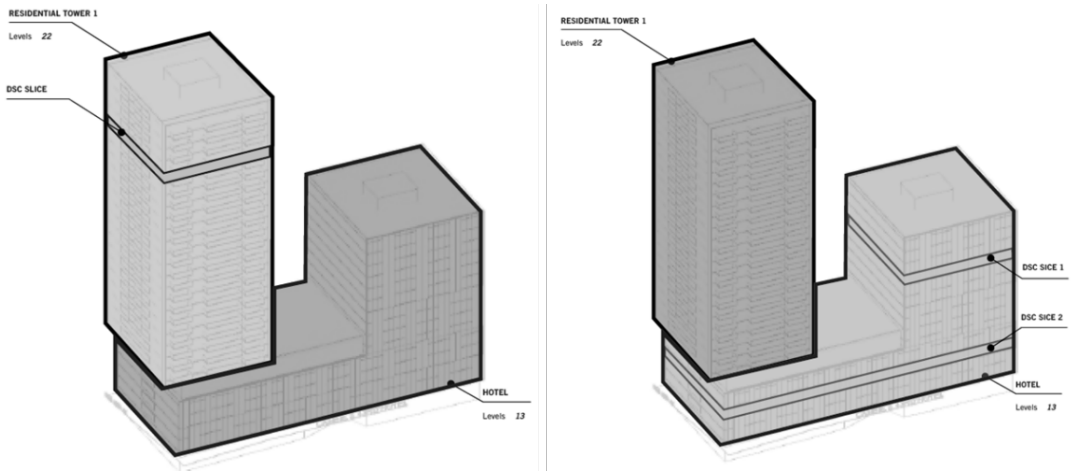


Figure 2:31 Hotel Residence, slices of typical levels.

How does the need for several simulation slices impact the parametric analysis and machine learning process? If we create a different machine learning model for each slice then we need $n \times$ *sample size* simulations to predict the full design space, where n is the number of slices. This is not desirable for simulation speed reasons, particularly if n is large. Unless parallel processing is used it will take n times as long to perform simulation.

Can we train one slice from the project and then use the machine learning model from this slice to make predictions on all other slices? This would give acceptable simulation speeds but how accurate would the resulting predictions be? Alternatively, can we use a few instances from each slice to create a hybrid training set of the recommended sample size and then use this to create a machine learning model? Once again, how accurate would the predictions from this model be? In other words, how robust is the machine learning model to these variations in the training dataset sample?

This is the design space complexity problem and is one of the research questions presented in Section 2.4. Design space complexity refers to variations in design conditions that prevent the development of simple data models for parametric analysis. Based on the reference projects we have introduced in this chapter we can identify four sources of design space complexity (Figure 2:32). The Residential Tower represents the simple model situation where there are no variations to take into account. For the design space complexity question, we will consider the Residential Tower as a baseline against which performance from complex design spaces will be compared.

1. Complex adjacency as seen on Balcony Tower. Complex adjacency conditions mean that the model has at least one surface on one slice, relevant for parametric analysis, that is adjacent to an internal or external condition that is significantly different from other slices.

2. Complex geometry as seen on Mid-Rise Towers. Complex geometry conditions mean that the model has at least one slice, relevant for parametric analysis, whose external or internal geometry is significantly different from other slices.

3. Complex occupancy as seen on Hotel Residence. Complex occupancy conditions mean that the model has at least one slice, relevant for parametric analysis, which has a significant occupancy different from other slices.

4. Design options as seen on Students' Residence. Design option complexity means that the design has distinct models with possible differences in geometry, program, occupancy and materials but which can nonetheless share one data analytical model.

In this study we will examine the design complexity question based on complex geometric and adjacency conditions (Mid-Rise Towers) as well as design option complexity (Students' Residence). We will take the Residential Tower simple model as a base line and compare base line accuracy to accuracies under complex design conditions. We will use one-way ANOVA and Tukey's HSD post processing to detect any significant differences at 90%-95% confidence levels.

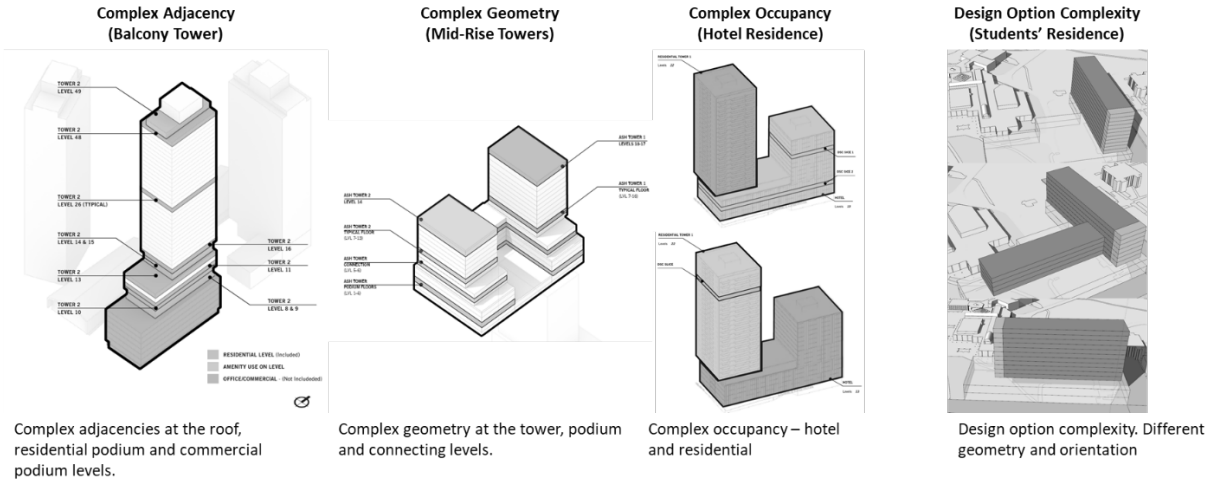


Figure 2:32 Sources of Design Space Complexity.

2.4 Research Questions

We have now demonstrated some of the limitations of conventional building performance analysis approaches and discussed some of the ways parametric analysis can improve on these limitations. We have discussed the importance of parametric analysis but acknowledged that practical implementation faces data analytical challenges. These challenges arise from the fact that the potential size of project design spaces can be prohibitive to simulate, and design space complexity can be difficult to model analytically. The primary hypothesis of this research is that machine learning algorithms can be effective in resolving these challenges. Accordingly, the primary research question of the study can be formulated as follows:

Can machine learning algorithms perform effectively at design space reduction of multi-discipline design spaces associated with parametric analysis in architectural practice?

To facilitate the experimental process, we develop three sub-questions to support the main research question:

1. Which machine learning algorithms perform most effectively at multi-discipline parametric analysis?

2. For each algorithm, what is the impact of sample size, sensitivity analysis, feature selection and hyperparameter tuning on performance?

3. Can machine learning algorithms perform effectively when used under complex design space conditions?

By the term 'effective' we are defining constraints of real-world practice that we wish to impose on the machine learning process. In this study the constraints are based on speed, accuracy and sample size considerations. The specific values of the constraints are a speed of 33 hours, assuming 2 minutes per simulation on generally accessible hardware; 90 – 95% accuracy (or 5 – 10% RAE), and a sample size of 1000 simulations. We should also ensure the design space size is within the recommended sampling ratio of 1:100.

The 33-hour constraint is based on a worst-case scenario where the designers need to respond to design changes overnight. The 2 minute per simulation constraint is based on measurement of the speed at which the hardware indicated in Section 2.3.2 can perform a multi-discipline simulation (energy and daylighting). 90 – 95% accuracy is based on research literature (section 2.3.3). The sample size of 1000 is calculated based on 33 hours at 2 minutes per simulation.

Specific values of the constraints can vary from study to study. For example, if the experiment is using more powerful hardware, we can increase the sample size constraint or reduce the speed constraint; alternatively, if we relax the accuracy constraint, we can reduce the speed or sample size constraints. However, the methodology proposed in this study should be capable of assessing the effectiveness of machine learning algorithms regardless of the specific values assigned to the constraints.

In addition to examining the primary research question the experiments will also attempt to address the challenges of design space complexity. The Residential Tower will be used as a baseline case of simple model complexity. The Mid-Rise Towers will be used to investigate if machine learning algorithms can be effective under complex geometric and adjacency conditions. The Students' Residence will investigate if machine learning algorithms can be effective under design option complexity.

The study seeks to build on existing research, both from architecture (such as Haymaker, et al., 2018) as well as building engineering (such as Magoules & Zhao, 2016), by asking research questions in the specific context of architectural practice. The study is limited in scope to three machine learning algorithms identified through literature as commonly used in building performance analysis namely ANN (MLP implementation), SVM (SMOReg implementation) and DT (RF implementation).

We expect that the outcome of this study will provide practicing architects with an understanding of both the value and the data analytical challenges involved with multi-discipline parametric analysis. We also expect that the study will provide insights into strategies for mitigating these challenges using machine learning algorithms.

Chapter 3 Literature Review

This chapter reviews literature related to parametric analysis in architecture. We begin by discussing the causes of high cardinality design spaces and the need for design space reduction. We make the case for machine learning as a design space reduction strategy. We then review research related to parametric analysis in architecture. We argue that it has not yet thoroughly investigated machine learning for multi-discipline design space reduction.

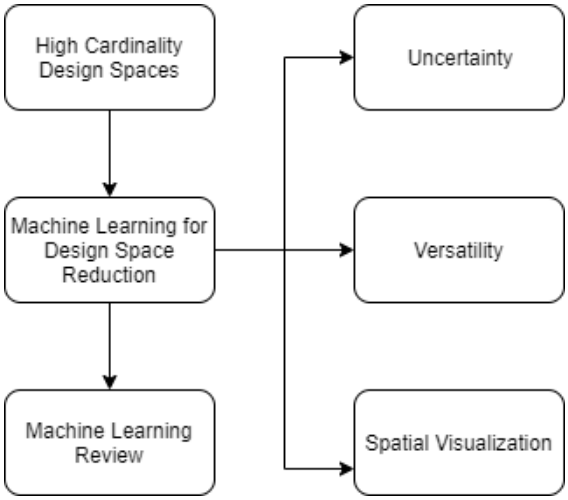


Figure 3:1 Diagram Summary of Chapter Three.

3.1 Introduction

This literature review explains the causes of high cardinality design spaces, makes the case for machine learning as a design space reduction method, and argues that research into parametric analysis in architecture has not sufficiently investigated machine learning for design space reduction. We identify the causes of high cardinality design spaces as 'combinatorial explosion' and the 'curse of dimensionality'. We make the argument that machine learning is suitable for design space reduction in architecture because it can model uncertainty, is more versatile and supports spatial visualization of simulated data. We reveal shortcomings in the literature on parametric analysis in architecture and argue that this study contributes by investigating machine learning in architectural practice under real world constraints.

The parametric analysis discussed in this study is a multi-discipline (and therefore multi-objective) process. In particular, the process involves both energy analysis and daylight analysis performed simultaneously. Architectural design is holistic, requiring the synchronous balancing of different, and sometimes conflicting, design considerations such as cost, function, performance and aesthetics. Architectural decision making is more effective when these constraints are assessed together rather than when each is assessed in isolation.

Since parametric analysis involves simulation, multi-discipline parametric analysis requires that distinct simulations be integrated and executed simultaneously. The DSC framework facilitates this multi-discipline integration. However, the integration gives rise to design spaces of high cardinality.

In computational disciplines the high cardinality of design spaces arises due to the phenomena of 'combinatorial explosion' as well as the 'curse of dimensionality'. In analytical terms the size of a design space is given by the product ΠX_i where $0 < i < n$, n is the number of parameters of the design space and the X_i are the number of values of each parameter indexed by i. This product is large if the X_i are large, in which case we have a combinatorial explosion; or else, if n is large, the combinatorial explosion is precipitated by the curse of dimensionality.

Number of Parameters		Size of Design Space
1	10	10
2	10x10	100
3	10x10x10	1000
4	10x10x10x10	10000
5	10x10x10x10x10	100000
6	10x10x10x10x10x10	1000000
7	10x10x10x10x10x10X10	10000000
8	10x10x10x10x10x10X10x10	100000000
9	10x10x10x10x10x10X10x10x10	1000000000
10	10x10x10x10x10x10X10x10x10x10	10000000000

Figure 3:2 Combinatorial Explosion. Each parameter has a value range of 10.

Glover & Greenberg (1989) describe combinatorial explosion as being encountered when alternatives are sequentially compounded leading to a vast mushrooming of possibilities. They describe the challenge using the Travelling Salesman Problem. The salesman must depart from his home city, visit every city on his route exactly once and then return home. It is assumed that the cost of moving from one city to another is the same for all cities enroute. The challenge is to make his journey at the lowest cost possible. If there are *n* cities on the route, then there are *n!* possible routes which is effectively an exponential problem and therefore computationally infeasible. This means that the Travelling Salesman cannot be solved by exhaustive brute force search. The problem space is too large.

Keogh & Mueen (2017) describe the curse of dimensionality as the problem caused by the exponential increase in volume associated with adding extra dimensions to Euclidean space. For example, if a line is divided into 99 equal parts then 100 numbers are required to specify each point on the line. If this line is used to create a two-dimensional grid, then 100^2 numbers are needed to specify each point on the grid. If the line is used to create a three-dimensional cube, then 100^3 numbers are needed to specify each point in the cube. In general, 100^n numbers are needed to specify the points of an *n*-dimensional hypercube and this represents exponential growth.

Multi-discipline design spaces will require many more input parameters to define than single discipline design spaces. For example, input parameters for energy analysis include WWRs, R-values of envelope materials, SHGC of glazing materials and dimensions of shading devices. Input parameters for daylight analysis include WWRs, visual transmittance values for glazing, reflectance and absorption properties of material surfaces, as well as radiosity and raytracing settings.

Clearly the integration of these two disciplines into a single multi-discipline analysis will generate a design space with a large number of input parameters. In addition, the side by side consideration of multiple disciplines is more than likely to increase the ranges of input parameters. For example, while daylight analysis on its own may explore high value ranges of WWR to maximize natural lighting, considerations of energy performance will require that lower values of WWR to be explored as well. The result is an expansion of the value ranges of the WWR input parameter.

The combined effect of a large number of input parameters and large input parameter value ranges gives rise to a high cardinality design space. The large number of input parameters corresponds to the curse of dimensionality while the large value ranges of input parameters exacerbates the combinatorial explosion. High cardinality design spaces are in general not computable and design space reduction is required. However, design space reduction cannot be performed at too great an expense in terms of prediction accuracy. Moreover, the design space reduction must consider the speed of simulations based on available hardware. This will set an upper threshold for the number of simulations that are feasible, which in turn affects the accuracy of design space reduction models.

3.2 Machine Learning for Design Space Reduction

It has been argued that, for practical reasons, parametric analysis requires design space reduction. But why choose machine learning as a design space reduction strategy? We have mentioned that solving the problem of high cardinality design spaces can be approached through hardware-based methods or software-based methods (Figure 3:3). Software-based methods can be classified into three broad approaches: algorithmic, statistical and machine learning methods. Algorithmic methods include genetic algorithms and other evolutionary algorithm methods. Statistical methods include regression and design of experiments methods. Machine learning methods include Decision Trees, SVMs and ANNs.

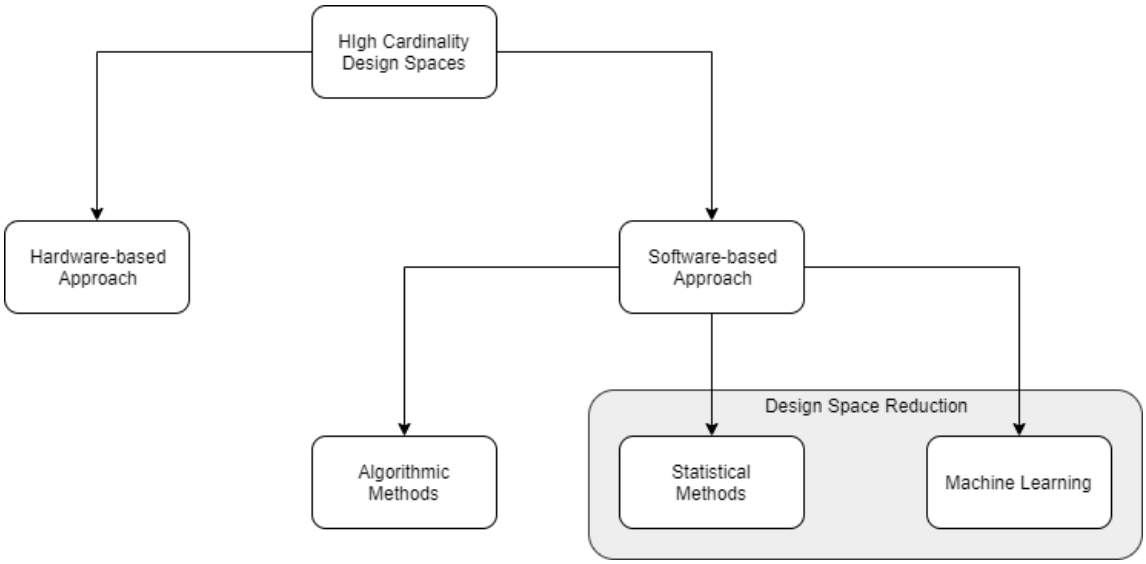


Figure 3:3 Approaches to solving high cardinality design spaces.

Algorithmic methods solve high-cardinality design spaces through optimized search strategies. Delgarm et al. (2016) used multi-objective particle swarm optimization algorithms to find a set of optimal solutions for a building energy performance problem. Ascione et al. (2015) evaluated cost-optimality of energy efficiency measures by means of a multi-objective optimization using genetic algorithms. Echenagucia et al. (2015) used genetic algorithms to perform a multi-objective search aimed at minimizing the energy need for heating, cooling and lighting.

Statistical methods, on the other hand, solve high-cardinality design spaces by developing data models based on small samples that can then be used to make predictions on the larger design space. Hygh et al. (2012) used a multi-variate linear regression model to assess the energy performance of a medium sized building in four climate zones. Dhariwal & Banerjee (2017) proposed a DOE to reduce simulation time on a three-story office building. Sadeghifam et al. (2015) used a DOE to optimize the selection of building envelope materials for improved thermal performance.

Machine learning methods also use the predictive data modeling approach. Wei et al. (2018) reviewed data-driven approaches for prediction and classification of building energy consumption. Amasyali & Gohari (2018) reviewed data-driven building energy prediction methods including ANNs, SVMs and Decision Trees. Ostergard et al. (2018) compared several design space reduction techniques applied to building performance simulations.

Although all three approaches (algorithmic, statistical and machine learning) provide promising outcomes, three reasons are given for preferring machine learning to the other methods. First, machine learning algorithms can model uncertainty while algorithmic methods typically do not. Machine learning algorithms report metrics associated with uncertainty such as accuracy, confusion matrices, kappa statistics, co-efficient

of determination, among others. Algorithmic methods, such as genetic algorithms, will return optimized values without giving any indication of the confidence levels in the results.

Second, machine learning algorithms are more versatile than other statistical methods. Statistical methods focus mostly on simple forms of regression, whereas machine learning methods provide complex regression and classification analysis tools with computationally efficient implementations.

Third, machine learning supports computer vision methods for graphically visualizing simulation data. In many cases this is preferable to reporting numerical data because architecture is a visual profession. In cases such as daylighting analysis it is imperative to have visual reporting of simulation results so the architect can understand exactly where in the design geometry design issues like glare or inadequate lighting are occurring. It is not enough to simply report a numerical daylighting metric as this does not adequately inform the spatial context of the problem.

Liu et al. (2018) developed a workflow for generating annual luminance maps of indoor space from training data using ANNs. In particular they sought to demonstrate the use of deep neural networks, a form of ANN used for deep learning, to predict long term luminance maps from a small training set. The trained ANN predicted annual luminance maps from High Dynamic Range (HDR) renderings of existing environments without time consuming simulation.

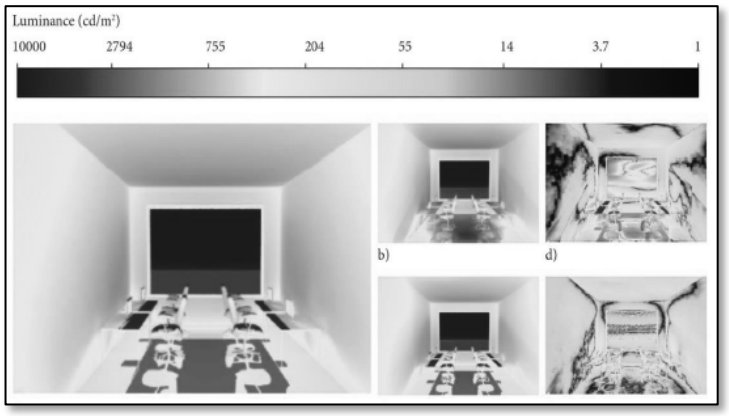


Figure 3:4 Liu et al. (2018) Luminance maps predicted by ANN algorithms.

3.3 Machine Learning Review

In this section we review research where machine learning has been applied to design space reduction for building performance analysis. We review research related to machine learning for energy analysis, daylighting and multi-objective optimization. We progressively build the argument that current architectural research has not sufficiently investigated the application of machine learning for multi-discipline (and therefore multi-objective) building performance analysis (Table 3:1).

Author	Year	Energy	Daylighting	Multi-Objective	Multi-Discipline	Machine Learning	Pareto or Value Function
Magoules & Zhao	2016	●				●	
Robinson et al.	2017	●				●	
Kazanasmaz et al.	2009		●			●	
Lorenz & Jabi	2017		●			●	
Diakaki et al.	2008	●		●			
Murray et al.	2014	●		●			
Nielsen et al.	2011	●	●	●	●		
Ahmad et al.	2017	●	●	●	●	●	
Lartigue et al.	2014	●	●	●	●		●
Fang	2017	●	●	●	●		●
Zhou & Lin	2015	●	●	●	●	●	
Okhoya	**2019**	●	●	●	●	●	●

Table 3:1 Comparative summary of literature review sources.

Magoules & Zhao (2016) identified different models used for building energy analysis. Physical models use thermal dynamics and physical principles to calculate performance for the whole building or its components. Gray models are used when only partial information on the design is available. Statistical models mainly use regression approaches to correlate the building performance metrics to input variables. They identify SVMs and ANNs as the most widely used artificial intelligence models for building performance analysis.

They describe different methods of data collection including surveys and questionnaires, measurements and simulations. Simulation is less expensive than surveys or measurements and can analyze complicated designs more rigorously. They identify DOE-2™ (James J. Hirsch & Associates., 2019), TRNSYS™ (TRNSYS, 2019) and EnergyPlus as the most common building energy simulation tools. They describe the use of SVMs for predicting energy consumption as well as the use of ANNs to perform faulty consumption detection and diagnosis.

Robinson et al. (2017) aimed to model commercial building energy consumption at the building level using machine learning models. They trained machine learning models using the 2012 Commercial Building Energy Consumption Survey (CBECS) data which had 6,720 instances. Due to the difficulty of data gathering their machine learning models used a limited set of building level features namely square footage, activity, number of floors, and heating and cooling degree days.

They used a range of models for their experiments including linear regression models, support vector machines and boosted decision tree models. They validated their model using the New York City Local Law 84 energy consumption dataset which had 13, 223 instances. They found that gradient boosted regression models performed best with a coefficient of determination (R^2) of 0.82. They applied their findings to 73,388 commercial buildings in Atlanta, Georgia.

These authors show the viability of machine learning for energy analysis predictions. On the other hand Kazanasmaz et al. (2009) observed that in order to design good daylight in buildings designers need to determine the design illuminance levels at strategic points. They recognized three techniques for estimating daylight illuminance levels – physical models, computer simulations and analytical formulas. Physical models, however, are found to have discrepancies from reality arising from a mismatch between model properties like surface reflectance and real-world surfaces. This has made computer simulations more desirable. The authors proposed a computer simulation-based approach using ANNs as a tool to predict daylight illuminance in office buildings.

The subject building for the research was an academic administrative building located in Izmir, Turkey. Data was obtained from field measurements using light sensors located on a grid within the building. Analysis

inputs were field measurements, building parameters and weather parameters. The input variables were Date, Hour, Outdoor Temperature, Solar Radiation, Humidity, UV index, UV dose, Distance to window, Number of windows, Orientation, Floor Level, Room Aspect Ratio and Sensor Point, while the output variable was Illuminance. They reported 97.80% accuracy levels.

Lorenz & Jabi (2017) proposed artificial neural networks for predicting daylighting performance using daylight autonomy as the climate-based metric. They noted the sparseness of research into machine learning for daylighting predictions. They also noted the need to clearly identify impactful input parameters and to empirically search for neural network architecture that best captures dynamic daylighting analysis interrelationships.

They used a generic typology of the ground floor of an office building and performed daylight autonomy simulation using Diva for Rhinoceros™ (Solemna, 2019) for analysis and Grasshopper for parametric modeling. The design variables of the experiment were the *x, y* coordinates of sensor locations, average distance of sensor points to the center of windows and overall dimensions of windows and walls as well as their respective orientations.

They reported that ANNs were an accurate method for solving the complex and non-linear dynamics of daylight simulation. For single room analysis they found a 12 neuron, single hidden layer architecture performed best. A noticeably lower sensor error rate was reported for rooms with two orientation (corner rooms) but sensor error gradually increased towards the rear of rooms with one dimension. This suggests that the more available light (more data) the more accurate the prediction. They, however, observed that due to the expense of gathering simulated data for training the utility of this method depends on the ratio of the size of the training dataset compared to the full design space.

While all the above authors indicate that machine learning is a viable approach for both energy analysis and daylighting, in all cases they consider single objective analysis. Architectural decision making on the other hand is inherently multi-objective often seeking to balance between conflicting design criteria. Diakaki et al. (2008) recognized multi-objective analysis as an established decision-making support tool for energy efficient design. They identified several measures for the improvement of energy efficiency including improvement of building envelope, reducing heating and cooling loads and using renewable sources of energy. With such a variety of measures the designer must compensate for environmental, energy, financial and social factors in developing the best possible solutions.

They performed a multi-objective optimization on a simple building studying appropriate choices for window type, wall insulation material, and the thickness of the wall insulation layer. The aim was to reduce the construction cost and increase the energy savings. They applied several solution methods to the problem and concluded that multi-objective optimization for cost reduction and energy efficiency was both feasible and desirable. However, they noted the simplicity of their reference project and the potential complexity of more extended design problems.

Murray et al. (2014) presented the retrofitting of buildings for improved energy performance as an optimization problem balancing construction cost, operating cost and occupant thermal comfort. They proposed a simulation-based method which coupled thermal modeling with a genetic algorithm optimizer. They demonstrated the method on a university building located in Cork, England.

The building was initially retrofitted without any energy assessment meaning the retrofits were based on expert judgement. In order to perform the back-sighting energy assessment the researchers modelled thermal energy consumption, building material costs, fuel costs and carbon emission factors. Optimization was performed using a genetic algorithm with payback period, carbon emissions and energy cost as objective functions. The simulated retrofit reduced thermal energy consumption by 72% post-retrofit compared to 53% from the actual retrofit. In addition, the simulated retrofit identified solutions that would reduce the annual energy cost by 83%, carbon emissions by 96% and achieve the payback period of under 5 years.

Although both Diakaki and Murray demonstrated the value of multi-objective analysis their studies are primarily concerned with energy performance. From an architectural point of view this cannot be considered truly multi-disciplinary. True multi-disciplinary analysis can be achieved, for example, by simultaneously analyzing daylighting and energy. But now the means of such an analysis involving separate analytical models and analytical engines presents a uniquely challenging problem.

Nielsen et al. (2011) recognized that to gain a full picture of a dynamic façade's performance the total energy consumption and indoor environment need to be considered simultaneously. They quantified the performance of three types of facades considering energy demand, indoor air quality, amount of daylight available and visual comfort. They argued that the potential energy reduction and increase in occupant comfort can only be achieved through an integrated multi-disciplinary analysis process. For example, improving interior daylighting can reduce energy consumption from artificial lighting but it can also increase heat gain. They used a 3 x 3 x 6 m office space located in Denmark to study the effect of three different solar shading designs on façade performance. Evaluation of the effect of design parameters on energy performance and indoor environment were based on thermal energy and daylight factor considerations. They concluded that an integrated multi-disciplinary simulation can reveal a high degree of interdependence between design parameters and make it possible to quantify performance along several axes to obtain a holistic solution.

Ahmad et al. (2017) used ANNs and RFs to predict the hourly energy consumption and daylight illuminance of a classroom in Cardiff, UK. They identified numerical approaches to building performance analysis as those using simulation tools like EnergyPlus and Radiance. They identified predictive approaches as those using methods like machine learning algorithms. They argued that predictive methods were quicker than numerical methods.

They compared ANNs and RFs for energy consumption and daylight illuminance. The input variables included solar altitude angle, solar azimuth angle, outdoor dry-bulb air temperature and wind speed. The output variables were energy consumption and daylight illuminance at a sensor point. Evaluation metrics were the RMSE, R^2, CV and Mean Absolute Deviation (MAD). The methodology involved sensitivity analysis to determine which input variables most impacted the output, and the tuning of 'maximum tree depth' and 'maximum number of leaves' hyperparameters. Results indicated that the ANN performed better for energy consumption while the RF performed better for daylighting.

One shortcoming with Nielsen and Ahmad is that they do not pursue systematic methods of evaluating the competitive criteria of the multi-discipline analysis. They rely largely on intuitive methods to evaluate and explain the analysis outcomes. This informality loses precision in decision making. It is preferable to have systematic methods for evaluating the joint impacts of multiple criteria in multi-disciplinary analysis. Lartigue et al. (2014) proposed a method for optimizing the envelope of a building based on energy load and daylighting considerations. They applied the method to the retrofitting of a dormitory building in Pittsburgh, USA. The main inputs were window type parameters as well as window to wall ratio (WWR). The output parameters to be optimized were heating load, cooling load and daylight illuminance. They acknowledged the antagonistic effects of the input parameters. The larger the WWR the better the daylighting but also the larger the heating load due to increased heat loss in winter, and the higher the cooling load due to increased heat gain in winter. They needed an optimization approach to balance these competing criteria.

They used GenOpt (University of California, Berkley, 2019), an optimization program that couples with external simulation tools to compute optimization value functions. They created a design space based on variations of the input parameters and simulated it to obtain a solution set. They used a Pareto front to optimize the solution set (Figure 3:5). A Pareto front is a set of non-dominated solutions to a multi-objective optimization problem. In Figure 3:5 the black items form a Pareto front because no quantity has a higher value along *both* axes. For the gray items, which are in what is called the feasible region, all items have at least one item that has higher value on at least one axis. This means they are all dominated.

At the building scale, Fang (2017) also proposed an optimization process for the evaluation of both energy and daylight performance. She also used a genetic algorithm with a Pareto front as an optimization technique. She used an integrated daylight and energy simulation process to generate the design alternatives of her design space. This involved Rhinoceros™ (Robert McNeel, 2019), Grasshopper™ (Grasshopper, 2019), Ladybug™ (Roudsari, 2019) and Honeybee™ (Roudsari, 2019) software coupled to the Radiance simulation engine for daylight analysis and to the Energyplus simulation engine for energy analysis.

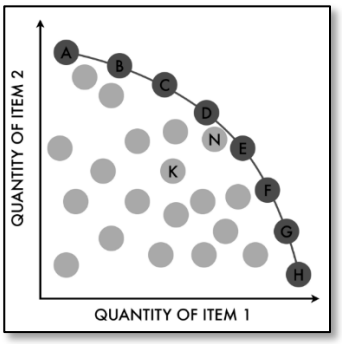


Figure 3:5 (Wikipedia, 2019) Pareto Front

She used a simple 3600 square foot office building as her case study. She optimized the case study in three different climate conditions: 2A (Orlando), 4A (Raleigh) and 6A (Minneapolis). Using the genetic algorithm, she optimized the building for energy and daylighting. Using the Pareto front she identified a balanced optimum for both energy and daylight (Figure 3:6).

While Lartgiue and Fang demonstrate the value of multi-disciplinary optimization, their methods do not involve machine learning. Zhou & Liu (2015) compared ANNs and SVMs for daylighting and thermal performance of building envelopes. Their inputs were weather data, window SHGC and window blind reflectance values. The daylight metric was useful daylight illuminance (UDI) while the energy metric was aggregated hourly energy consumption. Their algorithms involved generating building design data, using this along with weather data to perform simulations, generating daylighting and thermal performance data, feature selection and extraction, training the machine learning algorithms to make predictions, and performing comparative analysis. They also used principal component analysis (PCA) to perform sensitivity analysis and examine its impact on the performance of the algorithms. While Zhou & Liu used machine learning for both daylighting and energy analysis, they did not employ a systematic evaluation method for the joint evaluation of these potentially conflicting analyses.

In the experiments of this study we will combine strategies from the literature review. Like Zhou & Liu we will use machine learning algorithms for both energy and daylighting analyses. Since they did not perform systematic comparative analysis, like Lartigue and Fang we will adopt a systematic approach. However, unlike these latter we will not opt for a genetic algorithm with Pareto fronts. Instead we will construct a value function for the cross-comparison of the conflicting analysis criteria. This approach is more amenable to the application of machine learning algorithms.

The literature review demonstrates that research has been done into parametric analysis in architecture. However, machine learning based multi-discipline parametric analysis has not been thoroughly investigated. In this study we contribute by investigating machine learning for multi-discipline parametric analysis in the context of architectural practice. Real world projects have real world constraints and we seek to understand how effective machine learning algorithms are under such conditions. Real projects must use real data to develop predictive models with minimum simplification or idealization. Real projects must also respect project schedules and industry standards. It therefore seems reasonable to ask how effective are machine learning algorithms in these real environments?

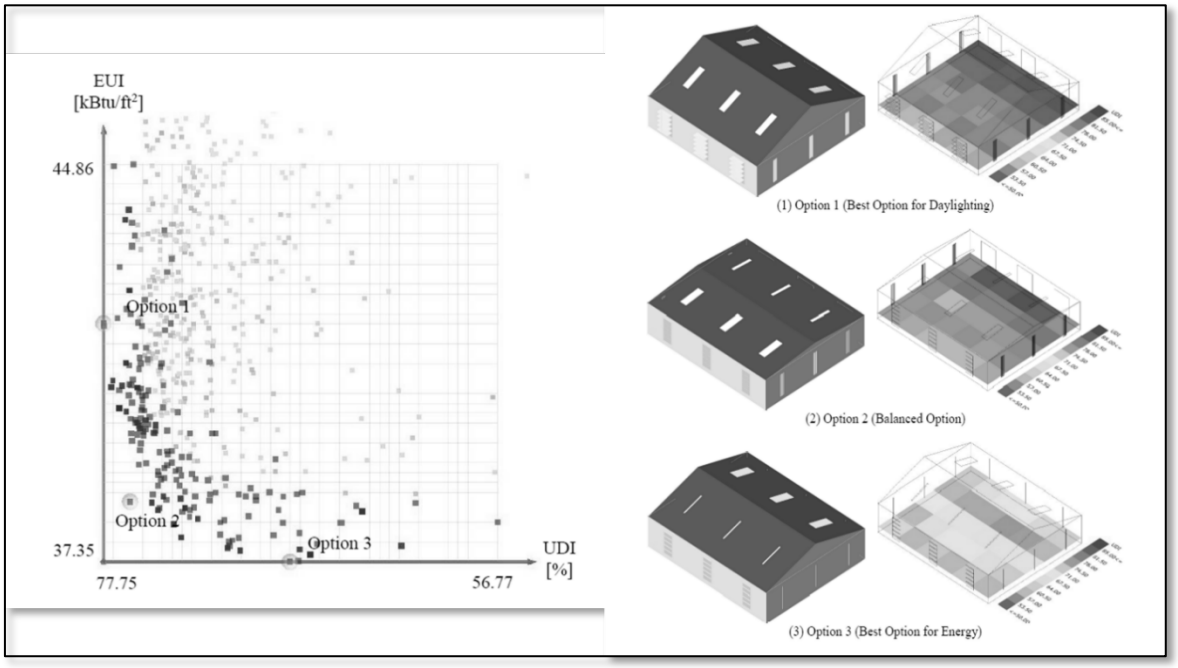


Figure 3:6 (Fang, 2017) Pareto front optimization for energy and daylighting analysis.

Chapter 4 Machine Learning Background

In this chapter we discuss the theoretical background to the machine learning algorithms used in this study. We first introduce two different types of machine learning algorithm – supervised and unsupervised learning. We focus on supervised learning for the study. We then discuss the theory of the machine learning algorithms we intend to use for our experiments– choosing to focus on SVMs, ANNs and DTs. Since building performance analysis is a regression problem, we identify regressive implementations of these algorithms that we will run – SMOReg, MLP and RF. Lastly, we discuss the impact factors that we will apply to attempt to improve algorithm performance beyond the default.

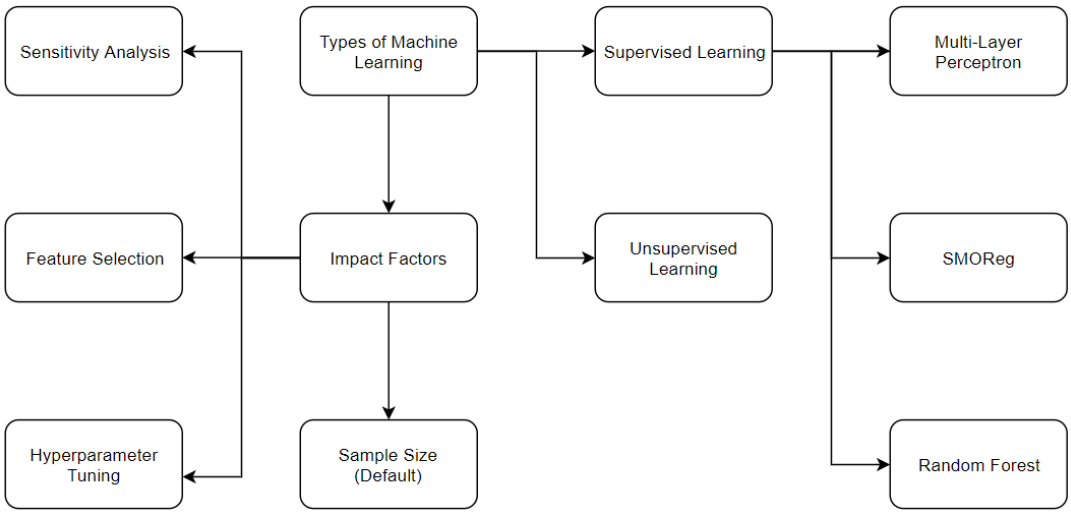


Figure 4:1 Diagram Summary of Chapter Four

4.1 Types of Machine Learning

Murphy (2012) identifies two main types of machine learning: supervised or sometimes called predictive learning and unsupervised or sometimes called descriptive learning. He also acknowledges reinforcement learning which is useful for learning when given reward or penalty signals.

4.1.1 Supervised Learning

	No.	X	Y	Z	Class
Training	1	T	T	T	T
Training	2	T	T	F	T
Training	3	T	T	F	F
Training	4	T	F	F	F
Training	5	F	F	F	F
Test	6	T	F	T	?

Table 4:1 Supervised learning example.

In supervised learning we use data that has the 'answers' provided. These answers are referred to as labels and the data is called a training set. A machine is trained using such training data and is then checked to see how well it has 'learnt' by providing it with a test set. The test set has the same structure as the training set except that it does not contain answers. The machine provides the answers. When the machine scores highly on a test set, that is, it provides correct answers to a high degree of accuracy, then training has been successful, and the machine will be useful in making future predictions.

As a simple example in Table 4:1, imagine providing a machine with data represented by input parameters X, Y, Z and an answer represented by the output parameter 'Class'. We can then ask the machine to classify a test example that is not part of the training set such as X = T, Y = F, Z = T. If it correctly answers that 'Class =

T', then it has been well trained and can be applied, with confidence, to other predictions. It should be noted that the answers given for training are not always 'correct' answers. Looking at row 2 and 3 in Table 4:1 they cannot both be correct. However, machine learning is probabilistic. Given a large enough training set the algorithm will classify test instances using the classification with the highest probability of being correct.

Supervised learning has several applications including medical diagnosis, image recognition and speech recognition. In the example above we can imagine that X stands for 'high temperature', Y for 'sore throat' and Z for 'exhaustion'. Then, 'Class' might be a classifier that asks whether an individual with symptoms X, Y and Z has the flu. Extrapolating, if there are hundreds of symptoms and hundreds of possible diseases, effective machine learning can help a doctor make the correct diagnosis.

In this study we will use the supervised machine learning approach. The training sets will be data generated by simulation engines such as EnergyPlus (energy data) and Radiance (daylighting data). It should be noted that while 'noise' Is typically an inherent aspect of machine learning data there is minimal noise in data generated by simulating engines.

4.1.2 Unsupervised Learning

In the case of unsupervised learning, the machine is provided with data that has no answers. It is essentially just given data to make sense out of. The goal is to discover interesting structure in the data. One of the key strategies to finding structure under unsupervised learning is clustering. Algorithms typically attempt to decompose raw data into related clusters. For example, suppose the given data is:

DBADABCDADAABBCACBABDACABD

An unsupervised learning algorithm, based on lexicographical sorting, might cluster this as:

AAAAAAAAA BBBBBBB CCCC DDDDDD

Examples of unsupervised learning usage include discovering clusters, discovering latent features and discovering structure. Discovering clusters is used in e-commerce to cluster users into groups based on their purchasing behavior. Discovering latent features is useful for dimensional reduction where high dimensional data is reduced to lower dimensional data. Discovering structure allows the algorithm to infer plausible values for missing data. This is called imputation and can be used to complete 'holes' in images, for example.

4.2 Supervised Machine Learning Algorithms

Building performance machine learning problems involve learning to make predictions on unseen input data using labelled training data. They are therefore supervised learning problems. In addition, the output parameters are typically continuous which means the machine learning algorithms required are regression-based. In this section we review three supervised learning algorithms commonly used for building performance analysis. The algorithms are Support Vector Machines, Artificial Neural Networks and Decision Trees. We introduce regression-based implementations of these algorithms for our experiments. For SVMs we use the SMOreg algorithm, for ANNs we use the MLP algorithm and for Decision Trees we use the Random Forest algorithm.

4.2.1 Support Vector Machines

SVMs are an approach to machine learning using decision boundaries. Given a classification problem, an SVM seeks to develop the widest distance between the classes. Given a vector $\boldsymbol{w}$ normal to the decision boundary lines and an unknown vector $\boldsymbol{u}$ to the test instance we can measure whether $\boldsymbol{w.u} \geq c$, where c is a constant. This is the same as $\boldsymbol{w.u} + b \geq 0$, where b is a constant. If $\boldsymbol{w.x} + b \geq 1$ then $\boldsymbol{x}$ can be classified as positive, while if $\boldsymbol{w.x'} - b \leq -1$ then $\boldsymbol{x'}$ is negative, where $\boldsymbol{x}, \boldsymbol{x'}$ are test vectors.

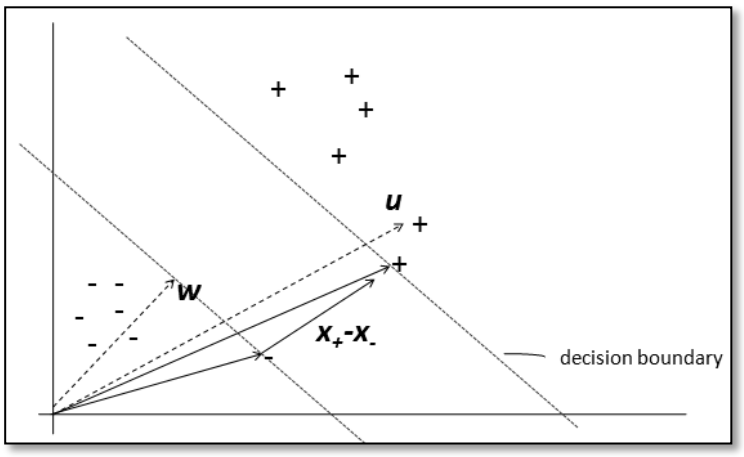


Figure 4:2 Demonstration of the SVM concept.

Introducing a convenience variable *y* such that *y = 1* for the positive sample and *y = -1* for the negative samples we have that, for both boundaries, $y_i(\mathbf{w}.\mathbf{x}_i + b) \geq 1$ and $y_i(\mathbf{w}.\mathbf{x}_i + b) - 1 \geq 0$, where *i* is an index over the training instances. At the decision boundary lines $y_i(\mathbf{w}.\mathbf{x}_i + b) - 1 = 0$.

The width of the decision boundary can be given by the difference of two vectors $\mathbf{x}_+ - \mathbf{x}_-$ where $\mathbf{x}_+$ is a value on the positive boundary line and $\mathbf{x}_-$ is a value on the negative boundary line. If we have a unit normal like $\mathbf{w}/|\mathbf{w}|$ then the width of the decision boundary is $\mathbf{w}/|\mathbf{w}|.(\mathbf{x}_+ - \mathbf{x}_-)$. But now $\mathbf{x}_+.\mathbf{w}/|\mathbf{w}| = 1 + b$ and $\mathbf{x}_-.\mathbf{w}/|\mathbf{w}| = 1 - b$. Therefore, the width is $2/|\mathbf{w}|$.

We wish to maximize $2/|\mathbf{w}|$ or minimize $|\mathbf{w}|$, which for convenience we can write as minimizing $\frac{1}{2}|\mathbf{w}|^2$, given the constraint that $y_i(\mathbf{w}.\mathbf{x}_i + b) - 1 \geq 0$. To solve this constrained optimization problem, we use the method of Lagrange multipliers to write the objective as:

$$L = \frac{1}{2}|\mathbf{w}|^2 - \sum\alpha_i [\, y_i(\mathbf{w}.\mathbf{x}_i + b) - 1]$$

$$\frac{\partial L}{\partial w} = \mathbf{w} - \sum\alpha_i y_i \mathbf{x}_i = 0 \text{ for extremum}$$

$$\frac{\partial L}{\partial b} = -\sum\alpha_i y_i = 0 \text{ for extremum}$$

And this gives $L = \sum\alpha_i - 1/2 \sum_i\sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i.\mathbf{x}_j$ where the α_i are slack variables. Thus, at the extremum, the value of the width depends only on dot products of pairs of training samples. This expression is referred to as a quadratic programming problem and sequential minimal optimization can be used to solve it.

If the data cannot be linearly separated, then it may be separable in a higher dimensional feature space (Figure 4:3). Kernels are a method for mapping data into higher dimensional feature spaces. A kernel is a function *K* such that $K(x,z) = (\varphi(x).\varphi(y))$ where φ is a mapping from *X* to an inner product feature space *F*. Examples of kernels are 'linear', 'polynomial' or 'radial basis function'.

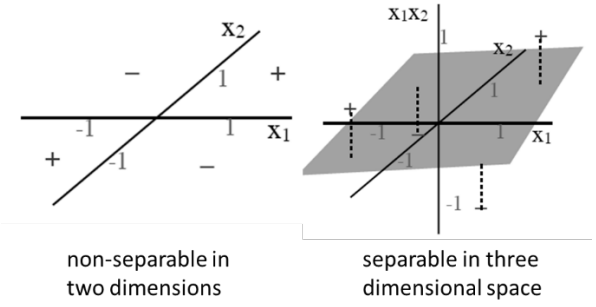


Figure 4:3 Demonstration of the kernel concept.

Since SVMs are typically classification algorithms we use a regression-based implementation of SVM called SMOReg (Shevade et al., 1999). SMOReg is a type of SVM called Support Vector Regression machine (SVR). Smola & Schölkopf (2004) describe the basic premise of an SVR. Given training data $\{(x_1, y_1), \ldots, (x_i, y_i)\}$ where x are the input parameters and y are the output parameters, the goal is to find a function $f(x)$ that has at most ε deviation from the actual target values. In other words, the boundaries associated with this function are as small as possible while containing as many of the actual target values as possible (Figure 4:4).

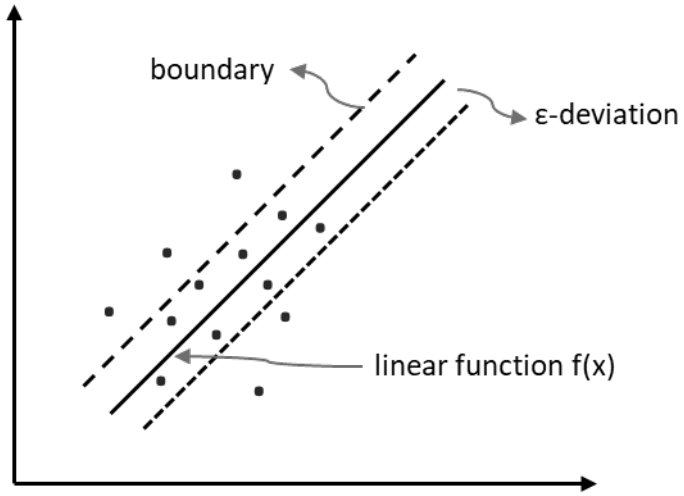
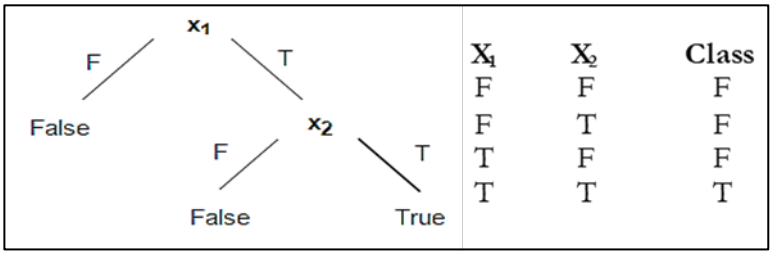


Figure 4:4 Demonstration of the Support Vector Regression concept.

4.2.2 Decision Trees

A decision tree is a representation of a function from a set of discrete attributes to a class. For example, the decision tree and table of attributes for the Boolean statement $x1 \wedge x2$ looks like Figure 4:5. How do we learn a decision tree? We cannot do an exhaustive search of all possible paths down the tree as this is exponential in the depth of the tree. We can possibly use a greedy algorithm such as ID3 or C4.5 (Mitchell, 1997). In ID3 we take a training set of labeled attributes as inputs. At the root we either stop growing the tree or we split the tree at one of the attributes. A sub-tree is then recursively grown for each attribute until the leaves of the tree.

What is the criterion for choosing an attribute to split on? To understand this, we need to understand the idea of bias. Consider the training set in Table 4:2. What is the true concept here? Is it *Attribute 1 AND Attribute 2* or is it *Attribute 2 OR Attribute 3*? In fact, we cannot tell which means our learning algorithm is likely to get stuck trying to learn this concept. This simple example illustrates why it is difficult at first glance to see how we can use learning to classify unseen instances.



X_1	X_2	Class
F	F	F
F	T	F
T	F	F
T	T	T

Figure 4:5 A decision tree for the Boolean statement x1 ∧ x2.

The solution is to accept that learning machines need to have an inductive bias in order to learn. This means they must prefer certain types of data or patterns even if such a preference is sometimes ambiguous. Thus, in Table 4:2, we could have an algorithm with an inductive bias for concepts with the fewest attributes. This means it will prefer a concept like *Attribute 2 = True* to other possibilities.

When applied to decision trees this bias means that they prefer short trees. A heuristic for determining the attribute that would lead to the shortest tree uses the idea of 'entropy' from information theory. The entropy can be interpreted as the amount of disorder in the data. The higher the entropy the less orderly the data and the deeper the trees it will produce. Thus, we should select low entropy attributes to split on.

However, if an attribute has several values, for example if it has a distinct value for each instance of the training set, then such an attribute tells us everything about the labelled class and has low entropy. Such an attribute would not generalize to unseen instances – it is overfitted to the training set. Therefore, a tree with such attributes needs to be pruned. This means that, at any node of the tree, we avoid splitting on attributes that have too many values.

Attribute 1	Attribute 2	Attribute 3	Classification
False	False	False	No
True	False	False	No
True	True	False	Yes
True	True	True	Yes

Table 4:2 Decision tree attributes.

To understand the performance of Decision Tree algorithms we can use the concept of generalized error. The generalized error of a Decision Tree algorithm consists of three parts: bias, variance and an irreducible error due to noise. Bias tells us how much, on average, the trained model differs from ground truth. Variance tells us how inconsistent the trained model is when compared to different test sets it has not encountered before. The bias problem is an underfitting problem where the trained model is not flexible enough to approximate ground truth data. The variance problem is an overfitting problem where the trained model fits noise in addition to ground truth data. The performance goal of a Decision Tree algorithm seeks to achieve a bias-variance trade off.

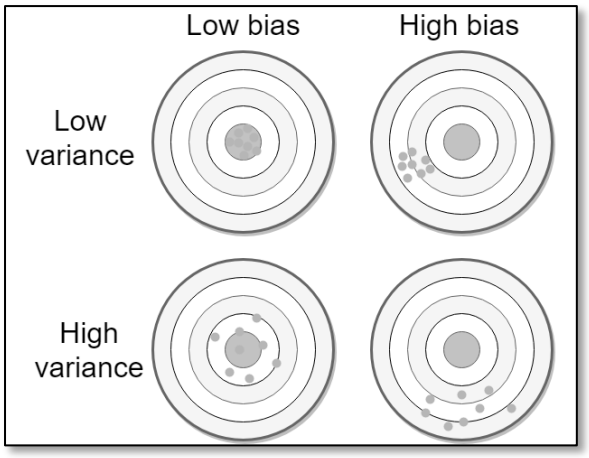


Figure 4:6 (Machine Learning, 2019) Bias-variance trade-off.

Decision Tree algorithms can be categorized as Classification and Regression Trees (CART), Ensemble algorithms, Boosting algorithms and Bagging algorithms. CARTs can be either classification or regression machines. They classify instances according to the following recursive algorithm:

1. If the root of the tree is a class, then stop.

2. If not, then the root is an attribute and the classification is the classification of the sub-tree reached by selecting appropriate values based on attribute values of the root.

However, CARTs face the challenge of bias-variance tradeoff. One approach to reducing the variance of a Decision Tree without increasing the bias is to use Ensemble Learning. Instead of learning using a single algorithm on a single training set, learning is achieved through multiple algorithms. For example, several supervised learners can be trained on a dataset and then prediction is performed by having each algorithm make a prediction and then voting to determine the final outcome. Ensemble learners do not increase bias because the average performance of the Ensemble is the average of each individual learner and these do not change their accuracy. However, Ensembles decrease variance because the individual variance of each learner can be voted down.

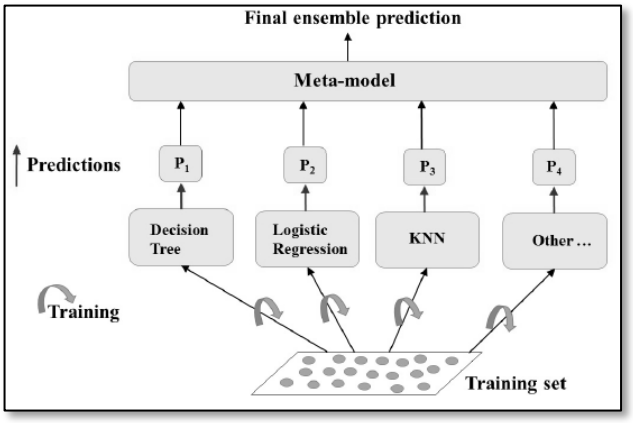


Figure 4:7 (Datacamp, 2019) Ensemble learning.

One type of Ensemble learning is Bagging (or bootstrap aggregating). Bagging uses one algorithm with different subsets of a training set. Bootstrap sampling refers to sampling with replacement from the training set. Each Bootstrap sample develops a model which is then used to make predictions on new instances. Classification is performed by aggregating predictions by majority voting.

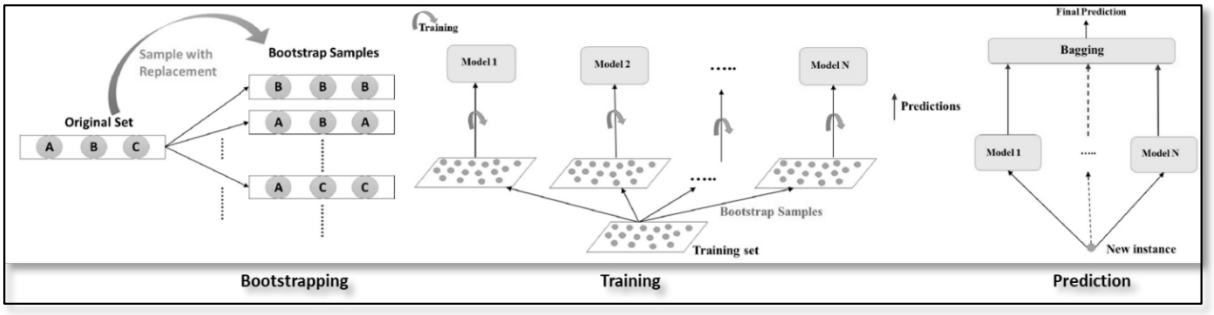


Figure 4:8 (Datacamp, 2019) Bagging.

In this study we use a Bagging method called Random Forest. Brein (2001) defined Random Forests as a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. In other words, each tree in the Random Forest is trained on a different bootstrap sample having the same size as the training set. At each node, each tree uses a specified number of features from the feature set.

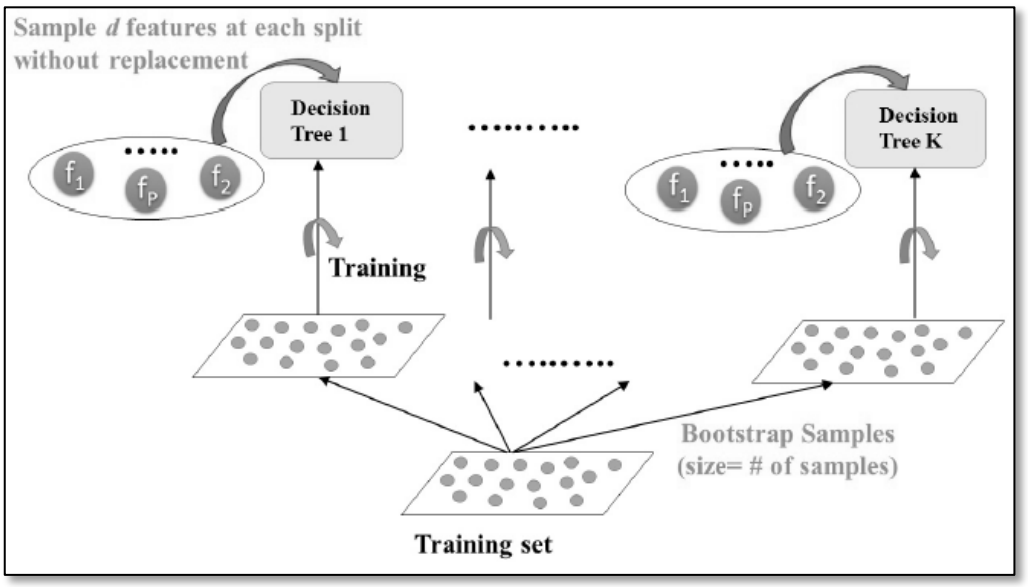


Figure 4:9 (Datacamp, 2019) Random Forest.

Heden (2016) used Random Forests and Support Vector Regression to predict the hourly residential energy consumption in residential buildings. In particular he used number of features and number of trees as hyperparameters. Ahmad et al. (2017) used Random Forests and Artificial Neural Networks to predict energy consumption and daylight illuminance in a classroom in Cardiff. They used maximum depth and maximum features as hyperparameters. In this study we will use the number of features, number of trees and maximum depth as hyperparameters.

Like SVMs, Decision Trees are classification algorithms that have to be adapted to perform regression. This can be done by replacing the idea of Information Gain (Entropy) with that of Standard Deviation Reduction. If a Decision Tree has an output parameter that is real-valued, then when we split on an attribute, we can compute the standard deviation of the output parameter values for each value of the attribute. For each attribute, the standard deviation reduction is the difference between the weighted sum of the standard deviations of each attribute value and the standard deviation of the attribute before the split. The attribute with the largest standard deviation is chosen for the split.

Splitting is performed recursively until some termination criterion is reached. A termination criterion might be when too few instances remain in the branch or when the coefficient of deviation for a branch becomes smaller than a given threshold. We assign the average of each branch to the related leaf node of the attribute.

4.2.3 Artificial Neural Networks

A neural network is a computational structure formed from a network of individual processing units, called neurons, that send messages to each other (Pfeffer, 2009). They are modelled after the human brain although this is not necessary for their mathematical formulation. The simplest model of a neural network is the perceptron. It takes a set of inputs from the training data and associates a weight with each input. It computes the weighted sum of all inputs and passes this to an activation function. If this sum is greater than a defined threshold the function returns 1 (it activates) otherwise it returns 0. The perceptron classifies by learning weights that correctly classify each input instance with the least error.

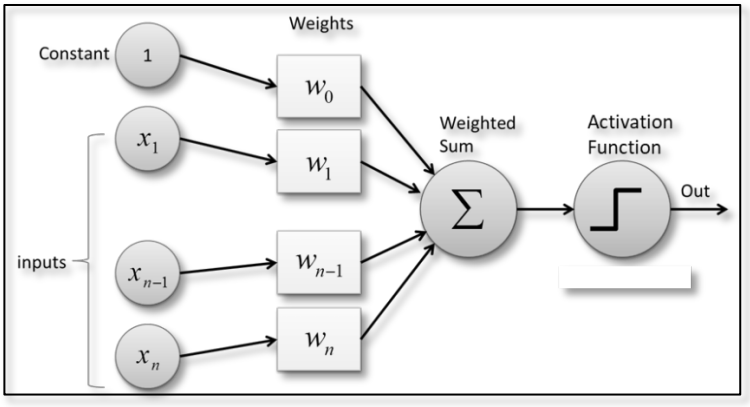


Figure 4:10 (Towards Data Science, 2019) Perceptron.

The structure of a perceptron means that it can only represent linear decision boundaries and linearly separable concepts. This means that there are concepts that perceptrons cannot learn. For example, the boolean concept *x XOR y* cannot be represented by a perceptron. The solution is to use additional layers of neurons which can learn the non-linear concepts by capturing interactions between the input values. Bryson & Ho (1969) developed a multi-layer feed-forward neural network that could learn non-linearly separable concepts like *x XOR y.* However, this type of network raises the question of how the output error is to be distributed between the nodes in the hidden layers so that the weights entering these nodes can be adjusted for learning. An algorithm called the backpropagation algorithm provides an answer to this question.

In this study we use a Multi-Layer Perceptron, a form of multi-layer feed-forward neural network. Multi-Layer Perceptrons have an input layer, hidden layers and an output layer. The nodes in each layer are connected to the nodes in the next layer. Each connection has an associated weight. Each node calculates the weighted sum of its incoming connections and returns a result depending on whether the sum is greater or less than a threshold value.

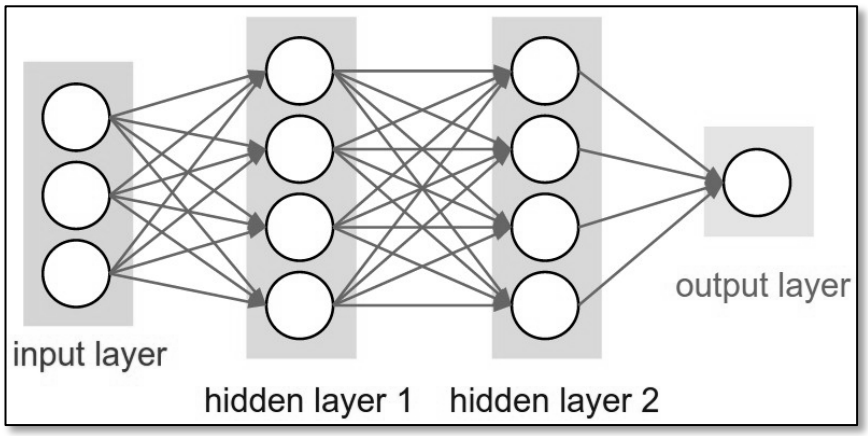


Figure 4:11 (Towards Data Science, 2019) Multi-layer feed-forward neural network

Two important questions can be asked about Multi-Layer Perceptrons. How many hidden layers should they have and how many nodes should each layer have? If they have no hidden layers, then it is a simple perceptron. If they have one hidden layer then Lippman (1987) shows that they can form any, possible unbounded, convex region in the space of the spanned inputs. If they have more than two layers, they can form arbitrarily complex decision regions. With regard to the number of nodes in each layer Witten (2011) recommends using the mean value of the number of nodes in the input and output layers.

The weights of the neural network are learnt from the training data. The idea is to begin with a random weight assignment and then to iteratively reduce the error between the actual output and the expected outputs. The method for achieving this is referred to as gradient descent. The error is then propagated backwards through the network and the weights updated according to the amount they contribute to the error. This is known as backpropagation.

The amount by which weights are updated is controlled by hyperparameters two of which are the *learning rate* and the *momentum*. The *learning rate* controls the amount by which a given weight is updated for a given error. The *momentum* allows weight updates to continue even when there is less error being calculated. In this study the hyperparameters we will explore are the *number of layers*, the *number of nodes per layer*, the *learning rate* and the *momentum*.

4.3 Impact Factors

Investigating the performance of machine learning algorithms for parametric analysis does not simply mean measuring the prediction accuracies. It is also important to understand the factors that influence these accuracies. This helps us fine tune the algorithms for optimal performance. The impacts we study in this research are the sample size, sensitivity analysis, feature selection and hyperparameter tuning (Figure 4:12). Sample Size provides the default performance by running the algorithms using default hyperparameter settings and the benchmark sample size (defined as 1000 instances in this study). Sensitivity Analysis answers the questions to which input parameters are the output parameters most sensitive? Does using sensitive parameters for machine learning improve algorithm performance? Feature Selection answers the question which combination of input parameters, regardless of sensitivity, gives the best performance? Hyperparameter Tuning answers the question what values of the algorithms' hyperparameter settings give the best performance?

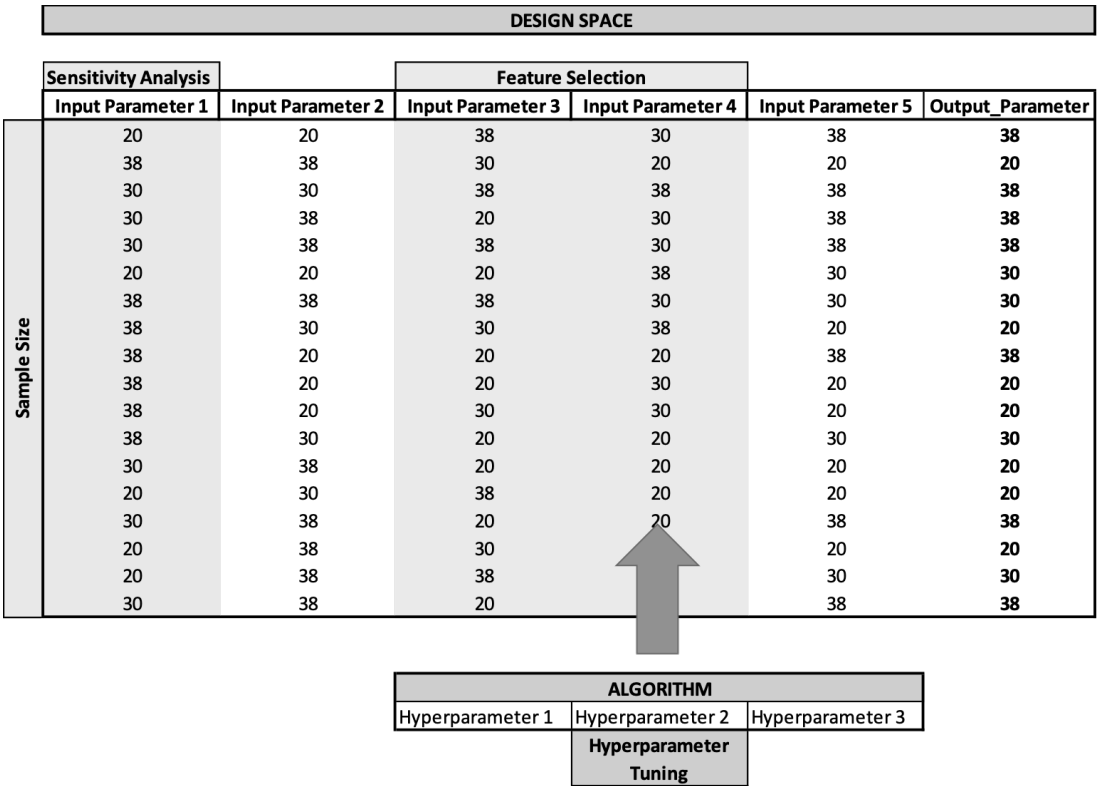


Figure 4:12 Diagram of impact factor study.

Sample Size refers to the number of instances in a training set. It is important for design space reduction because if the sample size required for acceptable accuracy (5 – 10% RAE) is too large then gathering a

training set for the machine learning algorithms will be too computationally expensive. In this study the range of sample sizes investigated is determined by the practical range that can be simulated in a 33-hour period on high specification but generally accessible hardware (see Section 2.3). Based on this assumption we investigate sample sizes ranging from 100 to 1000 instances. We seek to understand the minimum sample size within this range that will give acceptable accuracies.

Sensitivity Analysis is a measure of how the variation of input parameters affects the output parameters. This is important because it can tell the designer which input parameters will have the most effect on the output of interest and thereby help focus attention. It can also help to simplify the model by removing parameters that have very little influence on the output parameters. This can have the effect of making computation faster. Sensitivity Analysis can be calculated using a variety of methods including one-factor-at-a-time methods, scatterplots, regression analysis and correlation analysis. In this study we use the Assess Variable Importance feature of JMP for sensitivity analysis (Figure 4:13). The Assess Variable Importance feature lists the variables in order of their influence on the output parameter. It also indicates the Main Effect, or the impact the input parameter has on its own, and the Total Effect, or the impact it has in combination with other variables. In performing sensitivity analysis, we seek to understand if it has any effect on the performance of machine learning algorithms. Does the removal of insignificant input parameters improve the performance of the algorithms?

Summary Report

Column	Main Effect	Total Effect	.2 .4 .6 .8
Slice	0.742	0.775	
Wall_U_Val	0.123	0.156	
Window_U_Val	0.015	0.034	
WWR_N	0.011	0.026	
SHGC	0.008	0.019	
WWR_W	0.005	0.012	
WWR_E	0.001	0.004	
WWR_S	2e-4	0.001	

Figure 4:13 Sensitivity analysis report from JMP.

Like sensitivity analysis feature selection, or attribute selection, reduces the number of input parameters used for developing a machine learning model. This improves model performance by removing irrelevant or redundant features. Unlike sensitivity analysis which simplifies the model by checking for input parameters that do not impact the output parameter, feature selection simplifies the model by checking for subsets of the parameter space that may perform better than the parameter space itself.

Features targeted for removal can be noisy features, correlated features or duplicated features. We use Weka's Attribute Selection functionality for feature selection. Weka Attribute Selection can be performed by correlation calculation, information gain-based calculation or learner-based feature selection. We wish to understand what subset of input parameters will give the most accurate model for each of our machine learning algorithms. Can we improve algorithm performance by judiciously choosing a subset of the parameter space to train and test?

To better understand the distinction between feature selection and sensitivity analysis consider a training set with input parameters *{A, B, C}* and output parameter *X*. Assume that sensitivity analysis reveals that *X* is more sensitive to *A* than to *B*, and more sensitive to *B* than to *C*. Training sets for the impact of sensitivity analysis will be

{(A), (A, B), (A, B, C)}

on the other hand, training sets for feature selection will be

{(A), (B), (C), (A, B), (A, C), (B, C), (A, B, C)}.

Clearly the sensitivity analysis training sets are a subset of the feature selection training sets. However, in practical terms, the number of subsets of a given training set can be large (2^n, in fact, where n is the number of parameters). This means feature selection only searches a sample of the total subsets. And this means that it is still relevant to perform an impact study based on sensitivity analysis.

Hyperparameter tuning is the problem of selecting the best combination of hyperparameters for best algorithm performance. When working with machine learning models, model parameters are values associated with the model that are established as a result of the training. For example, in linear regression the slope and intercept values are model parameters. Hyperparameters, on the other hand, must be established before training and include weights, biases and options. In neural networks the choice of activation function is a hyperparameter; in decision trees the depth of the tree is an example of a hyperparameter; in support vector machines the choice of kernel is a model hyperparameter. We will use the Weka Experiment Environment to run hyperparameter tuning experiments. We wish to understand by how much we can improve algorithm performance through hyperparameter tuning.

4.3.1 Sample Size

Sample Size affects both sample accuracy and prediction accuracy. In a constrained problem we have a limit to how big we can make the sample as this affects the overall speed of simulation. On the other hand, we have a limit to how small we can make the sample because this affects accuracy. The sample size impact factor investigates the performance of algorithms at the default benchmark size (1000 instances). As such it is the default against which other impact factors are compared. In addition, it investigates the minimum sample size that gives effective performance.

4.3.2 Sensitivity Analysis

Sensitivity Analysis attempts to improve algorithm performance beyond the default by identifying what input parameters have the least impact on the output parameters and investigating whether excluding these parameters improves performance. It can also investigate whether input parameters that have the most impact on the output parameter are too closely correlated and are causing overfitting. As such, sensitivity analysis has two stages. First, we rank the input parameters according to their sensitivity. A simple way to do this is to calculate the correlation coefficient between each input parameter and the output parameter. The input parameter with the highest coefficient is the most sensitive. Second, we run machine learning algorithms on the training dataset sequentially excluding the least sensitive parameters. We check to see if at any point of the process algorithm performance improves beyond the default with all input parameters included.

4.3.3 Feature Selection

Feature selection, or attribute selection, is the process of selecting a subset of a parameter space for use in building data models. Feature selection is different from feature extraction because it does not create any new features or parameters. It simply includes or excludes parameters in an effort to find the selection that gives the best model performance.

Guyon et al. (2008) identify potential benefits of feature selection including facilitating data visualization and understanding, reducing measurement and storage requirements, reducing training and prediction times, and improving performance by counteracting the curse of dimensionality. Browntree (2014) identifies three techniques for performing feature selection using Weka: correlation based feature selection, information gain based feature selection and learner based feature selection.

Correlation based feature selection uses the correlation between input parameters and the output parameter to determine which parameters to include and which parameters to exclude. This is similar to

some approaches to sensitivity analysis. Correlation can be calculated using measures like the Pearson Correlation Coefficient. Input parameters, or features, with moderate to high correlation, positive or negative, are included while parameters with low correlation are excluded.

Information gain-based feature selection calculates the information gain, or entropy, for each input parameter in relation to the output parameter. For a given input parameter if all its values give the same value for the output parameter then it adds no information and has a low entropy. If, on the other hand, each of its values has a different value for the output parameter then it has a lot of information gain. We see that the output parameter is sensitive to changes in the input parameter. Parameters that add higher information gain to the output are selected while attributes that add little information gain are excluded.

Learner based feature selection uses wrapper methods that consider feature selection as a search problem. Different combinations of the input parameters are used for training with a generic algorithm and the performance compared. The combination of input parameters that performs best is selected as the feature set. The search can be performed by various search algorithms including greedy algorithms, best first algorithms and hill climbing algorithms.

4.3.4 Hyperparameter Tuning

According to Kuhn & Johnson (2013) many machine learning models have important parameters that cannot be estimated from data. These parameters are referred to as tuning parameters, or hyperparameters, because there are no analytical formulae for calculating their value. Bergstra & Bengio (2012) define hyperparameter optimization as the problem of identifying good values for hyperparameters to minimize the generalization error of the machine learning algorithm. They distinguish between grid search and random search as two approaches to hyperparameter optimization.

Grid search builds a model for each possible combination of hyperparameter values under consideration. It evaluates each combination and selects the best performing one. If there are several hyperparameters then grid search suffers from the curse of dimensionality and becomes inefficient. Random search provides a statistical distribution of values for each hyperparameter from which a random sample is selected for evaluation. One reason to motivate the use of random search is that not all hyperparameters are equally important. Thus, quickly identifying the less significant parameters and focusing the optimization on significant parameters gives better search performance than exhaustively evaluating every possible combination.

In this study we identify from literature a set of hyperparameters to investigate for each algorithm. We establish from literature a range of values for each hyperparameter. We run experiments where each value of each hyperparameter is tested. We check to see if at any point during the experiments algorithm performance improves beyond the default.

Chapter 5 Methodology

The research methods applied in this study are simulation methods and experimental methods. These are applied to three reference projects. The reference projects are real world projects designed by Perkins&Will Architects that provide the data used for machine learning. We build on a methodological framework called Design Space Construction (DSC) to implement parametric analysis. Simulation methodology describes the tools and processes for generating the building performance analysis data. Experimental methodology describes the machine learning experiments and the impact studies performed. The general experimental method provides details on how repetitive aspects of the experimental methodology will be performed. This will prevent having to describe these aspects for each of the reference projects.

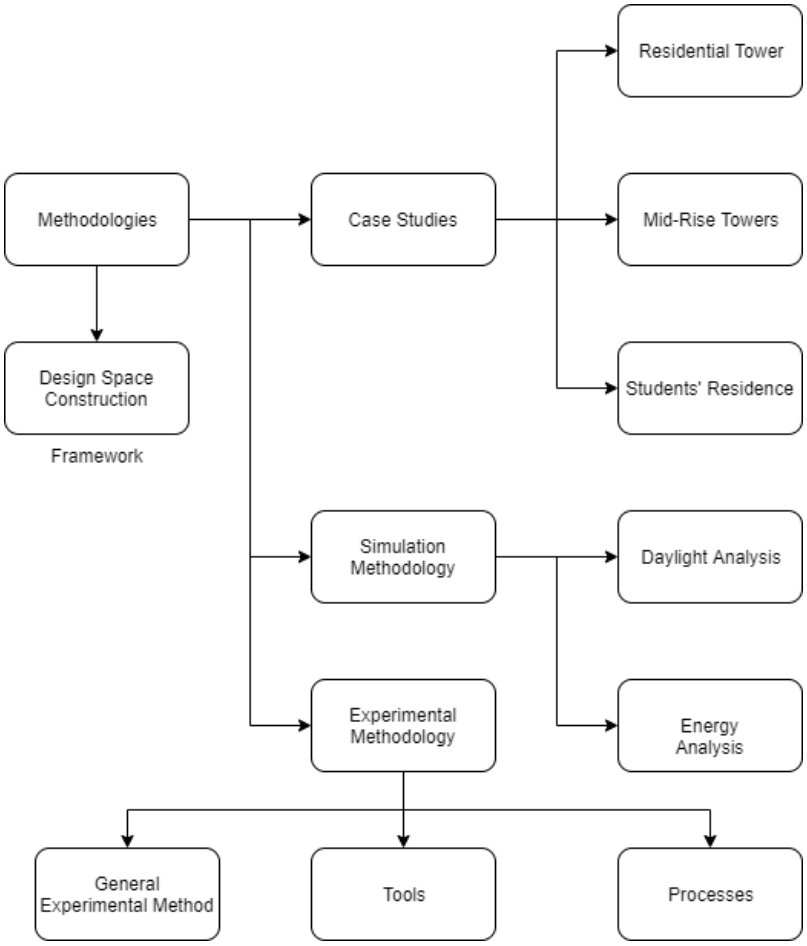


Figure 5:1 Diagram Summary of Chapter Five

5.1 Reference Projects

The three reference projects, all located in British Columbia, Canada, are *Residential Tower*, a mixed use commercial-residential tower development (Figure 5:2); *Mid-Rise Towers*, three mixed use commercial-residential towers standing on two podiums (Figure 5:3), and *Students' Residence,* a students' campus residential building (Figure 5:4). For Residential Tower a typical mid-level floor was analyzed for energy and daylighting performance. The design questions the team sought to answer were what the most appropriate window to wall ratios on each façade of the building are, and what is the most appropriate depth of balconies for optimum building performance.

For Mid-Rise Towers eight slices of the towers were analyzed for daylighting and energy performance. The challenge was to provide design guidance for each of the different facades at different locations of the design. Facades had different orientations and shading effects while different levels of the building had different adjacency conditions. For Students' Residence, a typical upper floor was selected for analysis. The design team was exploring three design alternatives with different geometries and orientations and wished to compare them in terms of building performance. All three alternatives were simulated, and statistical testing was used to determine which of the alternatives performed best.



Figure 5:2 (Perkins&Will, 2019) Residential Tower, Vancouver.



Figure 5:3 (Perkins&Will, 2019) Mid-Rise Towers, Vancouver.



Figure 5:4 (Perkins&Will, 2019) Students' Residence, Vancouver.

5.2 Design Space Construction

Research and practical experience show that parametric analysis needs systematic implementation. All the reference projects examined in this research utilized DSC as their implementation approach. DSC is a formal methodological framework that addresses the challenges of implementing design space exploration in practice (Haymaker, et al., 2018). DSC involves a set of roles and a four-step process for performing design explorations (Figure 5:5). The roles are stakeholders, decision makers, designers and gatekeepers. The steps are objective definition, alternative generation, impact analysis and value assessment. A stakeholder is a person or organization that a design decision may affect. A decision maker identifies stakeholders and appoints designers and then participates in assessing value for decision making. A designer defines design alternatives and analyzes those alternatives considering requirements and constraints. Gatekeepers are individuals or groups with the power to constrain the range of viable alternatives (Chachere & Haymaker , 2011).

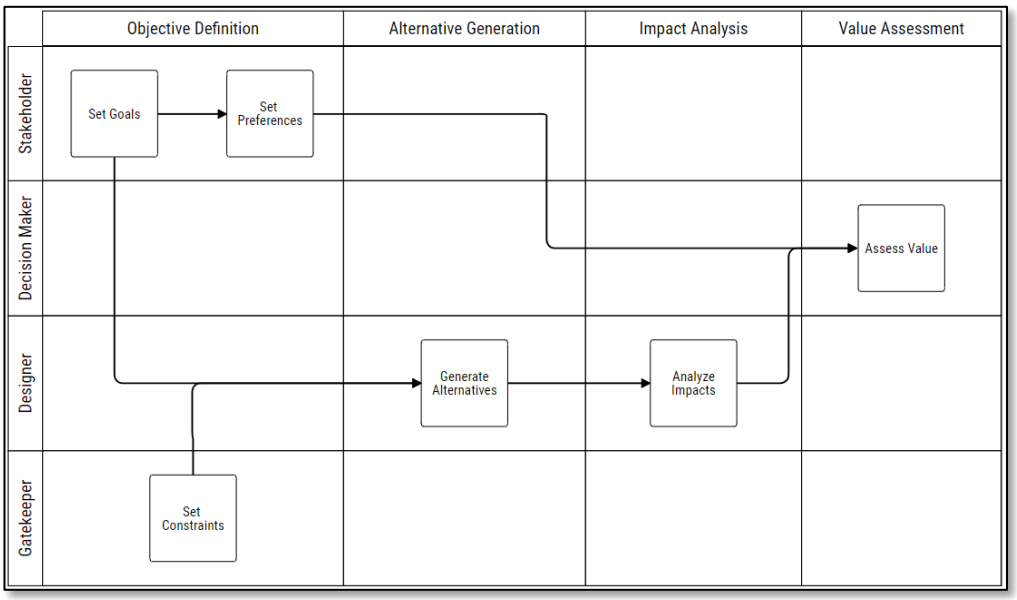


Figure 5:5 Design Space Construction (DSC) framework.

5.2.1 The DSC Process

Objective Definition (Table 5:1) formulates the problem by assembling the relevant team and establishing the design exploration objectives. Goals, defined by stakeholders, define specific desirable outcomes. Objectives are specific targets, associated with goals, to aim for or failures to prevent. Most design problems in the real world will have multiple goals, or objectives, and the optimization of one objective may well compromise other objectives. This gives rise to the need for multi-objective optimization. Constraints, defined by gatekeepers, represent the bounds of input and output variables and must be satisfied for an alternative to be considered viable. Ultimately, metrics allow for the verification of whether objectives were met, or constraints were obeyed.

Category	Metric	Description	Units	Constraint
Experience	Daylight Factor	Percentage floor area with a specified percentage of lux compared to outdoor lux value	%	2% < Daylight Factor < 5%
	Quality Views	Percentage of floor area with a direct view to desirable outdoor features.	%	> 50% landscape features visible
	Direct Line of Sight	Percentage of floor area with a direct line of sight to the outdoors	%	> 75% floor area with direct line of sight
Ecological	CO_2 Emissions	Amount of CO_2 associated with construction and operation of the building	tons/sqm	< 6
	Total Site Energy	Total annual heating and cooling loads	kwh/m^2/year	< 30 kwh/m^2/year
Economic	First Cost	Estimate of construction cost	$/sqft	< $300/sqft
	Life Cycle Cost	Estimate of energy cost to operate the building	$/year	< $30,000/year

Table 5:1 DSC Objective Definition example.

Alternative Generation (Figure 5:6) creates the different design concepts that are to be explored. Each alternative corresponds to an input parameter value for the input parameters in the parameter space. Parameters can be categorical or numerical and they define a range referred to as the parameter value space. The design space is generated by combinatorically selecting parameter values from each parameter value space. Due to the potentially large size of design spaces, design teams must select parameter value spaces judiciously in order to constrain the size of the design space while ensuring the design problem is adequately described.

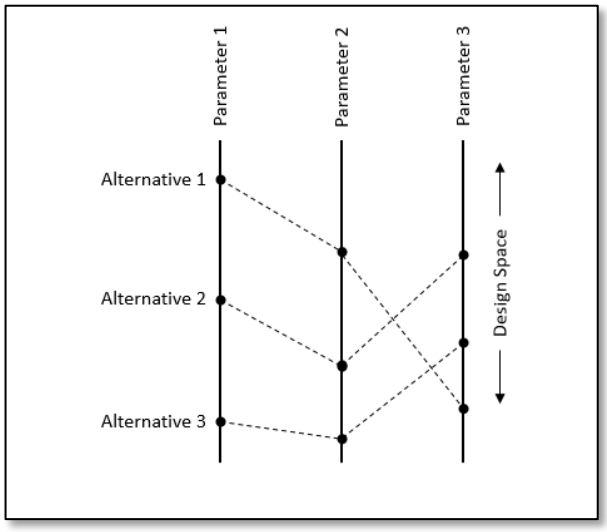


Figure 5:6 DSC Alternative Generation.

Impact Analysis (Figure 5:7) refers to quantifying the influence of the combined inputs of a design alternative on each of the relevant outputs. The outputs are typically classified as experiential, ecological and economical. Examples of experiential outputs include quality views and direct line of sight metrics; examples

of ecological outputs include heating loads, cooling loads and daylight factors; examples of economical outputs are first cost and life cycle costs. Impact analysis of each of these outputs involves the use of specialized analytical tools. The DSC framework seeks to integrate these tools into a unified analysis toolkit.

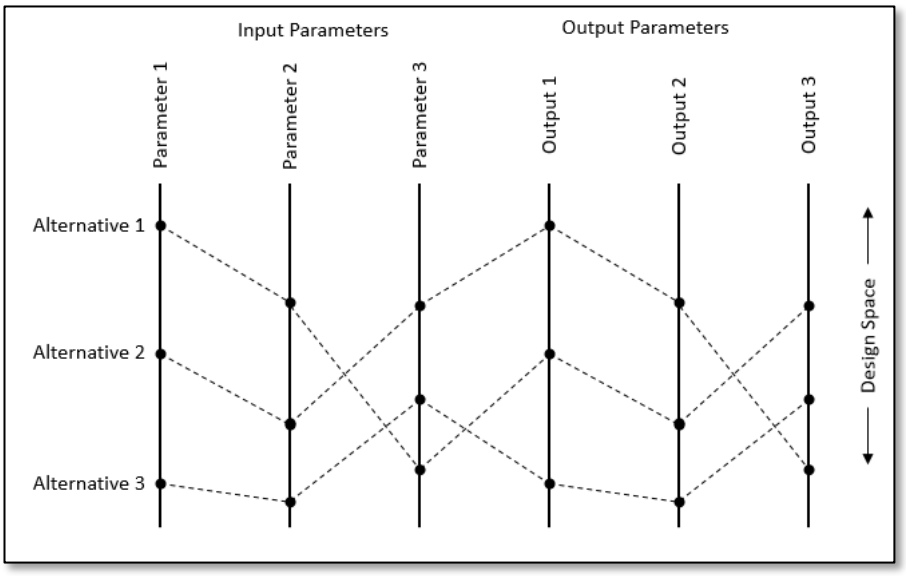


Figure 5:7 DSC Impact Analysis.

Value Assessment (Figure 5:8) evaluates the impact of each design alternative and identifies preferred alternatives to establish a design solution set. The design solution set is a range of alternatives that best respond to the design problem given the objectives, constraints and impacts that have been developed for the problem. Indeed, the goal of design space exploration is to find design solution sets. Value can be assessed as a single objective output such as finding the best solutions for the heating loads of a design; or it can be assessed as a multiple objective output such as finding the best solution for the heating load given that we require the best solution for daylighting as well. In this latter case a value function is used to calculate the values of a multi-objective output variable.

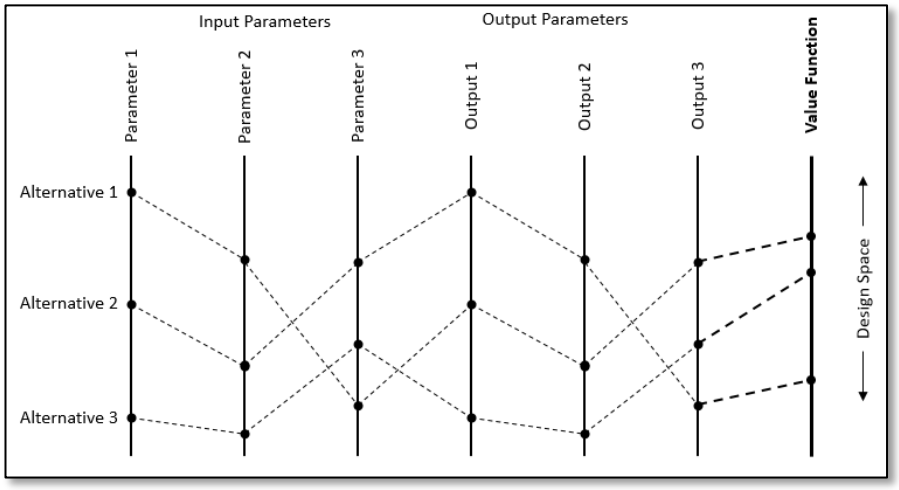


Figure 5:8 DSC Value Assessment.

In the reference project chapters of this research (Chapters 6, 7, 8) the project description will be given in terms of the DSC methodology. This will help give a structured description of the goals, the process and the outcomes of the analysis. In addition, the incorporation of machine learning modifies the steps of the simulation process.

5.2.2 The Value Function

It has been mentioned that one unique feature of architectural parametric analysis is that it is inherently multi-disciplinary. This raises the apples and oranges question of how to compare metrics from different disciplines measured in distinct units. For example, EUI is measured in kwh/m^2/year while daylight illuminance is measured in lux. Is having 30 kwh/m^2/year in a classroom better than having 200 lux? More to the point, is having 30 kwh/m^2/year and 200 lux in a classroom better than 40 kwh/m^2/year and 300 lux?

In single objective problems the superiority of one alternative over another is easily established by comparing their outputs. For multi-objective problems, including multi-discipline problems, more sophisticated methods are required. These methods include Pareto fronts, weighted sum methods and epsilon-constraint methods. Pareto fronts make use of the concept of dominance. In particular a multi-objective alternative *x* dominates a multi-objective alternative *y* if *x* is no worse that *y* in all objectives, and *x* is strictly better than *y* in at least one objective. For a given multi-objective design space the set of all non-dominated alternatives is called a Pareto-optimal set. We can graph all alternatives in the Pareto-optimal set to form a Pareto-optimal front (Figure 3:5).

In this study we make use of the weighted sum method to calculate the value function. The weighted sum method normalizes the objectives of a multi-objective problem and adds them up into a single objective. The objectives are multiplied by a weight which reflects their relative importance for the goals of the analysis. If the objectives are conflicting, then it may be necessary to invert one or more objectives in developing the summed single objective. The advantage of the weighted sum method is that it is conceptually simple, and it has a format suitable for machine learning. However, it is not a straightforward matter to establish the weights needed, particularly in a multi-discipline case. In addition, if the Pareto front of the problem is non-convex then the weighted sum method cannot find certain Pareto-optimal solutions.

In order to transform the energy and daylight outputs so that we can compare them on the same scale we will make use of the normalization formula

$$x = x - x_{min} \,/\, x_{max} - x_{min}$$

where *x* is the output value of any given alternative, x_{min} is the minimum output value of the objective and x_{max} is the maximum output value of the objective. The results are given on a scale from 0 to 1. Because energy and daylighting are conflicting objectives, which means we seek to minimize the energy objective while maximizing the daylight objective, we invert one of the objectives, say energy, by subtracting its normalized value from 1. This way we can simply seek to maximize the summed objective.

For the purposes of this study we will assume all objectives of a multi-objective analysis have equal weight. Although in the real world this is often not the case, for the purpose of evaluating machine learning algorithms it is a reasonable assumption. In summary, the value function of the multi-discipline analysis will be calculated according to

$$VF = E + D$$

where *VF* is the summed objective for any given design alternative, *E* is the normalized, weighted and inverted energy objective, and *D* is the normalized and weighted daylighting objective.

5.3 Simulation Methodology

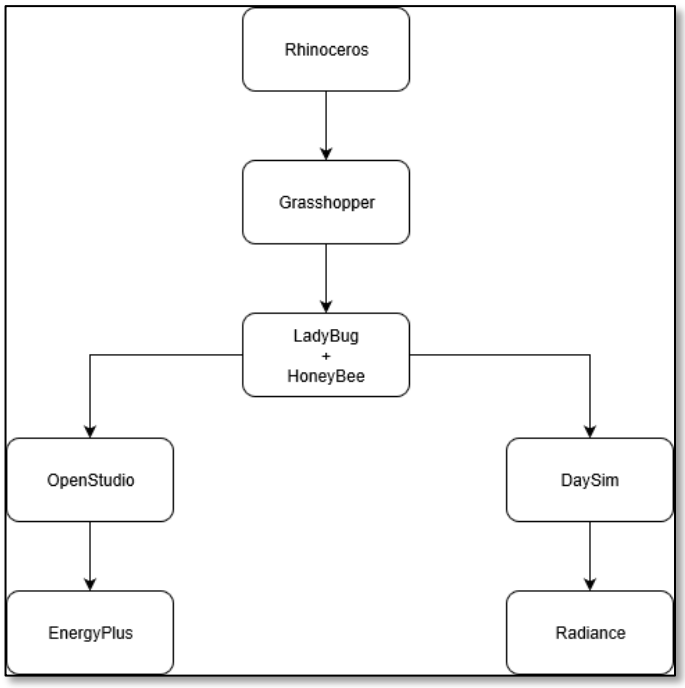


Figure 5:9 Simulation tools.

5.3.1 Tools

One of the distinctive features of the parametric analysis process investigated in this study is that it is a multi-discipline process. This has the benefit of providing holistic insights to the architectural designer. We perform daylight and energy analyses and seek a systematic approach to balance the conflicting influences of these criteria on the design. In order to perform this multi-discipline comparative analysis, we need to solve the problem of simultaneously simulating energy and daylighting based on identical input parameters.

Ladybug and Honeybee are plug-ins for Grasshopper that assist designers in building performance analysis. Ladybug uses EnergyPlus weather files (.epw) to provide a variety of interactive graphics such as sun path, wind rose, shadow studies and view analysis. Honeybee provides an interface between Grasshopper and EnergyPlus, Radiance, DaySim™ (Reinhart, 2019) and OpenStudio™ (US Department of Energy, 2019) for building energy and daylight simulation.

EnergyPlus is a whole building energy simulation program used for modeling energy consumption and water use in buildings. Some of its features include solving thermal zone conditions and HVAC system responses, heat balance-based solutions of radiant and convective effects, user-definable time step calculations, combined heat and mass transfer models, advanced fenestration modeling capability, illuminance and glare calculations and standard as well as user-definable comprehensive reporting. Radiance is a suite of programs for the analysis and visualization of lighting design. It takes scene geometry, materials, luminaires, time, date and sky conditions as inputs and outputs spectral radiance, irradiance and glare indices. It is used by architects and engineers to predict illumination, visual quality and appearance of design spaces.

It is worth mentioning that in practical terms several other plug-ins and helper tools are needed through the process. Two that are worth mentioning are DaySim and OpenStudio. DaySim is a Radiance based daylight analysis software that models the annual amount of daylight for building design. While Radiance performs radiosity and raytracing simulation, DaySim allows for time series metrics like Daylight Autonomy (DA) and Useful Daylight Illuminance (UDI). OpenStudio is a collection of tools that support building performance

analysis using EnergyPlus and Radiance. In this study it provides supplementary functionality that facilitates the creation of EnergyPlus input files (.idf). This functionality includes space types, internal loads and schedules; constructions and materials; HVAC systems, and service hot water inputs.

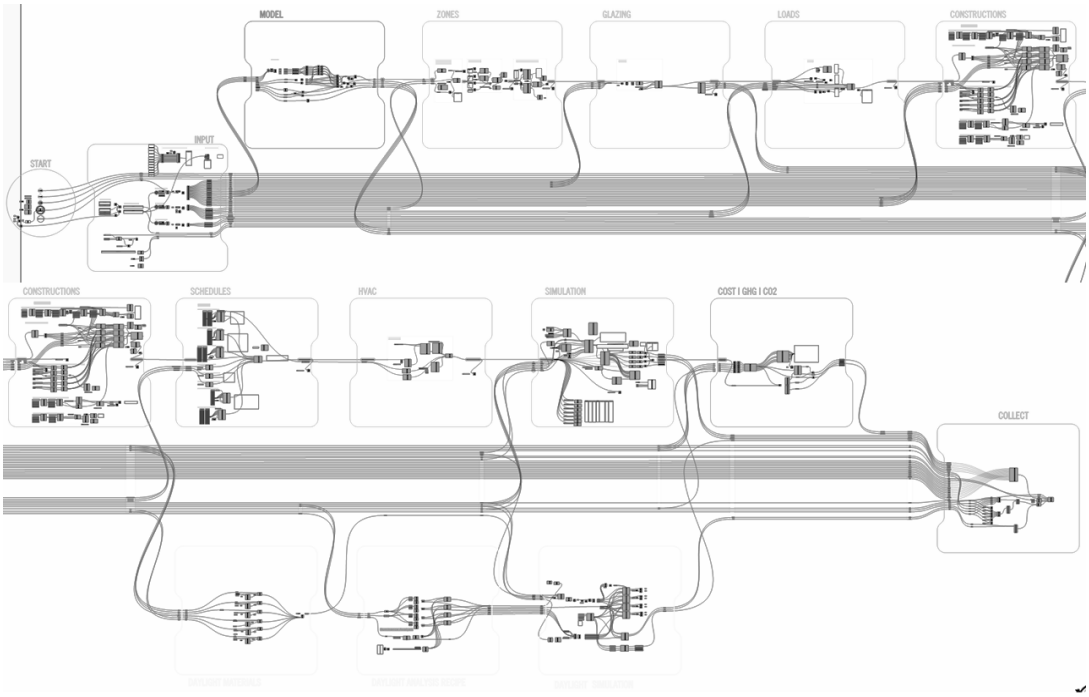


Figure 5:10 (Perkins&Will, 2019) DSC Script in Rhinoceros + Grasshopper.

5.3.2 Energy Analysis

Energy analysis (Figure 5:11) is performed using EnergyPlus as the simulation engine, Rhinoceros + Grasshopper as the parametric geometry modelers and Honeybee as the interface between the geometric modelers and the simulation engine. The metrics of energy analysis are Heating Load, Cooling Load and Energy Use Intensity (EUI). Heating Load is the amount of heat energy that needs to be added to a space to maintain comfort zone temperature. Cooling Load is the amount of heat energy that needs to be removed from a space to maintain comfort zone temperature. EUI is the measurement of a building's annual energy consumption in relation to its gross area.

The analysis is performed using a custom Grasshopper script developed at Perkins&Will referred to as the DSC Script (Figure 5:10). The general steps for performing energy analysis are:

1. Create design zone geometry and design context geometry in Rhinoceros.
2. Create parametric window and sun shading geometry in Grasshopper.
3. Pass model geometry and energy analysis parameters into the Honeybee component in Grasshopper.

Key energy analysis parameters entered into the Honeybee interface include:

1. True north bearings: this provides the orientation of the design model.
2. Weather file: this is an EnergyPlus weather file in .epw file format.
3. Analysis period: this can be daily, monthly, annual, etc.
4. Zone data: this refines the definition of zone geometry.

5. Context data: this refines the definition of surrounding objects.

6. Simulation outputs: this selects specific output metrics to be reported from EnergyPlus.

The Honeybee component in Grasshopper calls EnergyPlus and passes the inputs into the EnergyPlus .idf file format. EnergyPlus runs the simulations and writes out the result files to a specified folder location.

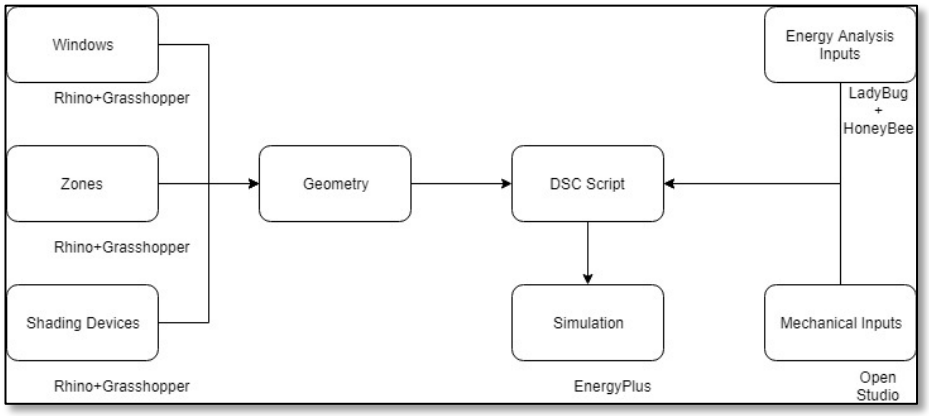


Figure 5:11 Energy analysis process.

5.3.3 Daylight Analysis

Daylight analysis (Figure 5:12) is performed using Radiance as the simulation engine, Rhinoceros + Grasshopper as the parametric geometry modelers and Honeybee as the interface between the geometric modelers and the simulation engine. The metrics of daylight analysis include Daylight Factor, Illuminance and Useful Daylight Illuminance. Daylight Factor is the ratio between exterior sky illuminance on an overcast day and illuminance inside a building. Illuminance is a point-in-time metric that measures the light level at a specific date and time. Useful Daylight Illuminance is the fraction of occupied hours per year where the Illuminance at a point lies between minimum and maximum threshold values.

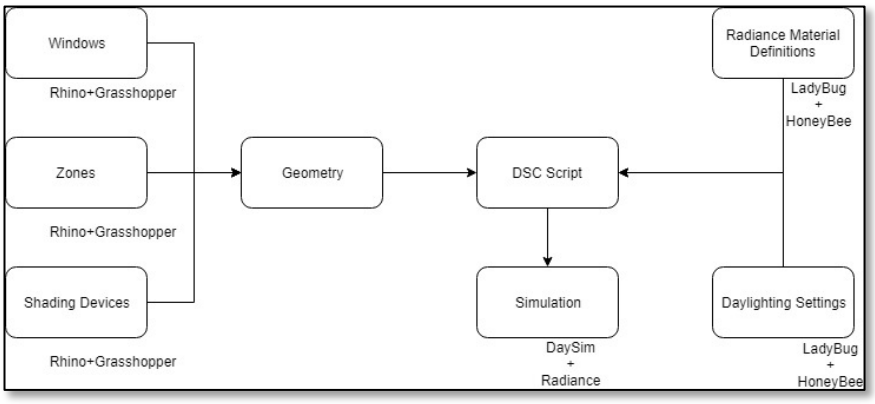


Figure 5:12 Daylight analysis process.

Daylight analysis is also performed using the custom DSC Script (Figure 5:10). The general steps for performing daylight analysis are:

1. Create design geometry and context geometry in Rhinoceros.
2. Create parametric window and sun shading geometry in Grasshopper.
3. Associate Radiance materials with Rhino geometry surfaces.
4. Create an analysis recipe for Radiance.

Key Radiance analysis recipe parameters include:

1. Sky: this defines sky metrics and can be annual, climate-based, etc.
2. Rhino views: this identifies the Rhinoceros views that will be rendered by Radiance.
3. Camera type: this defines the type of camera such as fisheye or perspective.
4. Section plane: this defines a clipping plane for the analysis.
5. Radiance parameters: this controls render settings such as ambient bounces of light.

The Honeybee component in Grasshopper calls Radiance and passes the design geometry and analysis recipe. Radiance runs the simulation and writes out image and output files to a specified folder location. Image files are a graphic representation of the analysis results (Figure 5:13).

In the DSC framework, daylight analysis and energy analysis are amalgamated into one Grasshopper definition in order to run simultaneously based on the same parametric analysis inputs. Identical inputs mean that each individual design alternative in the design space has a result for both energy and daylighting. This allows a multi-discipline comparison of the outputs and the identification of a balanced optimum solution. Such solutions are more holistic in their design guidance than conventional analysis solutions which optimize each discipline individually.

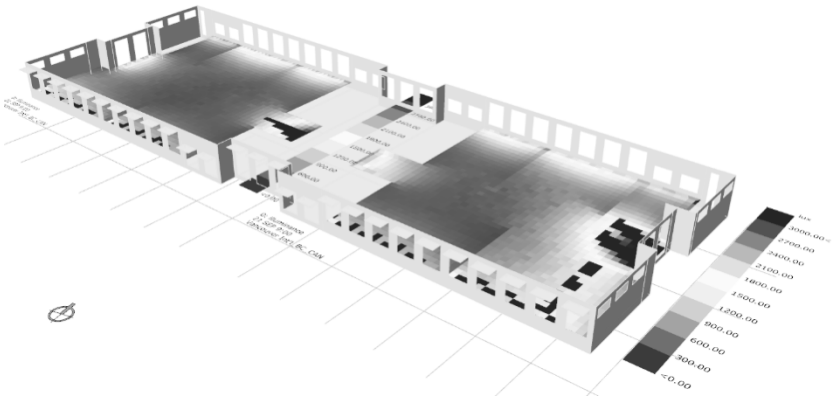


Figure 5:13 (Perkins&Will, 2019) Daylight analysis graphic output.

5.4 Experimental Methodology

The goal of the experimental methodology is to describe the process by which machine learning algorithms are evaluated for effectiveness in performing parametric analysis in architectural practice. The methodology consists of tools, impact studies and evaluation methods. The tools are the specific software needed to undertake the analytical tasks. The purpose of the impact studies is to evaluate the performance of algorithms using default sample sizes and settings, and then to investigate whether any impact factors can improve the performance beyond the default. In this study the defaults are investigated using the sample size impact factor and the other impact factors are sensitivity analysis, feature selection and hyperparameter tuning. The evaluation methods describe how analysis will be performed on experimental results to draw conclusions. This will include comparative graphs and tables, and statistical tests.

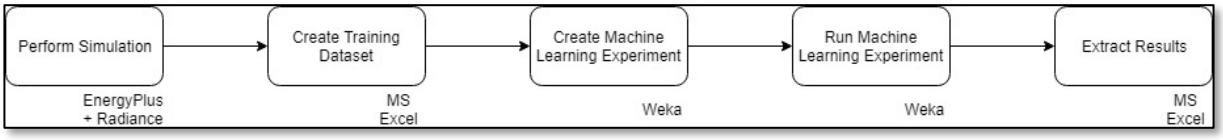


Figure 5:14 Experimental process.

5.4.1 Tools

The tools for the machine learning experiments of this study are Weka™ (University of Waikato, 2019), JMP™ (SAS, 2019), SPSS™ (IBM, 2019) and MS Excel™ (Microsoft, 2019). Weka is an open source toolbox of machine learning algorithms developed at the University of Waikato. It includes tools for data preparation, classification, regression, clustering, association rules mining and visualization. JMP is a version of the statistical package SAS™ (SAS Institute, 2019), while SPSS is IBM's statistical package. MS Excel is Microsoft's flagship spreadsheet product. Weka is the machine learning engine where the machine learning models will be developed and tested. JMP will be used to perform experimental designs which create space-filling random samples of design spaces to serve as training sets for machine learning. It will also be used for performing sensitivity analysis which establishes the input parameters with most influence on the output parameters. SPSS will be used for statistical methods such as parametric tests (Pearson correlation, t-tests, ANOVA) and MS Excel will be used for data exploration and data wrangling. It will also be used for graphing and data visualization.

The Weka Experimenter (Figure 5:15) is the machine learning interface for most of our experiments. It allows the user to upload several training sets and then select several algorithms to run on the training sets. It also allows the user to select suitable experiment and algorithm settings. Sample Size, sensitivity analysis and hyperparameter tuning experiments will use the Weka Experiment. Feature Selection will use the Weka Explorer environment with the WrapperSubsetEval algorithm. Experimental settings will be 10-fold cross-validation with 10 repetitions. The repetitions use a random seed to ensure the 10-fold cross-validation uses different folds for each repetition. This means that each training set is iterated 100 times. Accordingly, we will also report the standard deviation for each training set on the output graphs. Figure 5:16 shows an example of experimental results from the Weka Experimenter.

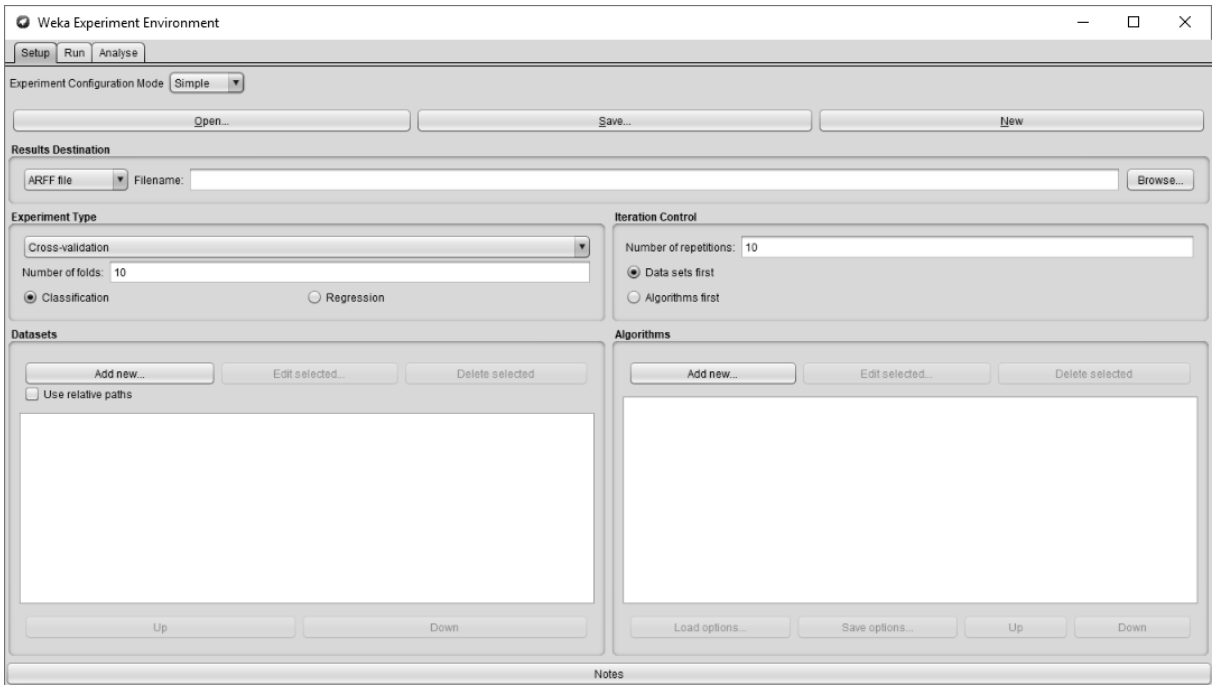


Figure 5:15 The Weka Experimenter interface.

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -
Analysing:  Relative_absolute_error
Datasets:   10
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       11/16/19 2:59 PM


Dataset                   (1) functions.Mult | (2) functions.S (3) trees.Rando
----------------------------------------------------------------------------
RT_100_SS_E               (100)   11.34(3.18) |   17.24(4.99) v   27.99(5.47) v
RT_200_SS_E               (100)    7.62(2.20) |   15.92(2.93) v   20.55(3.58) v
RT_300_SS_E               (100)    6.92(1.74) |   17.39(3.02) v   18.41(2.70) v
RT_400_SS_E               (100)    6.79(1.67) |   17.71(2.79) v   18.08(2.03) v
RT_500_SS_E               (100)    6.11(1.79) |   17.47(2.30) v   16.24(1.81) v
RT_600_SS_E               (100)    5.57(1.37) |   17.51(2.16) v   15.20(1.66) v
RT_700_SS_E               (100)    5.62(1.58) |   17.13(1.66) v   14.24(1.31) v
RT_800_SS_E               (100)    5.39(1.46) |   17.27(1.85) v   13.62(1.19) v
RT_900_SS_E               (100)    5.14(1.11) |   16.66(1.57) v   12.87(1.07) v
RT_1000_SS_E              (100)    5.42(1.26) |   16.38(1.56) v   12.50(1.15) v
----------------------------------------------------------------------------
                                  (v/ /*) |        (10/0/0)        (10/0/0)


Key:
(1) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a'
(2) functions.SMOreg '-C 1.0 -N 0 -I \"functions.supportVector.RegSMOImproved -
(3) trees.RandomForest '-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1' -
```

Figure 5:16 The Weka Experimenter experiment results

5.4.2 Impact Studies

Impact studies have been discussed in Section 4.3.

5.4.3 Evaluation Methods

In the Results section of each experiment we discuss results specific to each impact study. For the sample size we plot a comparative graph of RAE at different sample sizes for each algorithm (Figure 5:17). We also plot the standard deviations of each experiment recalling that they are run 100 times (10 iterations each with 10-fold cross-validation). For the sensitivity analysis experiments we plot a comparative graph of RAE at different parameter counts (Figure 5:18). For feature selection we extract the results of the wrapper algorithm into a table (Figure 5:19) and then graph the table. For hyperparameter tuning we again extract results into a table (Figure 5:20) and graph the table.

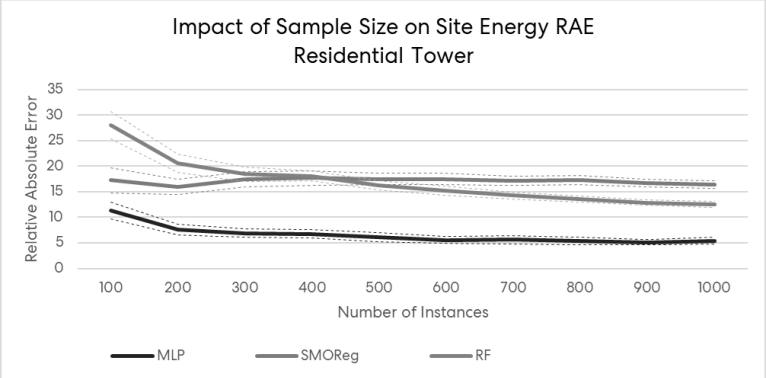


Figure 5:17 Evaluation of sample size results.

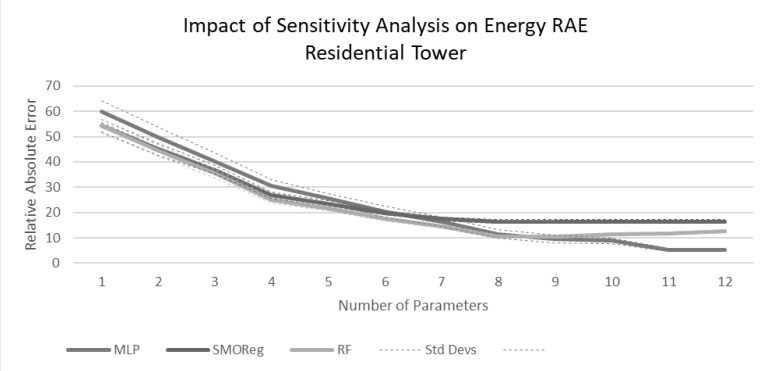


Figure 5:18 Evaluation of sensitivity analysis results.

ALGORITHM	OBJECTIVE	BEST FEATURES	RAE
MLP	Energy	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Shading_N, Shading_W, Shading_S, Shading_E, Window_U_Value, SHGC	4.56
	Illuminance	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Shading_N, Shading_W, Shading_S, Shading_E, Window_U_Value, SHGC	8.93
	Value Function	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Chi_Value, Shading_N, Shading_W, Shading_S, Shading_E, Window_U_Value, SHGC	9.81
SMOReg	Energy	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Shading_W, Shading_S, Window_U_Value, SHGC	16.3
	Illuminance	WWR_N, WWR_W, WWR_S, WWR_E, Shading_N, Shading_S	27.7
	Value Function	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Shading_N, Shading_W, Shading_S, Shading_E, Window_U_Value, SHGC	27.2
RF	Energy	WWR_N, WWR_S, WWR_E, Wall_U_Value, Shading_N, Shading_W, Shading_S, Window_U_Value, SHGC	9.98
	Illuminance	WWR_N, WWR_W, WWR_S, WWR_E, Chi_Value, Shading_N, Shading_W, Shading_S	11.36
	Value Function	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Chi_Value, Shading_N, Shading_W, Shading_S, Window_U_Value, SHGC	18.11

Figure 5:19 Evaluation of feature selection results.

SMOReg	complexity			kernel			epsilonParameter			Best Value	Hyper-parameter
	0.001	*0.01*	*0.1*	*10*	*rbfKernel*	*normalized-Polykernel*	*0.01*	*0.1*	*1*		
Energy	32.97	16.48	16.38	16.37	11.97	11.69	16.31	19.15	104.08	11.69	kernel
Illuminance	46.09	28.05	27.73	27.71	22.9	11.24	27.71	34.21	106	11.24	kernel
Value Function	42.93	27.6	27.23	27.24	22.85	14.31	27.25	29.3	100.08	14.31	kernel

Figure 5:20 Evaluation of hyperparameter tuning results.

In the Discussion section of each experiment we identify the best performance for each algorithm from each impact study. This allows us to compare the best performance from the impact studies for each algorithm in one comparative graph (Figure 5:20). This graph shows if algorithms are effective for each objective. It also shows if any of the impact factors improves algorithm performance beyond the default (sample size).

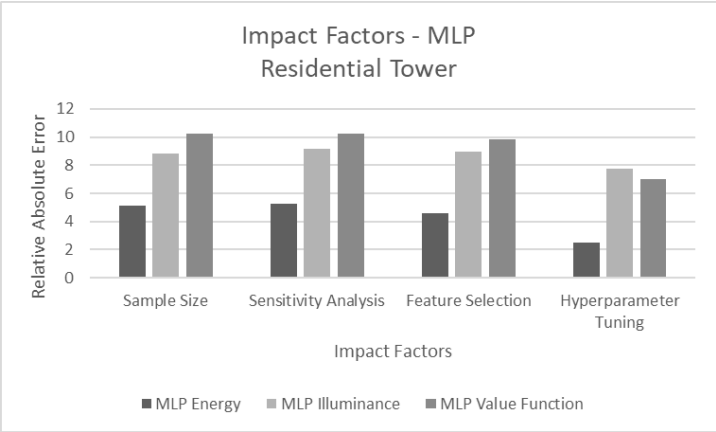


Figure 5:21 Evaluation of impact factors for one algorithm on one experiment.

In Chapter 9 we aggregate the findings from the three experiments to answer the research questions. We gather all tables and graphs from each Discussion section of each experiment and use these to identify which algorithms were effective. We report this in a tabular matrix. We also use the tables to calculate the average performance of each impact factor for each algorithm across all experiments (Figure 5:22). This answers the question whether impact factors improve algorithm performance beyond the default.

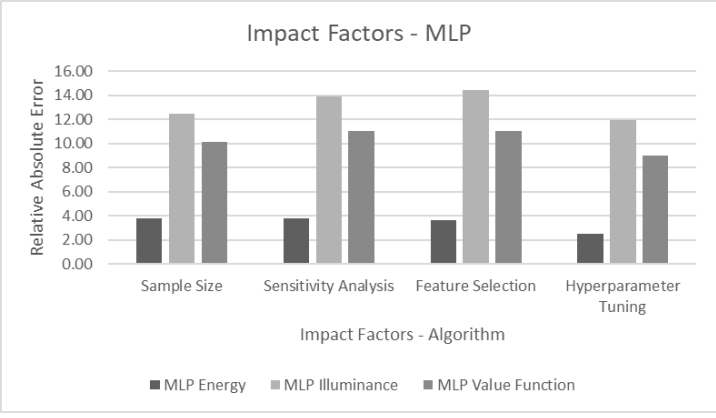


Figure 5:22 Evaluation of impact factors for one algorithm across experiments.

We then create a consolidated dataset with the sample size results from each experiment for all algorithms and for all objectives (Table 5:2). We use a one-way ANOVA test on the *Algorithm* column of this data set to determine if there are significant differences between algorithm performance. We then use a Tukey HSD post-processing test to identify the best performing algorithm. We use a one-way ANOVA test on the *Experiment ID* column of this dataset to determine if there are significant differences in performance

between the experiments. This will indicate if there is a significant difference in performance due to design space complexity.

Experiment ID	Algorithm	Objective	RAE

Table 5:2 Structure of consolidated dataset.

5.5 General Experimental Method

This section describes the steps of the experimental method to be used for all reference projects. This involves describing the training datasets, defining the design spaces with their parameter value ranges and calculating the value functions for multi-discipline analysis. The general experimental method also describes how to assess the impacts of sample size, sensitivity analysis, feature selection and hyperparameter tuning on machine learning algorithm performance. Doing this in one place will avoid repetition.

The reference projects will seek to answer the challenges related to implementing parametric analysis in practice. The Residential Tower seeks to answer the speed and accuracy question. What is the minimum sample size that will allow for acceptable accuracy (defined for this study as less than 5 – 10% RAE)? The Mid-Rise Towers, in addition to speed and accuracy questions, investigates the accuracy of machine learning algorithms under complex design space conditions. We perform eight different parametric analyses for the different geometric conditions of the project and investigate the impact of this design space complexity on the performance of the machine learning algorithms. For the Students' Residence, in addition to speed and accuracy, we investigate the impact of geometric variation on the performance of machine learning algorithms. Three design alternatives will be simulated and used to train machine learning algorithms.

In this section we use a simple example called *Sprout Space* to guide the discussion. Sprout Space (Figure 5:23) is a Perkins&Will research initiative focused around developing high performance modular and portable classrooms. The prototype is a 1,000 square foot pre-engineered and pre-built design aimed, in part, at solving the problems of poor daylighting, views, and energy efficiency experienced in portable classrooms.

In conducting a parametric analysis of Sprout Space, six variables were selected with parameter value spaces ranging in size from 4 - 12 (Table 5:3), giving 64,800 design alternatives. This cannot be computed in reasonable time on generally accessible high specification hardware. At 2 minutes per simulation (daylight and energy) this would take 90 days to execute.

In accordance with the scope of this study we consider only energy and daylighting as multi-discipline objectives. Energy is measured using Total Energy Use Intensity (TEUI) and daylighting is measured using Daylight Factor as the performance metric.

Parameter	Parameter Value Space	Number
Orientation	0, 30, 60, 90, 120, 150, 180, 210, 240, 270,300, 330	12
Window Width	10, 15, 20, 25, 30	5
Overhang Depth	0, 1, 2, 3, 4, 5	6
Roof Angle	1, 2, 3, 4, 5, 6, 7, 8, 9	9
Offset	4, 8, 12, 16, 20	5
Construction Assemblies	GFRC, SIP, CLT, PCP	4
	Total	64,800
GFRC = Glass Fiber Reinforced Concrete; SIP = Structural Insulated Panels; CLT = Cross Laminated Timber; PCP = Precast Concrete Panels		

Table 5:3 Design space for the Sprout Space project.



Figure 5:23 (Perkin&Will, 2019) Sprout Space example in Vancouver, British Columbia.

5.5.1 Data Description

We describe the machine learning datasets by a table of input and output parameters with an indication of parameter data types, units and a brief description. The data description for Sprout Space is shown in Table 5:4.

Input	**Data Type**	**Units**	**Description**
Orientation	Numerical - Discrete	(◦)	Direction of the building in relation to true north.
Window Width	Numerical - Discrete	feet	The width of the window opening
Overhang Depth	Numerical - Discrete	feet	The depth of the projecting roof
Roof Angle	Numerical - Discrete	(◦)	The angle of the roof in relation to the ground
Offset	Numerical - Discrete	feet	The dimension the building geometries are staggered by
Construction Assemblies	Categorical	-	The materials of the exterior walls
Output	**Data Type**	**Units**	**Description**
Total Energy Use Intensity	Numerical - Continuous	kWh/m^2/year	Total energy consumed in one year divided by the gross floor area
Daylight Factor	Numerical - Continuous	%{lux/m^2)	The ratio of the light level inside to the light level outside the building
Direct Line of Sight	Numerical - Continuous	%{m^2)	Percentage of the floor area with exterior view
Quality Views	Numerical - Continuous	%(m^2)	Percentage of the floor area that allows the view of specified elements of the exterior
Carbon Dioxide Emission	Numerical - Continuous	metric tons	Estimated construction cost

Table 5:4 Sprout Space data description.

5.5.2 Design Space Definition

The process of defining the design space with its parameter value ranges involves:

1. The designer describes design intent and requirements to the energy modeler.
2. The energy modeler develops an analytical model including a data specification.
3. The designer and energy modeler define ranges for the parameters in the data specification.

The Sprout Space design space was shown in Table 5:3. The design space has 64,800 combinations which is too large to simulate. Design space reduction is required. A DOE is used to create a reduced sample of 1000 instances for simulation. The steps for creating the reduced sample involve:

1. Use input parameter value ranges to create a 1000 instance DOE in JMP software. The DOE creates a reduced space-filling sample of the design space. A space-filling sample is required to ensure random selection along all axes of the multi-dimensional design space.

2. Simulate the reduced design space to create a *Simulation Results* file in csv format.

3. Use the *Simulation Results* file to create several *Training Set* csv files, with different sample sizes, for each objective of the experiment.

4. Build predictive models by training machine learning algorithms on the *Training Set* files.

5. Use the predictive models to predict rather than simulate the full design space of 64,800 alternatives.

5.5.3 Calculate the Value Function

Since the parametric analysis is multi-objective (and multi-disciplinary) we use the weighted sum method for comparative analysis across objectives. The process of calculating the value function using the weighted sum method involves:

1. Normalize the output values using the normalization equation $x = (x - x_{min}) / (x_{max} - x_{min})$.

2. If the objectives are conflicting, like daylight and energy, then invert one of them by subtracting the normalized value from 1. In the case of daylight and energy we invert energy so that we are seeking to maximize the value function.

3. Establishing appropriate weights for the normalized objectives. In this study we assume equal weights for all objectives.

4. We calculate the value function by summing the normalized, inverted and weighted objectives into a single objective.

5.5.4 Algorithm Default Values

Machine learning algorithms have several settings. Before we can perform impact studies it is useful to understand the default values of these settings. In these experiments we will modify some hyperparameter settings to investigate if this can improve performance beyond the default. The three algorithms used for the experiments are Weka implementations of ANNs, SVMs and DTs. The default values of the algorithms in our experiments are based on the Weka defaults shown in Figure 5:24, Figure 5:25 and Figure 5:26. The meaning of the various hyperparameters is available in the Weka java documentation (University of Waikato, 2019).

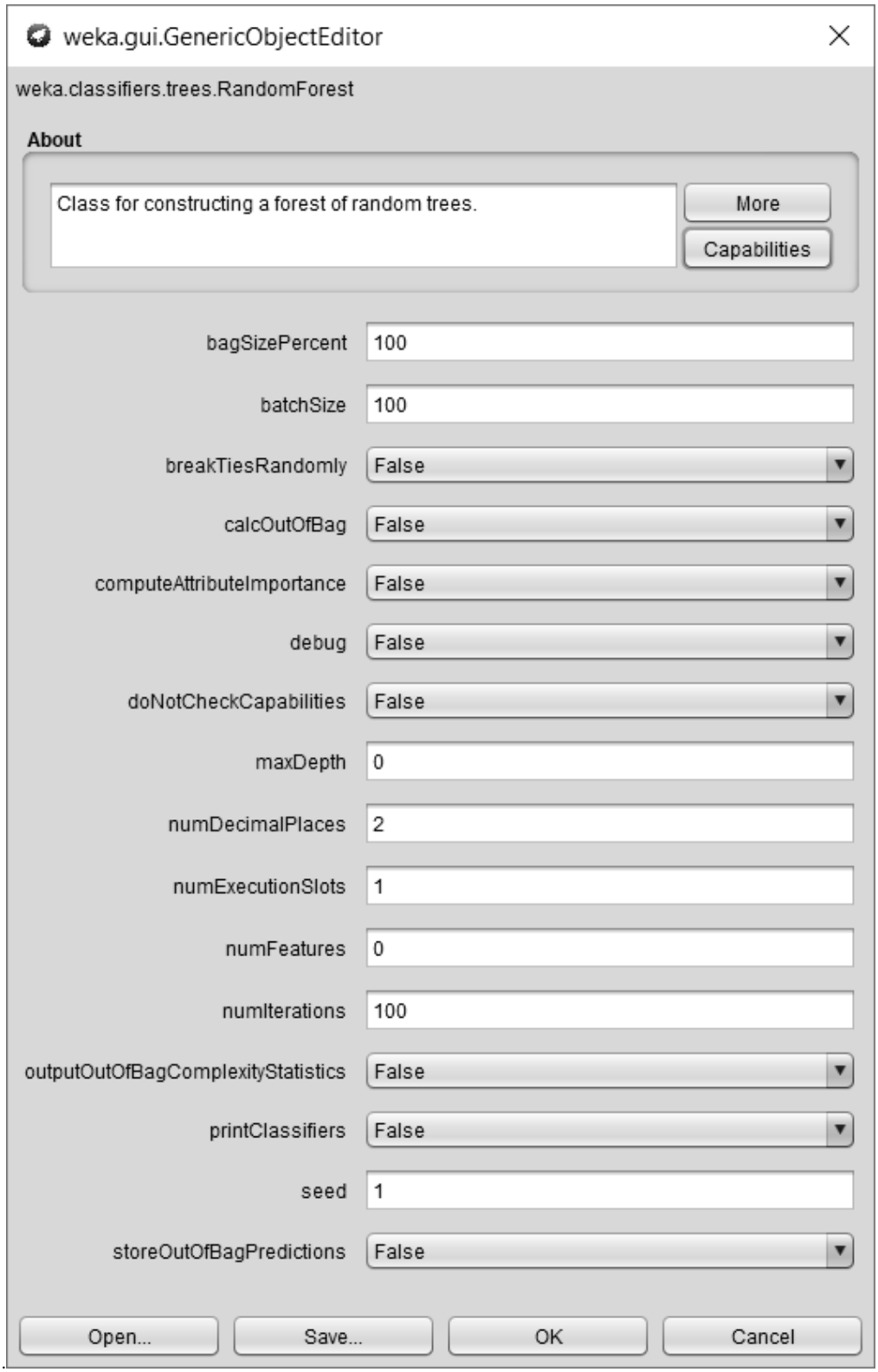


Figure 5:24 Weka Random Forest defaults.

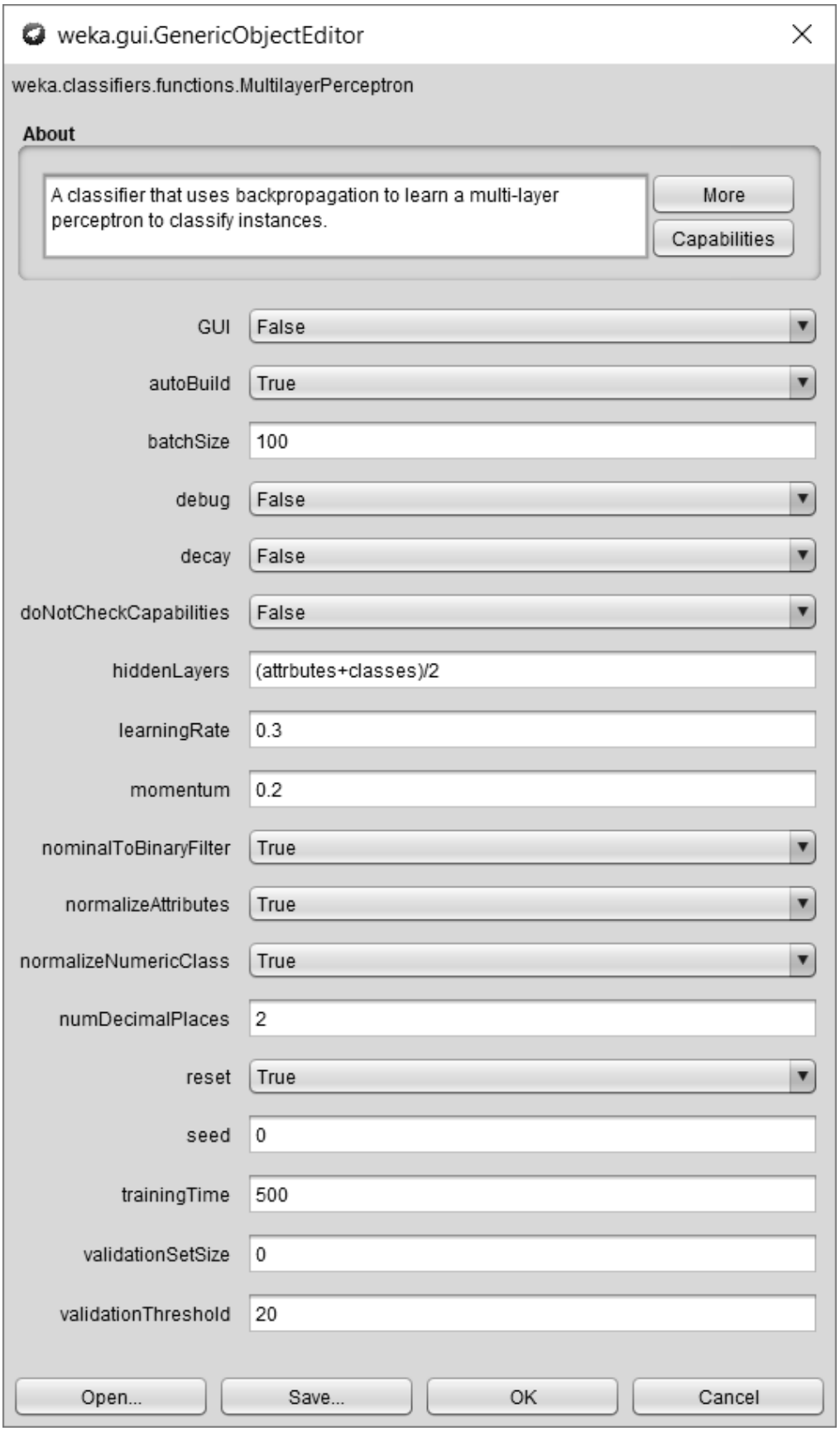


Figure 5:25 Weka Multi-Layer Perceptron defaults.

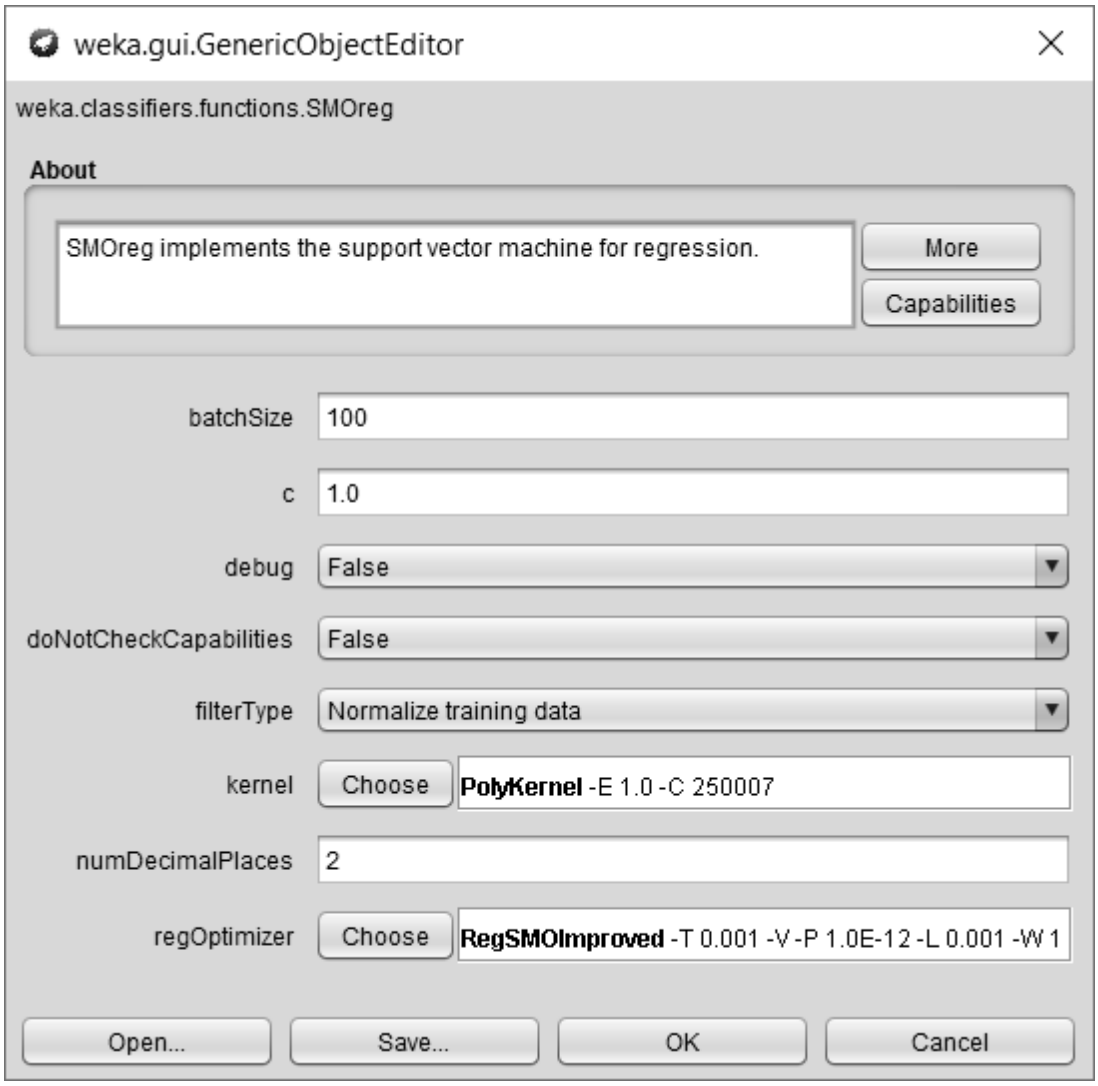


Figure 5:26 Weka SMOreg defaults.

5.5.5 Impact of Sample Size

The specific steps for assessing the impact of sample size on algorithm performance are calculating the value function, preparing training dataset files for the different machine learning experiments, running the machine learning algorithms on the training datasets and graphing the results for interpretation.

Calculating the Value Function

1. Using the Simulation Results file calculate the maximum and minimum value of Energy (TEUI) and Illuminance.

2. Normalize Energy and Illuminance and then invert Energy.

3. Assume equal weight distribution for Energy and Illuminance and calculate the Value Function VF = (0.5 × TEUI) + (0.5 × Illuminance).

Prepare training datasets

From the Simulations Results file prepare three Training Master files in csv format with 1000 samples each (Table 5:5).

Training Master File	Description
SS_1000_Energy.csv	Use TEUI as training class for sample size impact.
SS_1000_Illuminance.csv	Use Illuminance as training class for sample size impact.
SS_1000_Value_Function.csv	Use Value Function as training class for sample size impact.

Table 5:5 Sprout Space training master files.

Using a custom MS Excel macro prepare 10 *Training Set* files in csv format from each of the three *Training Master* files. The 10 *Training Set* files will randomly sample the *Training Master* file with sample sizes ranging from 100 to 1000. Ultimately the training sets will comprise several files as shown in Table 5:6.

Sprout Space Energy Analysis	Sprout Space Illuminance Analysis	Sprout Space Value Function Analysis
SS_1000_Site_Energy.csv	SS_1000_Illuminance.csv	SS_1000_Value_Function.csv
SS_900_Site_Energy.csv	SS_900_Illuminance.csv	SS_900_Value_Function.csv
SS_800_Site_Energy.csv	SS_800_Illuminance.csv	SS_800_Value_Function.csv
SS_700_Site_Energy.csv	SS_700_Illuminance.csv	SS_700_Value_Function.csv
SS_600_Site_Energy.csv	SS_600_Illuminance.csv	SS_600_Value_Function.csv
SS_500_Site_Energy.csv	SS_500_Illuminance.csv	SS_500_Value_Function.csv
SS_400_Site_Energy.csv	SS_400_Illuminance.csv	SS_400_Value_Function.csv
SS_300_Site_Energy.csv	SS_300_Illuminance.csv	SS_300_Value_Function.csv
SS_200_Site_Energy.csv	SS_200_Illuminance.csv	SS_200_Value_Function.csv
SS_100_Site_Energy.csv	SS_100_Illuminance.csv	SS_100_Value_Function.csv

Table 5:6 Sprout Space training set files.

Run machine learning algorithm experiments

Use the Weka Experimenter to run three experiments, one for each metric: TEUI, Illuminance, and Value Function.

1. Open the Weka Experimenter and select Simple Experiment Configuration Mode.
2. Create a new experiment and set the Results Destination folder location.
3. The Experiment Type is Regression with 10-fold cross validation and 10 iterations.
4. For each experiment add the paths to the Training Set files for the respective metric.
5. Add the three algorithms of the study: Multi-Layer Perceptron, SMOReg and Random Forest.

Run the experiments and then, using the RAE metric, report the results and graph them in MS Excel for comparative analysis.

5.5.6 Impact of Sensitivity Analysis

Sensitivity Analysis (SA) investigates the input parameters that have the greatest effect on the output in question. From a design point of view SA can help guide designers focus their decision making on parameters that are most likely to influence design outcomes. From a data modeling point of view SA simplifies data models making them faster to compute, easier to manipulate and interpret and potentially more accurate. We therefore investigate whether using SA to reduce the number of input variables results in design spaces that create better performing machine learning models.

The approach to SA in this study will make use of the Assess Variable Importance feature in JMP. A DOE is generated with the input parameters as the *X* variables of the DOE model and the output parameter as the *Y* variable. The *Y* variable is initially blank. After simulation the values of the output parameter are imported into the *Y* variable in JMP. The data model is then generated, and the Assess Variable Importance graph generated. It indicates the level of influence each of the *X* variables has on the *Y* variable. We can exclude variables that have a low level of influence on the *Y* variable.

Perform Sensitivity Analysis

1. Create a 1000 instance DOE in JMP and export to MS Excel to use as input for dynamic simulation. Obtain the simulation results for each performance metric and import into JMP.

2. Generate a data model for each performance metric and produce the Assess Variable Importance graph. Use the graph to order the input parameters in order of importance.

3. Edit the Training Master files from Table 5:5 to obtain three new Training Master files for the SA impact study (Table 5:7).

4. Using each of the three master files run sequential machine learning experiments removing the least important input parameters, in the order from 2 above, and recording the resulting accuracy of the machine learning algorithms.

Training Master Files	Description
SS_1000_SA_Energy.csv	Use TEUI as training class for sensitivity analysis impact.
SS_1000_SA_Illuminance.csv	Use Illuminance as training class for sensitivity analysis impact.
SS_1000_SA_Value_Function.csv	Use Value Function as training class for sensitivity analysis impact.

Table 5:7 Sprout Space sensitivity analysis master files.

5.5.7 Impact of Feature Selection

In this study we will use a wrapper method from the Weka Attribute Selection interface. Wrapper methods involve two aspects: evaluation and search. Evaluation will be done using the WrapperSubsetEval algorithm. Search will be done using the BestFirst algorithm with the backward method. Search methods include forward search, backward search, bi-directional search and exhaustive search. Forward searches start with no attributes and then add attributes; backward searches start with all attributes and then remove attributes. Bi-directional searches start with a random number of attributes and add or remove attributes depending on what produces improved performance. Exhaustive search systematically searches all combinations of attributes but is obviously computationally expensive.

Perform Feature Selection

1. Use the Select Attributes panel in Weka with the WrapperSubsetEval algorithm and the BestFirst algorithm with backward search to identify the best input parameters for each of the three Training Master files (Table 5:8).

2. Sequentially select each of MLP, SMOReg and RF as parameters of the WrapperSubetEval algorithm.

3. Using each of the three master files run a machine learning experiment using only the best input parameters for each algorithm and each output and record the resulting accuracies.

Training Master Files	Description
SS_1000_FS_Energy.csv	Use TEUI as training class for feature selection impact.
SS_1000_FS_Illuminance.csv	Use Illuminance as training class for feature selection impact.
SS_1000_FS_Value_Function.csv	Use Value Function as training class for feature selection impact.

Table 5:8 Sprout Space feature selection master files.

5.5.8 Impact of Hyperparameter Tuning

In order to understand the impact of hyperparameters on algorithm performance we first need to identify the key hyperparameters for each algorithm. Machine learning algorithms can have dozens of hyperparameters many of which may have minimal impact on a given problem. We therefore use supporting literature to identify the key hyperparameters relevant to our domain of study.

Having identified the key hyperparameters we will perform tuning tests for each algorithm and objective to identify the impact on accuracy. At this point we are not expecting to optimize the algorithm performance but rather to identify which hyperparameters have the biggest impact. The impact of the hyperparameters is assessed by performing experiments in the Weka Experiment Environment. For each algorithm (SVM, ANN, RF), and for each metric (Site Energy, Illuminance, Value Function) we sequentially identify the optimal value for each hyperparameter.

Perform Hyperparameter Tuning

1. Use the Weka Experiment Environment to insert the three Training Master files as input datasets.
2. Create a set of experiments for each of the machine learning algorithms (MLP, SMOReg, RF).
3. Create an experiment for each value of each hyperparameter with ranges established from literature sources as well as trial and error.
4. Run the experiments and retrieve the results from the Analyzer. The results report the RAE for each algorithm with the error corresponding to the best performing value of each hyperparameter.
5. The result of 4 above can be compared to the accuracy of the algorithm when run with default values to assess the impact of tuning hyperparameters.

Training Master Files	Description
SS_1000_HP_TEUI.csv	Use TEUI as training class for hyperparameter impact.
SS_1000_HP_Illuminance.csv	Use Illuminance as training class for hyperparameter impact.
SS_1000_HP_Value_Function.csv	Use Value Function as training class for hyperparameter impact.

Table 5:9 Sprout Space hyperparameter tuning master files.

Ranges of Hyperparameter Values

Table 5:10 shows the hyperparameters and ranges we will use for the MLP algorithm. *hiddenLayers* refers to the architecture of the neural network. (2,3), for example, represents two hidden layers with 2 neutrons and 3 neutrons in the layers respectively. *learnRate* and *momentum* refer to the learning rate and the momentum as discussed in Section 4.2.3.

Hyperparameter	Description
hiddenLayers	(10,10), (20,20), (30,30), (40,40), (50,50)
learnRate	0.001, 0.01, 0.1, 0.2, 0.3
momentum	0.2, 0.4, 0.6, 0.8

Table 5:10 Hyperparameter ranges for the MLP algorithm.

Table 5:11 shows the hyperparameters and ranges we will use for the SMOReg algorithm.

Hyperparameter	Description
complexity	0.001, 0.01, 0.1, 1,10
kernel	Polykernel, NormalizedPolyKernel, RBFkernel
epsilonParameter	0.001, 0.01, 0.1,1

Table 5:11 Hyperparameter ranges for the SMOReg algorithm.

Table 5:12 shows the hyperparameters and ranges we will use for the Random Forest algorithm. *numIterations* refers to the number of classifiers in the Random Forest; *numFeatures* refers to the number of parameters to randomly investigate; *maxDepth* refers to the maximum depth of the trees in the Random Forest.

Hyperparameter	Description
numIterations	100, 300, 500, 700
numFeatures	2, 4, 6, 8
maxDepth	50, 100, 150, 200

Table 5:12 Hyperparameter ranges for the Random Forest algorithm.

Chapter 6 Residential Tower Experiment

This chapter performs machine learning experiments on data from the Residential Tower reference project. We begin by providing a project description based on the DSC methodology. We then make use of the general experimental method discussed in Chapter Five. We discuss the impact of sample size, sensitivity analysis, feature selection and hyperparameter tuning on algorithm performance. We develop and analyze the experimental results using comparative graphs. We find that machine learning algorithms are effective for all objectives. MLP is the best performing algorithm and hyperparameter tuning is the most influential impact factor.

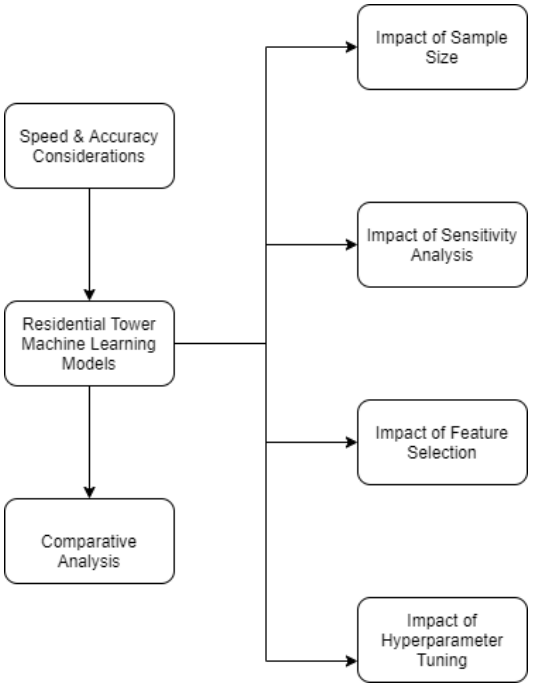


Figure 6:1 Diagram Summary of Chapter Six.

6.1 Introduction

The Residential Tower (Figure 6:2) is a commercial residential development situated in the neighborhood of East Vancouver, consisting of a highly insulated tower clad in clear glass with shadow box assemblies. The tower base features a series of retail spaces and a street-oriented lobby. The designers proposed a community library on the second and third floors, while at the top of the tower they proposed amenity spaces housed in a double height space featuring a communal lounge, a mezzanine library with views to the city, mountains and water, and an outdoor pool area.

The design was inspired both by the character of the neighborhood as well as by the heritage of Northwest modernist tradition of lightweight structural systems, nautical concepts of outriggers and tension elements, and contemporary concepts of prefabrication and plug-and-play assemblies. The design team proposed lightweight balconies to create outdoor living rooms. They intended the balconies to provide full solar shading during peak demand load hours. The balconies also featured planters for privacy and additional screening elements designed as plug-and-play elements of the facade. The team proposed constructing the balconies from lightweight steel outriggers suspended by a network of steel cables, minimizing the thermal bridging and creating a diaphanous scrim for the tower. To create design interest, they proposed a dramatic horizontal shift in geometry at mid-tower height, as seen in Figure 6:2.

Before we can state the questions explored by the parametric analysis process, it is useful to provide some of the regulatory background to the Residential Tower. According to the City of Vancouver Green Building for Rezoning standards (2017), which all rezoning applications need to meet, projects have a choice between pursuing Near Zero Emissions Building (NZEB) or Low Emissions Green Building (LEGB). The NZEB pathway requires teams to design projects to Passive House (PH) or an alternative similar standard. The key metrics of this pathway are space heating demand at 15 kWh/m^2/year, 0.6 ACH @ 50 Pascals pressurization, and maximum 60 kWh/m^2/year renewable primary energy demand.

The LEGB pathway requires LEED Gold Certification (US Green Building Council, 2019) or an alternative holistic green building rating system. For residential high-rise towers, Total Energy Use Intensity (TEUI) cannot exceed 120 kWh/m^2/year, Total Energy Demand Intensity (TEDI) 32 kWh/m^2/year and a Green House Gas Inventory (GHGI) of 6. However, it also has a list of additional requirements, including requirements for whole-building airtightness testing, requirements for enhanced commissioning and about a dozen more provisions. Therefore, the design team wished to understand which of these pathways was most feasible for their design objectives. Were the stringent NZEB Passive House energy requirements attainable or was it better to focus on the less stringent, but more numerous requirements of the LEGB pathway? In addition, the design team sought to optimize energy, daylight and comparative cost considerations.

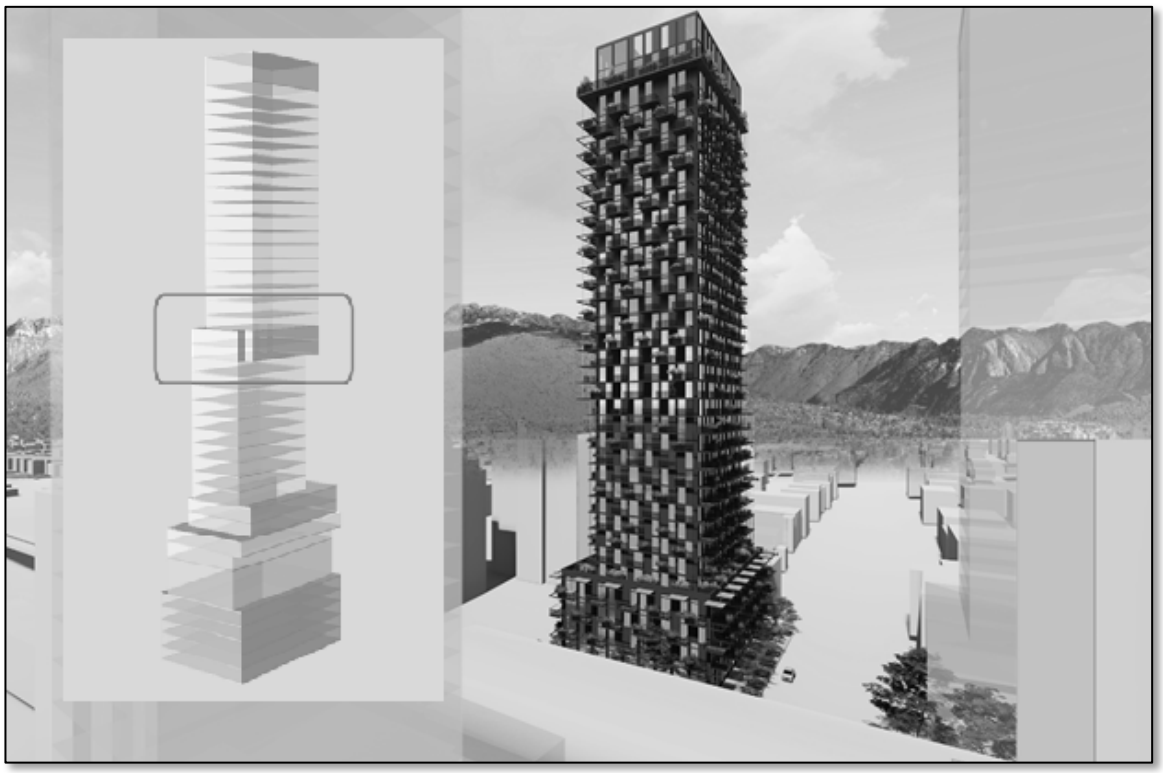


Figure 6:2 (Perkins&Will, 2019) RT design impression. Inset: proposed tower shift.

6.2 DSC Process

Once the design team had settled on the overall program and massing of the building, they engaged the DSC process. The designers were looking for guidance on how to modify the form and envelope of the massing to optimize energy and daylight performance. They were also looking for guidance on the impact of the building form shift on building performance, as well as the appropriate dimensions and configurations of the balconies and facade elements.

In using the DSC framework to structure the description of the process we will proceed as follows:

1. *Objective Definition*: what are the questions the design team wished to answer?
2. *Alternative Generation*: what is the design space they developed to answer the questions?
3. *Impact Analysis*: how were the results obtained?
4. *Value Assessment*: how were the questions answered?

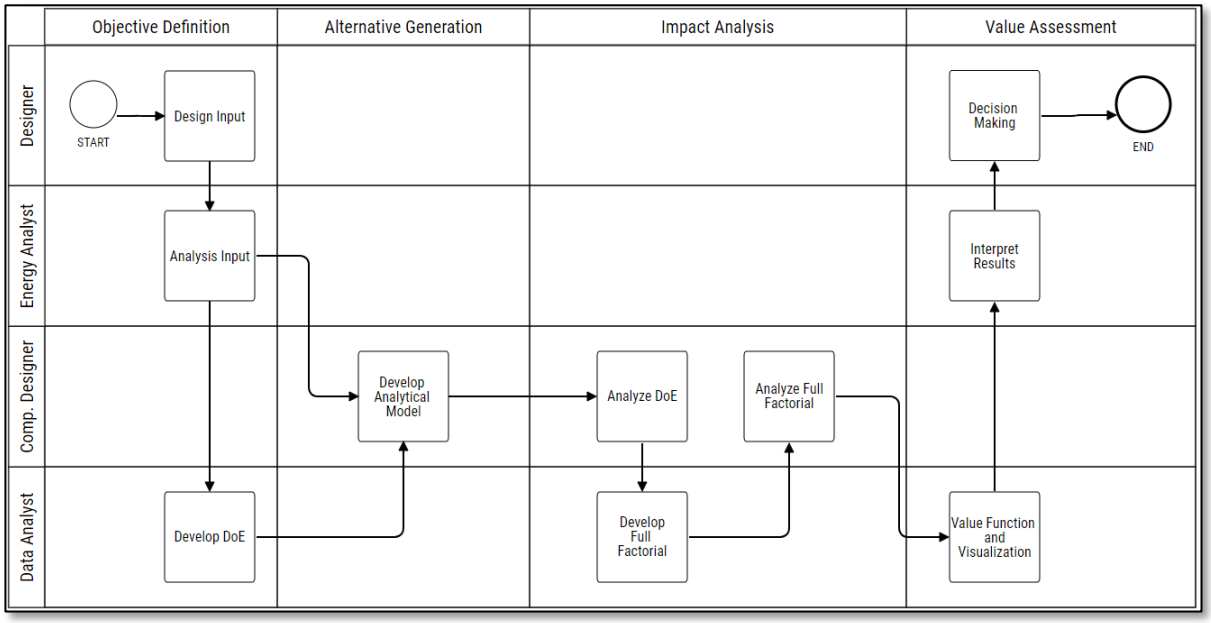


Figure 6:3 DSC process for the Residential Tower project.

6.2.1 Objective Definition

The design team asked three questions. First, which regulatory pathway should they follow? In other words, could the building perform at 15 kwh/m^2/year heating demand as required by the PH standard or was this out of the question? Second, what is the optimal balcony depth on each façade orientation? They wanted balcony depths that would allow them to meet their energy targets while allowing maximum daylighting? In addition, balconies carried the cost of thermal bridging effectively reducing the R-value of the envelope. Third, the design team and the client wished to use WWR of 50% on all facades whereas the building envelop consultant counseled a maximum WWR of 42% on all facades. The design team looked to the DSC process to advise on whether their desired target was achievable.

6.2.2 Alternative Generation

The energy modeler and the design team typically specify the design space in the DSC process. This involves identifying the input parameters needed for analysis based on design intent, project description and model geometry. The ranges of these input parameters are then selected based on industry standards and the objectives of parametric analysis (Figure 6:4). The key input parameters for the Residential Tower were identified as WWR on all orientations, effective Wall R-Value, Chi-Value, Balcony Depth on all orientations and SHGC of glazing. Industry practice, client request and consultant recommendation together with the need to optimize energy and daylight helped determine the range of WWR. Effective Wall R-Value was used due to thermal bridging caused by exposed balconies. This effective Wall R-Value was computed with the help of the Chi-Value parameter. In addition to the effect on energy performance, Balcony Depth was also influenced by daylighting considerations as well as costs.

Window to Wall ratio (%)	Effective U value (W/m2K)	Win U/SHGC	Balcony/Shades depth (ft)
30	0.40	0.85/0.3	0
	0.30	0.85/0.5	
	0.25	0.85/0.7	5
	0.33	1.4/0.3	
	0.23	1.4/0.5	10
	0.19	1.4/0.7	
40	0.41	0.85/0.3	0
	0.32	0.85/0.5	
	0.27	0.85/0.7	5
	0.34	1.4/0.3	
	0.24	1.4/0.5	10
	0.19	1.4/0.7	
50	0.44	0.85/0.3	0
	0.35	0.85/0.5	
	0.30	0.85/0.7	5
	0.35	1.4/0.3	
	0.25	1.4/0.5	10
	0.20	1.4/0.7	
60	0.48	0.85/0.3	0
	0.39	0.85/0.5	
	0.34	0.85/0.7	5
	0.36	1.4/0.3	
	0.27	1.4/0.5	10
	0.22	1.4/0.7	
70	0.55	0.85/0.3	0
	0.45	0.85/0.5	
	0.40	0.85/0.7	5
	0.39	1.4/0.3	
	0.29	1.4/0.5	10
	0.25	1.4/0.7	

Figure 6:4 (Perkins&Will, 2019) Residential Tower ranges of input variables.

Initially the team undertook DSC on the Residential Tower project at two levels of the tower, below and above the proposed mid-height shift. The team modeled a representative level for the levels 14 to 28 above the shift, and another representative level between levels 3 to 13 below the shift. Although the program configuration was slightly different above and below the shift, simulation results found no significant difference in performance. In addition, because this is a tower project, the team assumed that the effect of different adjacency conditions at the exposed roof and ground level slab would be minimal. Therefore, generalization from the typical level was an adequate approximation.

6.2.3 Impact Analysis

The energy modeler developed the zone geometry (Figure 6:5) and zone inputs for the analytical model. Given the proposed variables and input ranges the initial design space included about 2 million alternatives. Reducing the design space to more manageable size (sampling ratio less than 1:100) required a stratified design space reduction. The design team first used design judgement to reduce the design space to 14,400 instances and then used a DOE-based heuristic method to reduce the design space to 1296 simulations.

For the DOE heuristic, first they designed an experiment of the typical level, or slice, with 64 runs. They simulated the 64 runs and used the results from the JMP software Prediction Profiler (Figure 6:6) to estimate optimized values for all the input parameters. Then, they developed a full factorial design space of 1296 simulations centered on the estimated optimum values. This design space was simulated for both energy and daylighting (in about 2 days) and the results processed and posted to the Design Explorer™ (Thornton Tomasetti, 2019) PCP visualization interface (Figure 6:7).

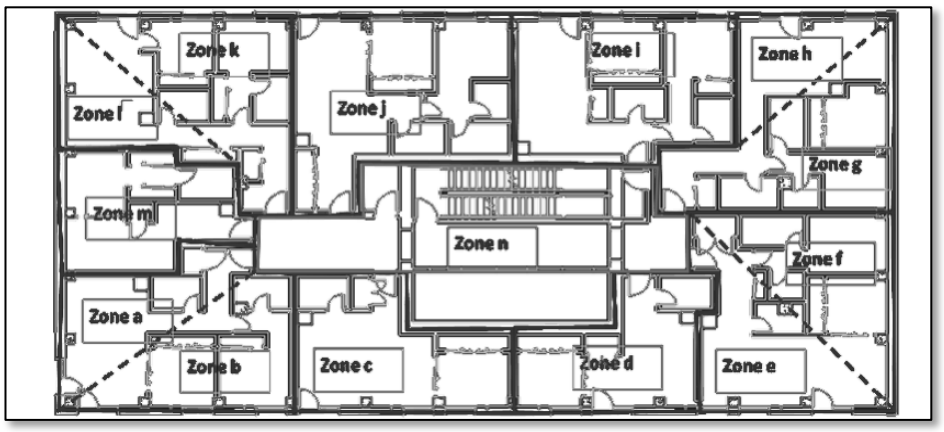


Figure 6:5 (Perkins&Will, 2019) Residential Tower zoning on typical level.

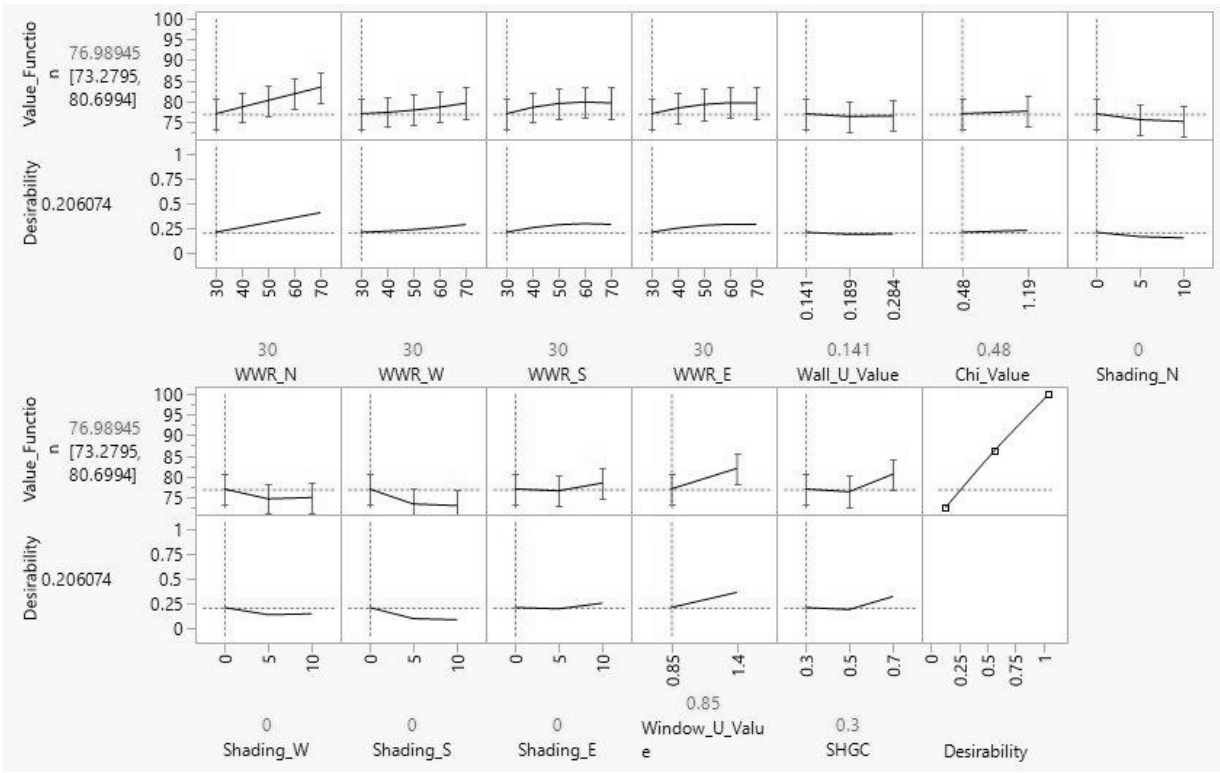


Figure 6:6 Residential Tower Prediction Profiler from JMP.

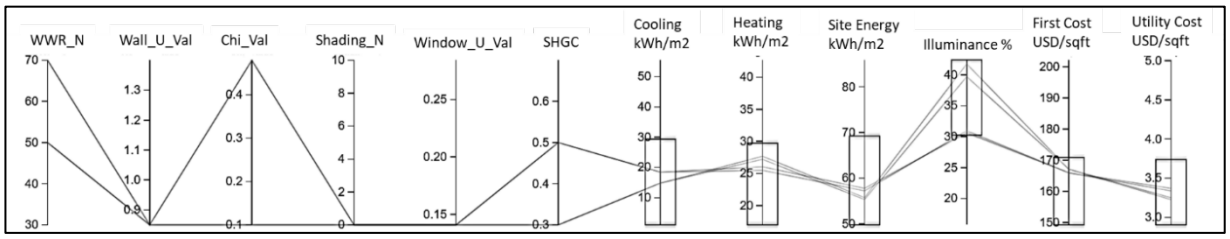


Figure 6:7 Residential Tower partial PCP.

6.2.4 Value Assessment

The findings from the Residential Tower indicated that both PH heating demand and cooling demand could only be achieved while using an efficient Heat Recovery System (HRV). Both 40% WWR and

50% WWR met the cooling requirement, and both failed to meet the heating requirement without heat recovery. Larger (50%) WWR was obviously superior in terms of daylighting, and the DSC recommendation was that 50% WWR should be further investigated through more detailed modeling. The design team had proposed 5' deep balconies on North and East facades, and 10' deep balconies on the South and West facades. The DSC process found that there was no significant difference in building performance by reducing the depth of balconies along the south and west facades to 5'. Sensitivity Analysis (Figure 6:8) indicated that the most sensitive input parameters were WWR on the North façade, Window U-Values and Balcony Depth on the South façade.

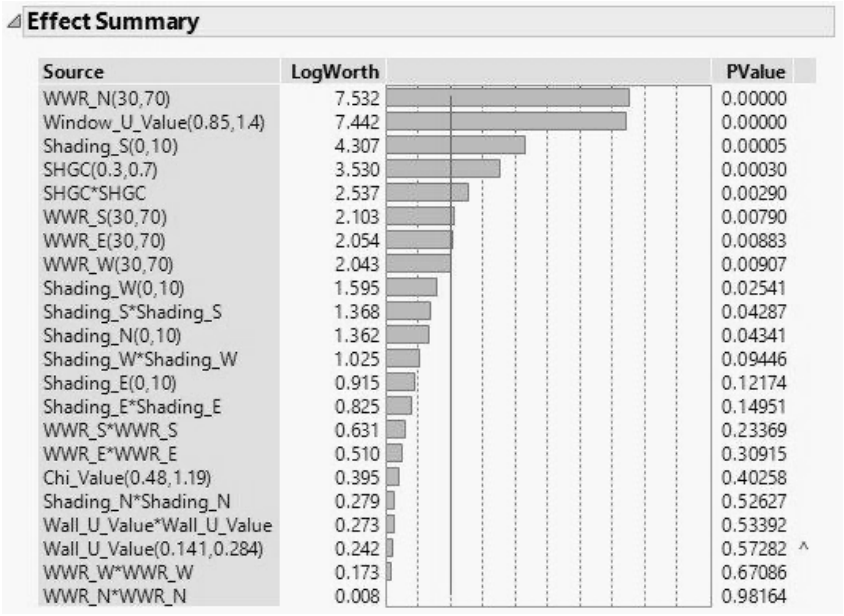


Figure 6:8 Residential Tower sensitivity analysis.

6.2.5 Discussion

Parametric analysis on the Residential Tower answered questions that could not accurately be answered using conventional analysis. It was shown that the PH standard could only be pursued with a high efficiency HRV system. It was also shown that WWR of 50% was feasible, and Balcony Depth was optimized on all facades. The DOE heuristic method developed a design space of 1296 simulations which can run in 43.2 hours. This is an acceptable level of speed for architectural practice.

However, the DOE heuristic method does not report its level of accuracy; it is therefore not an effective method according to our definition of effectiveness. We do not know the error between the actual optimum and the DOE sample optimum. The DOE heuristic method has been evaluated by Perkins&Will Research who find that the DOE heuristic sample mean is significantly superior to the mean from a random sample. However, it would still be important to understand the actual error associated with a data analytical process.

Another shortcoming of the DOE heuristic method is that it did not provide a full design space. Instead it identifies a sample with a high-performance bias. This means that design space exploration is limited. For example, if we wish to identify poor performing alternatives (so we know what to avoid) then the DOE heuristic method fails. Moreover, since sensitivity analysis is performed on a reduced sample of data there is a likelihood that it is less accurate than if it is performed on the full design space.

6.3 Method

6.3.1 Data Description

The general experimental method used for this reference project was described in Chapter Five. The data used for the Residential Tower machine learning experiments is described in Table 6:1. Note that Effective Wall U-Values are computed by taking into account thermal bridging using the Chi-Value.

Input	**Data Type**	**Units**	**Description**
WWR_N	Numerical - Discrete	%	Window to wall ratio on the North façade
WWR_W	Numerical - Discrete	%	Window to wall ratio on the West façade
WWR_S	Numerical - Discrete	%	Window to wall ratio on the South façade
WWR_E	Numerical - Discrete	%	Window to wall ratio on the East façade
Wall U Value	Numerical - Discrete	$W/m^2 K$	Wall thermal transmittance value
Chi Value	Numerical - Discrete	W/K	Point Heat Transmittance Coefficient
Shading N	Numerical - Discrete	feet	Balcony depth on the North façade
Shading W	Numerical - Discrete	feet	Balcony depth on the West façade
Shading S	Numerical - Discrete	feet	Balcony depth on the South façade
Shading E	Numerical - Discrete	feet	Balcony depth on the East façade
Window U Value	Numerical - Discrete	$W/m^2 K$	Window thermal transmittance value
SHGC	Numerical - Discrete	-	Solar Heat Gain Coefficient
Effective U N	Numerical - Continuous	$W/m^2 K$	Effective wall thermal transmittance value - North façade
Effective U W	Numerical - Continuous	$W/m^2 K$	Effective wall thermal transmittance value - West façade
Effective U S	Numerical - Continuous	$W/m^2 K$	Effective wall thermal transmittance value - South façade
Effective U E	Numerical - Continuous	$W/m^2 K$	Effective wall thermal transmittance value - East façade
OUTPUT	**Data Type**	**Units**	**Description**
Cooling	Numerical - Continuous	kWh/m^2	Cooling load
Heating	Numerical - Continuous	kWh/m^2	Heating load
Site Energy	Numerical - Continuous	kWh/m^2	Total Site Energy
Illuminance %	Numerical - Continuous	%	Percentage of usable floor area with 300 - 3000 lux illuminance

Table 6:1 Residential Tower data description.

6.3.2 Design Space Definition

The design space definition of the Residential Tower is shown in Table 6:2. The design space had 14,400 combinations which was too large to rapidly simulate. Design space reduction was therefore required.

Parameter	**Description**	**Range**	**Size**
WWR (all orientations)	Window to wall ratio	30,40,50,60,70	20
Wall U Value	Wall U Value	0.141,0.189,0.284	3
Chi Value	Point heat transmittance coefficient	0.48, 1.19	2
Shading (all orientations)	Balcony shading	0,5,10	20
Window U Value	Window U Value	0.85,1.4	2
SHGC	Solar Heat Gain Coefficient	0.3,0.5,0.7	3
			14,400

Table 6:2 Residential Tower design space description.

6.4 Results

6.4.1 Sample Size

Sample Size experiments were conducted with default values for all algorithms as discussed in Section 5.6.2. As discussed in Section 2.3.3 we define RAE between 5-10% to be effective. In addition, we will describe RAE between 10-15% to be close to effective and RAE between 15-20% to be somewhat effective. The intention of these descriptions is to indicate which results, though not yet effective, are within the range of improvement given that the best average improvement is approximately 10% (11.36% for SMOReg under Hyperparameter Tuning).

The impact of Sample Size on Site Energy was that all algorithms perform better with a large sample (1000 instances) than with a small sample (100 instances) by an average of 7.42%. 100 instances are a reasonable bottom threshold since it is unlikely that a smaller sample size will give effective results. The best performing algorithm was MLP with 1000 instances (RAE 5.42%) while the poorest performing was RF with 100 instances (RAE 27.99%). Only MLP was effective with RAE of 7.62% at sample sizes starting at 200 instances.

The effect of Sample Size on Illuminance was that all algorithms performed better with a large sample size by 15.06% on average. The best performing algorithm was MLP with 1000 instances (RAE 9.58%) while the poorest performing algorithm was RF with 100 instances (RAE 45.85%). Only MLP was effective with RAE of 9.58% at a sample size of 1000.

The effect of Sample Size on the multi-discipline Value Function was that all algorithms performed better with a large sample size by 16.23% on average. The best performing algorithm was MLP with 1000 instances (RAE 10.23%) while the poorest performing algorithm was RF with 100 instances (RAE 45.56%). Only MLP was close to effective (RAE less than 15%) with RAE of 10.23% at a sample size of 1000.

Overall MLP was effective for the Energy and Illuminance objectives and close to effective for the Value Function. SMOReg was somewhat effective (RAE less than 20%) for the Energy objective but not for Illuminance or the Value Function. RF was close to effective for Energy and somewhat effective for Illuminance and the Value Function.

Sample Size improved effectiveness of MLP for the Energy objective. MLP was effective at 200 simulations for Energy. Sample Size did not improve the effectiveness of MLP for Illuminance or Value Function objectives. Sample Size did not improve the effectiveness of SMOReg or RF for any of the objectives.

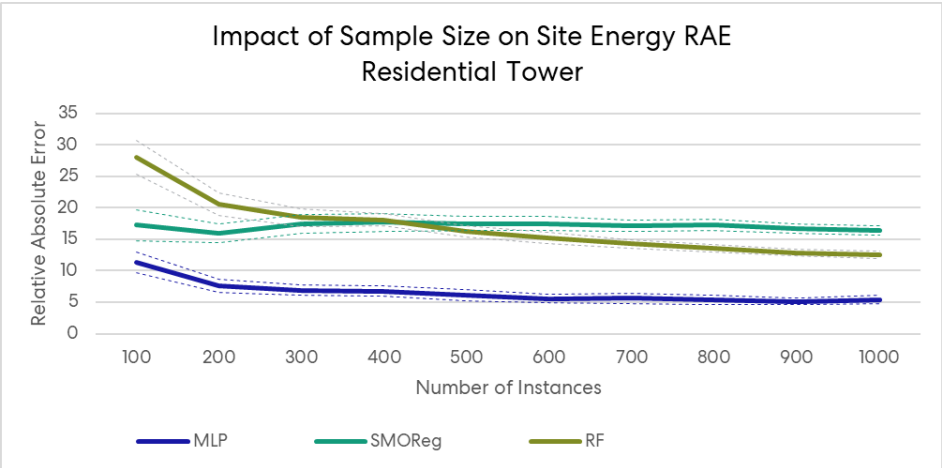


Figure 6:9 Residential Tower impact of Sample Size on the RAE of Site Energy.

Sample Size	100	200	300	400	500	600	700	800	900	1000
MLP	11.34	7.62	6.92	6.79	6.11	5.57	5.62	5.39	5.14	5.42
SMOReg	17.24	15.92	17.39	17.71	17.47	17.51	17.13	17.27	16.66	16.38
RF	27.99	20.55	18.41	18.08	16.24	15.2	14.24	13.62	12.87	12.5

Table 6:3 Residential Tower impact of Sample Size on the RAE of Site Energy.

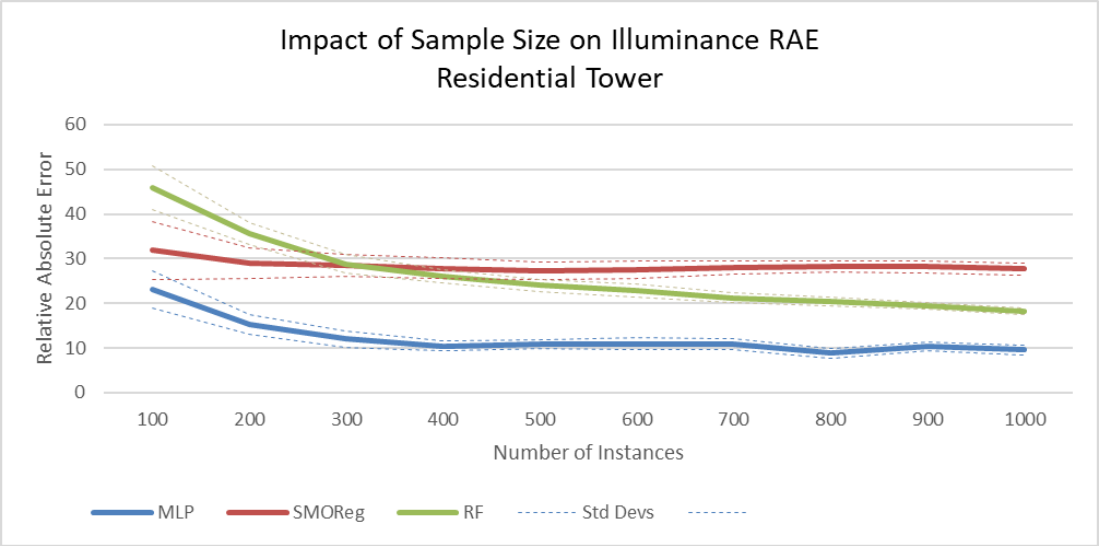


Figure 6:10 Residential Tower impact of sample size on the RAE of Illuminance.

Sample Size	100	200	300	400	500	600	700	800	900	1000
MLP	23.02	15.32	12.07	10.46	10.9	10.98	10.8	8.83	10.4	9.58
SMOReg	31.81	29.07	28.54	27.82	27.27	27.56	28.02	28.32	28.17	27.7
RF	45.85	35.6	28.85	26.05	24.06	22.82	21.26	20.4	19.4	18.23

Table 6:4 Residential Tower impact of sample size on the RAE of Illuminance.

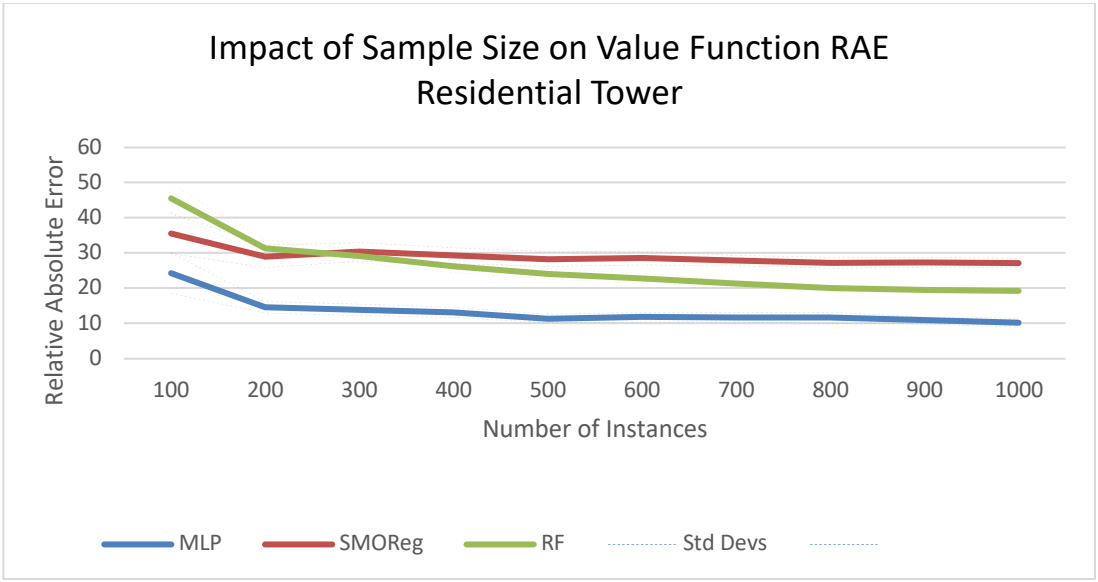


Figure 6:11 Residential Tower impact of Sample Size on the RAE of the Value Function.

Sample Size	100	200	300	400	500	600	700	800	900	1000
MLP	24.31	14.65	13.93	13.16	11.39	11.85	11.76	11.63	10.91	10.23
SMOReg	35.57	29.05	30.43	29.27	28.34	28.57	27.96	27.23	27.44	27.24
RF	45.56	31.4	29.18	26.28	24.11	22.87	21.36	20	19.57	19.27

Table 6:5 Residential Tower impact of sample size on the RAE of the Value Function.

6.4.2 Sensitivity Analysis

The goal of impact factor studies is to help identify influencing factors that can most improve the performance and effectiveness of the machine learning algorithms. Sensitivity Analysis is one of the possible influencing factors. In performing sensitivity analysis, we examine the effect on algorithm performance of considering the parameters ranked in order of sensitivity.

Sensitivity Analysis established that Window U-Value was the most sensitive parameter impacting Site Energy. The order of sensitivity is shown in Figure 6:12. The best performing algorithm was MLP with 12 parameters (RAE 5.25%) while the worst performing algorithm was again MLP with a single parameter (RAE 60.05%). At best MLP performed better than RF by 5.06% and better than SMOReg by 11.09%. RF performed better than SMOReg by 6.03%. Only MLP was effective starting at 9 parameters. RF was close to effective starting at 7 parameters and somewhat effective starting at 6 parameters. SMOReg was somewhat effective starting at 6 parameters.

For Illuminance sensitivity analysis, WWR North was the most sensitive parameter. The order of sensitivity is shown in Figure 6:13. The best performing algorithm was MLP with 10 parameters (RAE 9.35%) while the worst performing algorithm was MLP with a single parameter (RAE 75.09%). At best MLP performed better than RF by 1.96% and better than SMOReg by 18.28%. RF performed better than SMOReg by 16.32%. Only MLP was effective starting at 9 parameters. RF was close to effective between 6 and 10 parameters and only somewhat effective with 11 parameters. SMOReg was not at all effective.

For the Value Function Window U-Value was the most sensitive parameter. The order of sensitivity is shown in Figure 6:14. The best performing algorithm was MLP with 12 parameters (RAE 10.23%) while the worst performing algorithm was MLP with a single parameter (RAE 71.15%). At best MLP performed better than RF by 6.98% and better than SMOReg by 16.46%. RF performed better than SMOReg by 9.48%. None of the algorithms was effective but MLP was close to effective starting at 10 parameters while RF was somewhat effective starting at 10 parameters.

Sensitivity Analysis did not improve the effectiveness of any of the algorithms for any of the objectives.

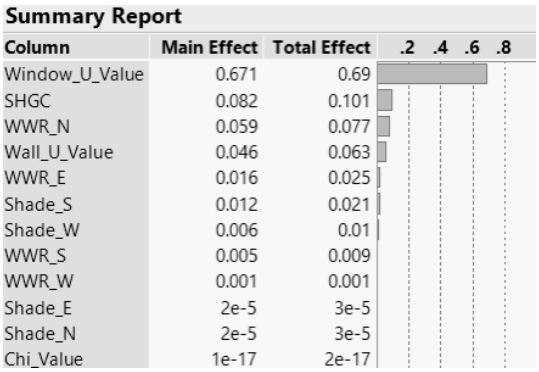


Figure 6:12 Residential Tower Sensitivity Analysis, Energy.

Summary Report

Column	Main Effect	Total Effect
WWR_N	0.61	0.632
WWR_S	0.159	0.18
WWR_E	0.092	0.113
WWR_W	0.038	0.057
Shade_N	0.011	0.021
Shade_S	0.001	0.003
Shade_E	0.001	0.001
Shade_W	2e-4	0.001
Window_U_Value	2e-5	5e-5
Wall_U_Value	2e-5	5e-5
SHGC	5e-6	1e-5
Chi_Value	1e-17	2e-17

Figure 6:13 Residential Tower Sensitivity Analysis, Illuminance.

Summary Report

Column	Main Effect	Total Effect
Window_U_Value	0.57	0.581
WWR_N	0.092	0.104
SHGC	0.085	0.097
WWR_S	0.064	0.076
Wall_U_Value	0.035	0.045
WWR_W	0.025	0.033
WWR_E	0.012	0.018
Shade_W	0.009	0.012
Shade_N	0.007	0.011
Shade_S	0.007	0.009
Shade_E	0.001	0.002
Chi_Value	1e-17	2e-17

Figure 6:14 Residential Tower Sensitivity Analysis, Value Function.

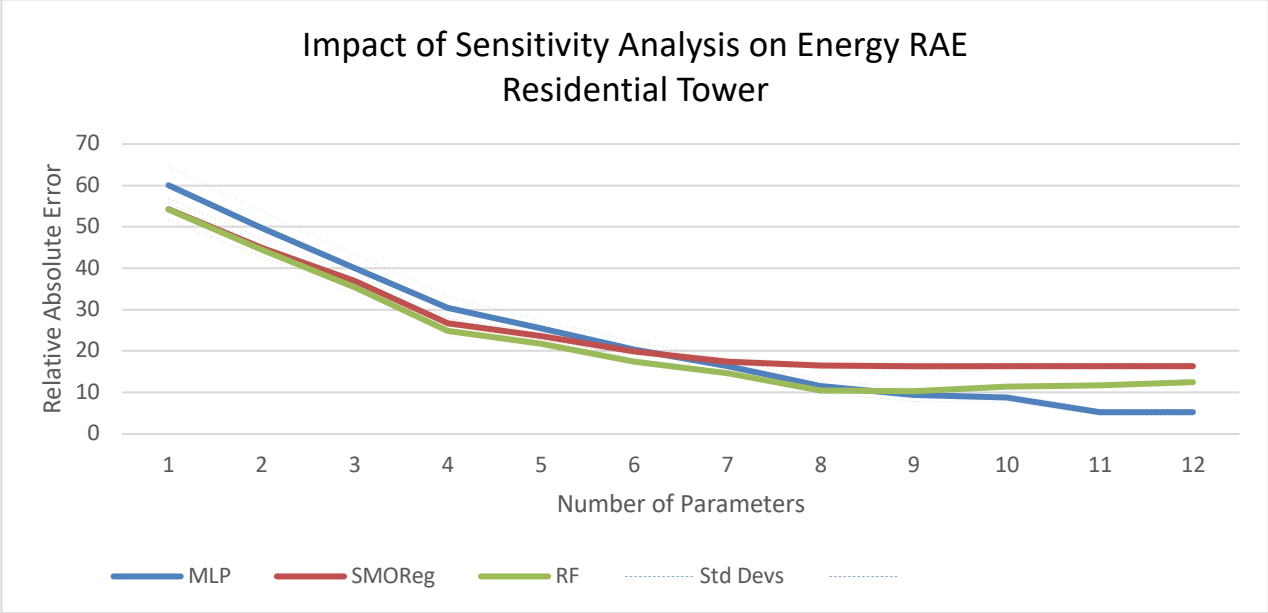


Figure 6:15 Residential Tower impact of Sensitivity Analysis on the RAE of Energy.

		1	2	3	4	5	6	7	8	9	10	11	12
Energy	**MLP**	60.05	49.74	40.05	30.47	25.46	20.35	16.36	11.51	9.46	8.78	5.27	5.25
	SMOReg	54.27	45.02	36.85	26.75	23.58	19.86	17.51	16.45	16.34	16.38	16.38	16.38
	RF	54.2	44.42	35.34	24.85	21.71	17.49	14.61	10.45	10.31	11.36	11.8	12.51

Table 6:6 Residential Tower impact of Sensitivity Analysis on the RAE of Energy.

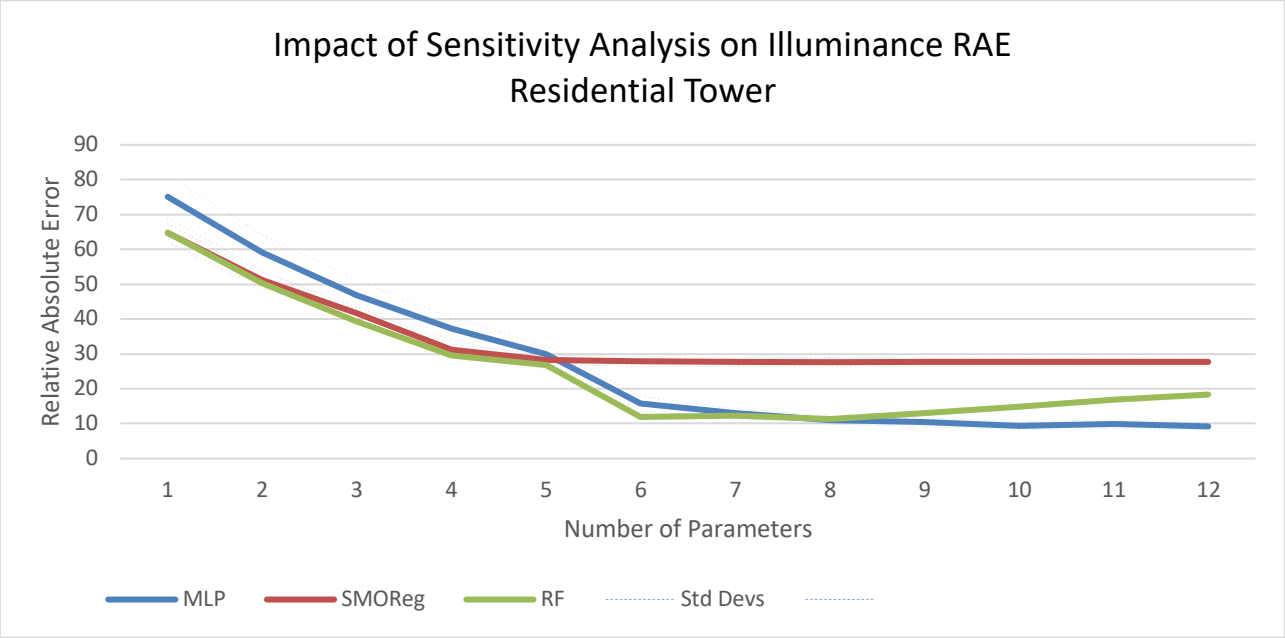


Figure 6:16 Residential Tower impact of Sensitivity Analysis on the RAE of Illuminance.

		1	2	3	4	5	6	7	8	9	10	11	12
Illuminance	**MLP**	75.09	59.13	46.89	37.21	29.98	15.78	12.97	11.05	10.46	9.35	9.92	9.19
	SMOReg	64.75	51.19	41.74	31.26	28.33	27.81	27.71	27.63	27.7	27.72	27.7	27.7
	RF	64.8	50.32	39.27	29.6	26.71	11.9	12.26	11.31	13.05	14.88	16.92	18.3

Table 6:7 Residential Tower impact of Sensitivity Analysis on the RAE of Illuminance.

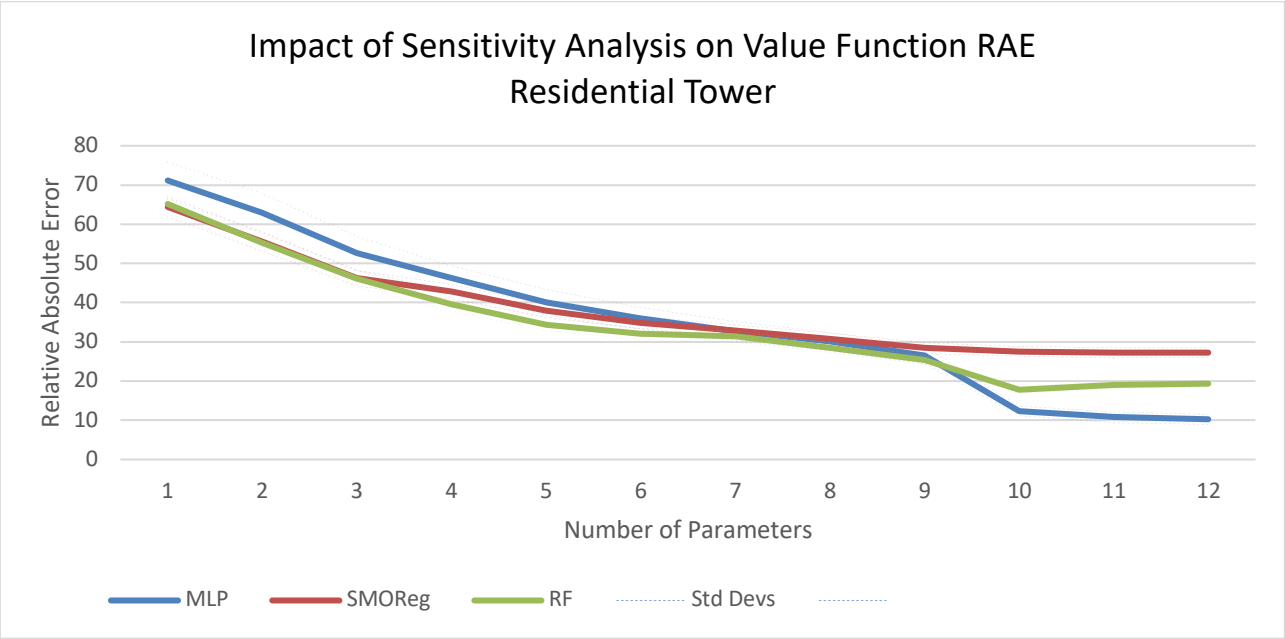


Figure 6:17 Residential Tower impact of Sensitivity Analysis on the RAE of the Value Function

		1	2	3	4	5	6	7	8	9	10	11	12
Value Function	**MLP**	71.15	62.84	52.56	46.25	40	35.94	32.65	30.14	26.54	12.33	10.78	10.23
	SMOReg	64.4	55.56	46.26	42.8	37.95	34.86	32.89	30.67	28.43	27.52	27.24	27.24
	RF	65.16	55.31	46.03	39.58	34.39	32.03	31.34	28.51	25.33	17.76	19.02	19.27

. Table 6:8 Residential Tower impact of Sensitivity Analysis on the RAE of the Value Function.

6.4.3 Feature Selection

Feature Selection is different from sensitivity analysis in that feature selection uses a combinatorial approach to analyze the performance of subsets of the parameter space. It reports the subset with the best performance. However, sensitivity analysis uses methods like correlation analysis to rank parameters in order of their sensitivity.

Table 6:9 establishes the best features for each algorithm and for each objective. The best performing algorithm under feature selection was MLP for the Energy objective (RAE 4.56%). The poorest performing algorithm was SMOReg for the Illuminance objective (RAE 27.7%). On average, for all objectives, MLP performed 5.38% better than RF and 15.97% better than SMOReg. RF performed 10.58% better than SMOReg.

MLP had an insignificant improvement in performance compared to the default baseline for both Energy (0.86%) and Illuminance (0.65%). RF had a more substantial improvement for Energy (2.52%) and Illuminance (6.87%). SMOReg had no substantial change for Energy (0%), Illuminance (0%) and Value Function (0%).

Overall MLP was effective for Energy, Illuminance and the Value Function under feature selection. RF was also effective for Energy but only close to effective for Illuminance and the Value Function. SMOReg was somewhat effective for Energy.

We conclude that Feature Selection has a small impact on machine learning algorithm performance.

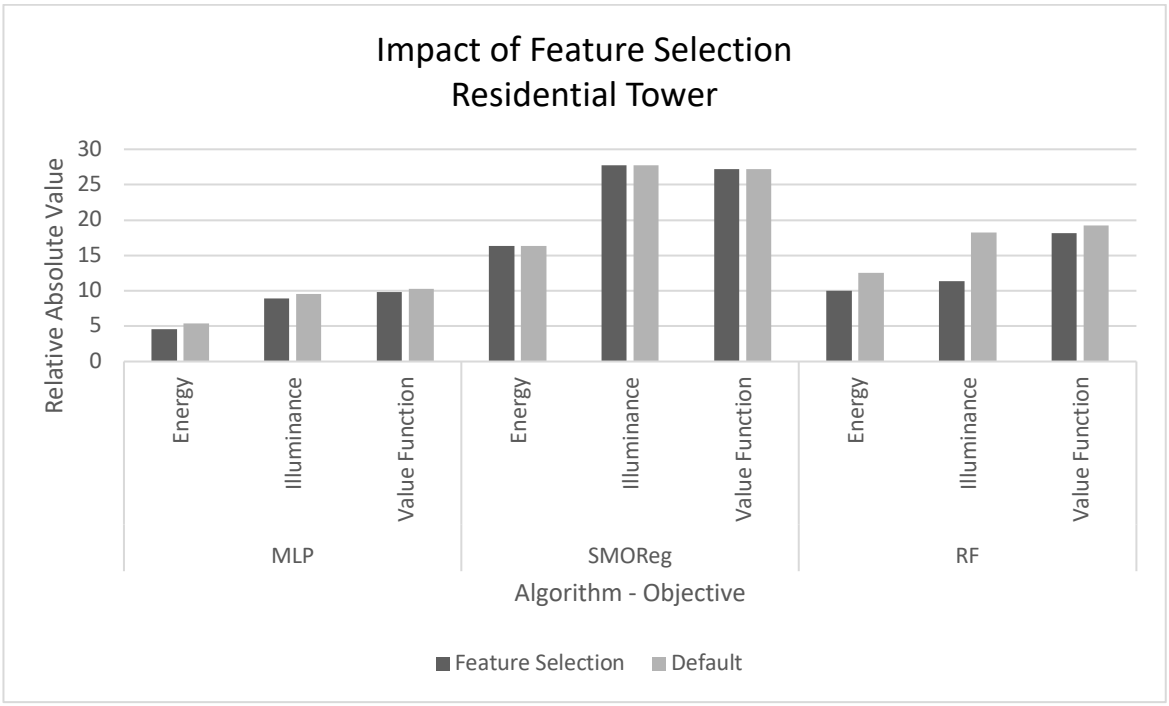


Figure 6:18 Residential Tower impact of Feature Selection.

ALGORITHM	OBJECTIVE	BEST FEATURES	RAE
MLP	Energy	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Shading_N, Shading_W, Shading_S, Shading_E, Window_U_Value, SHGC	4.56
	Illuminance	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Shading_N, Shading_W, Shading_S, Shading_E, Window_U_Value, SHGC	8.93
	Value Function	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Chi_Value, Shading_N, Shading_W, Shading_S, Shading_E, Window_U_Value, SHGC	9.81
SMOReg	Energy	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Shading_W, Shading_S, Window_U_Value, SHGC	16.3
	Illuminance	WWR_N, WWR_W, WWR_S, WWR_E, Shading_N, Shading_S	27.7
	Value Function	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Shading_N, Shading_W, Shading_S, Shading_E, Window_U_Value, SHGC	27.2
RF	Energy	WWR_N, WWR_S, WWR_E, Wall_U_Value, Shading_N, Shading_W, Shading_S, Window_U_Value, SHGC	9.98
	Illuminance	WWR_N, WWR_W, WWR_S, WWR_E, Chi_Value, Shading_N, Shading_W, Shading_S	11.36
	Value Function	WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Chi_Value, Shading_N, Shading_W, Shading_S, Window_U_Value, SHGC	18.11

Table 6:9 Residential Tower impact of Feature Selection.

6.4.4 Hyperparameter Tuning

In this section we seek to understand the impact of each of the key hyperparameters of each of our algorithms. For the MLP algorithm the most impactful hyperparameter for Energy and the Value Function was hiddenLayers. Two hidden layers with 30 neurons each gave the best performance for Energy while two hidden layers with 20 neurons each gave the best performance for the Value Function. The most influential hyperparameter for Illuminance was the learningRate (value = 0.1).

Hyperparameter Tuning for MLP improved performance for the Energy objective by 2.92%, for the Illuminance objective by 1.82% and for the Value Function objective by 3.24%. On average Hyperparameter Tuning for MLP improved performance by 2.66%.

For the SMOReg algorithm the most impactful hyperparameter for Energy, Illuminance and the Value Function was kernel. In all cases the normalizedPolykernel gave the best performance. Performance for the Energy objective improved by 4.69%, for the Illuminance objective by 16.46% and for the Value Function objective by 12.93%. On average Hyperparameter Tuning for SMOReg improved performance by 11.36%.

For the RF algorithm the most impactful hyperparameter for Energy, Illuminance and the Value Function was numFeatures. For Energy and Illuminance, the best value of numFeatures was 8, while for Value Function it was 6. Performance for the Energy objective improved by 2.05%, for the Illuminance objective by 5.25% while the Value Function got worse by 1.24%. On average Hyperparameter Tuning for RF improved performance by 2.02%.

Overall Hyperparameter Tuning was effective for MLP for all objectives. It was close to effective for SMOReg for all objectives. It was also close to effective for RF for Energy and Illuminance.

MLP	hiddenLayers					learningRate			
	(10,10)	*(20,20)*	*(30,30)*	*(40,40)*	*(50,50)*	*0.001*	*0.01*	*0.1*	*0.2*
Energy	3.6	2.82	2.5	2.59	2.69	13.38	6.7	4.13	4.38
Illuminance	8.71	8.41	8.76	8.89	8.9	28.87	9.89	7.76	7.91
Value Function	8.21	6.99	7.61	7.33	7.55	26.86	10.59	8.35	8.86

	momentum				Best Value	Hyperparameter
	0	*0.4*	*0.6*	*0.8*		
Energy	4.94	5.91	7.98	18.52	2.5	hiddenLayers
Illuminance	8.71	10.34	12.34	27.78	7.76	learningRate
Value Function	9.63	11.66	13.27	31.94	6.99	hiddenLayer

Table 6:10 Residential Tower impact of Hyperparameter Tuning on MLP.

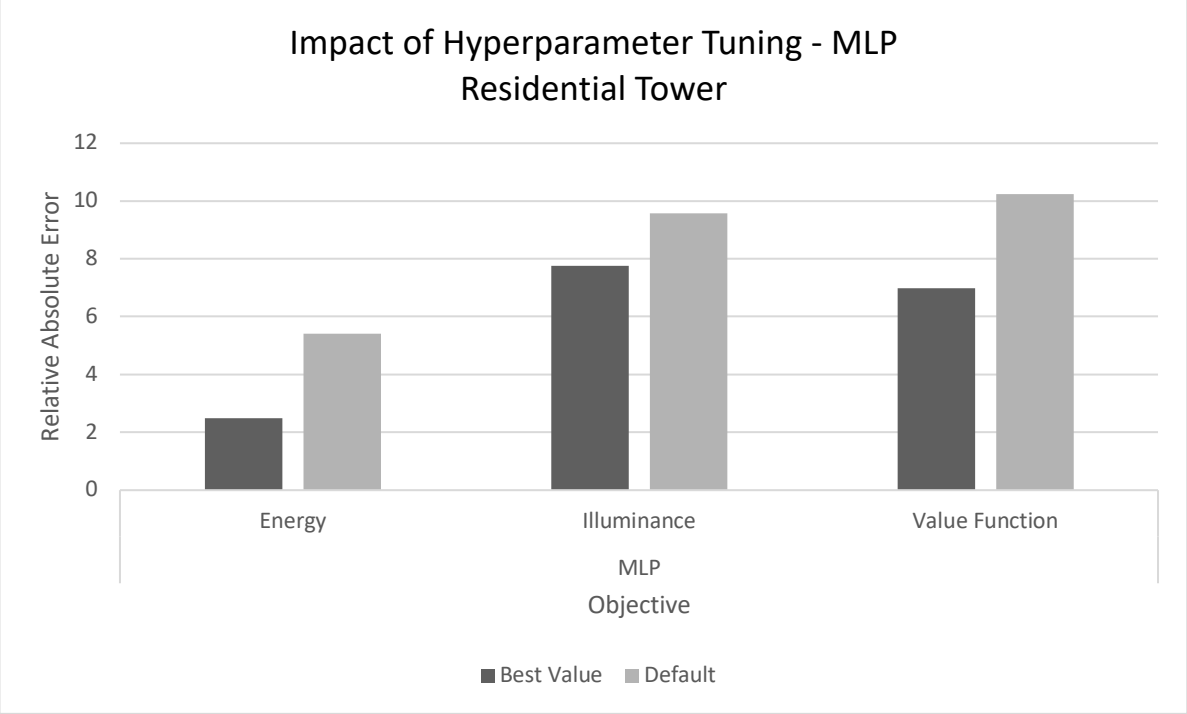


Figure 6:19 Residential Tower impact of Hyperparameter Tuning on MLP.

		Best Value	Default
MLP	**Energy**	2.5	5.42
	Illuminance	7.76	9.58
	Value Function	6.99	10.23

Table 6:11 Residential Tower comparative impact of Hyperparameter Tuning on MLP.

SMOReg	complexity			kernel			epsilonParameter			Best Value	Hyper-parameter
	0.001	*0.01*	*0.1*	*10*	*rbfKernel*	*normalized-Polykernel*	*0.01*	*0.1*	*1*		
Energy	32.97	16.48	16.38	16.37	11.97	11.69	16.31	19.15	104.08	11.69	kernel
Illuminance	46.09	28.05	27.73	27.71	22.9	11.24	27.71	34.21	106	11.24	kernel
Value Function	42.93	27.6	27.23	27.24	22.85	14.31	27.25	29.3	100.08	14.31	kernel

Table 6:12 Residential Tower impact of Hyperparameter Tuning on SMOReg.

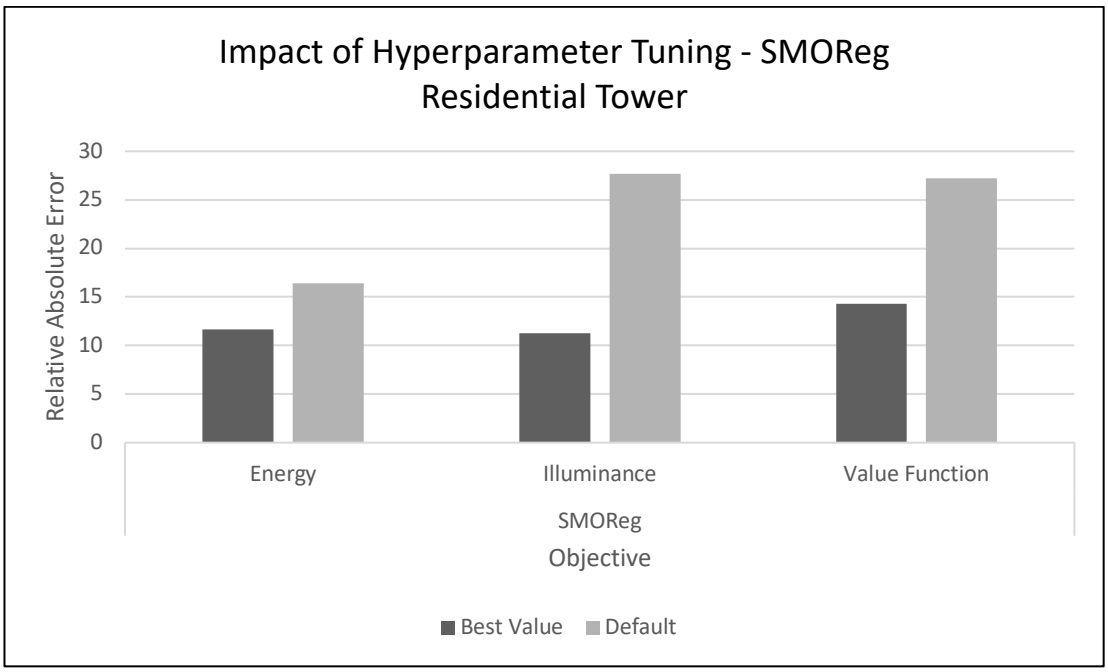


Figure 6:20 Residential Tower impact of Hyperparameter Tuning on SMOReg.

		Best Value	Default
SMOReg	**Energy**	11.69	16.38
	Illuminance	11.24	27.7
	Value Function	14.31	27.24

Table 6:13 Residential Tower comparative impact of Hyperparameter Tuning on SMOReg.

RF	iterations				numFeatures			
	100	*300*	*500*	*700*	*2*	*4*	*6*	*8*
Energy	12.5	12.2	12.12	12.09	23.02	12.5	10.52	10.45
Illuminance	18.23	17.81	17.77	17.75	31.84	18.23	14.02	12.98
Value Function	20.92	20.71	20.67	20.67	24.32	20.92	20.51	20.72
	maxDepth				**Best Value**	**Hyperparameter**		
	50	*100*	*150*	*200*				
Energy	12.5	12.5	12.5	12.5	10.45	numFeatures		
Illuminance	18.23	18.23	18.23	18.23	12.98	numFeatures		
Value Function	20.92	20.92	20.92	20.92	20.51	numFeatures		

Table 6:14 Residential Tower impact of Hyperparameter Tuning on RF.

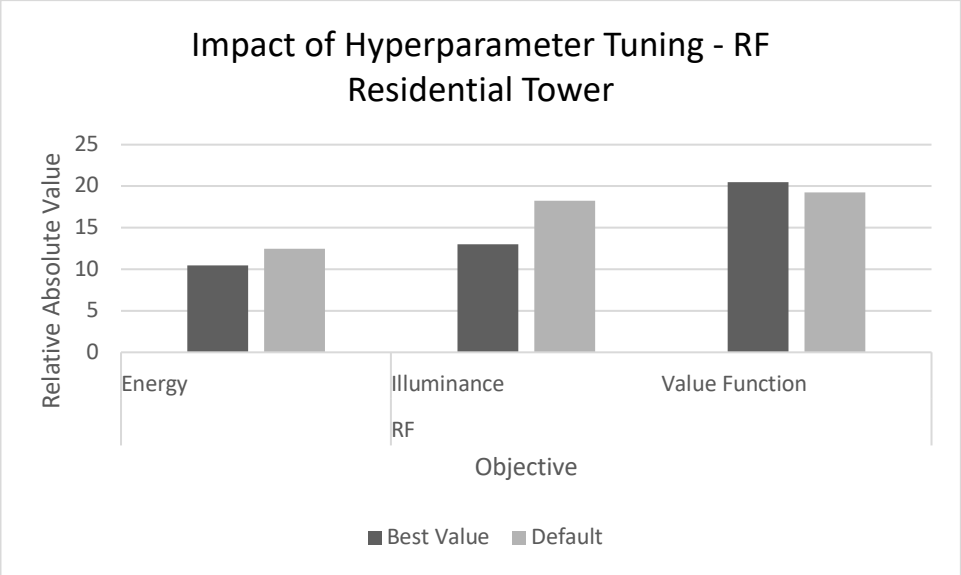


Figure 6:21 Residential Tower impact of Hyperparameter Tuning on RF.

		Best Value	Default
RF	**Energy**	10.45	12.5
	Illuminance	12.98	18.23
	Value Function	20.51	19.27

Table 6:15 Residential Tower impact of Hyperparameter Tuning on RF.

6.5 Discussion

In the reference projects parametric analysis has been performed using the DSC methodology. Machine learning has been used as a design space reduction strategy. The initial design space of the Residential Tower was 14,400 instances. This was reduced to 1,000 instances using DOE sampling. A sample size of 1,000 instances can be simulated in about 33 hours using generally accessible equipment without substantial investments in hardware.

The reduced sample was used to perform tests on machine learning algorithms. The goal of the tests was to establish if machine learning algorithms can perform effectively at design space reduction of multi-discipline parametric analysis. By effective performance is meant acceptable accuracy at acceptable speed. Acceptable

accuracy is considered to be 90 – 95% accuracy (5 – 10% RAE). Acceptable speed is considered to be 33 hours of simulation.

The machine learning algorithms are Weka implementations of ANN, SVM and DT namely MLP, SMOReg and RF. The multi-discipline objectives were Energy and Illuminance (daylighting) and these were combined into a single Value Function for comparative purposes. The tests investigated the impacts of sample size, sensitivity analysis, feature selection and hyperparameter tuning on algorithm performance.

The best performing algorithm for all objectives was MLP. MLP improved to RAE 2.5% for the Energy objective, RAE 7.76% for Illuminance and RAE 6.99% for the Value Function. At its best MLP performed 9.2% better than SMOReg and 7.48% better than RF for Energy; it performed 3.44% better than SMOReg and 3.60% better than RF for Illuminance, and it performed 7.31% better than SMOReg and 11.12% better than RF for the Value Function.

The most important impact factor for MLP was hyperparameter tuning which improved by 2.66% compared to the default. The most important factor for SMOReg was also hyperparameter tuning which improved by 14.15% compared to the default. The most important impact factor for RF was feature selection which improved by 3.52% compared to the default. On average the most important factor was hyperparameter tuning which was 6.27% better than the default.

Overall, MLP was effective for the Energy objective and the Illuminance objective when the algorithm was improved by feature selection and hyperparameter tuning. The algorithm was effective for the Value Function objective when it was improved by feature selection and hyperparameter tuning.

SMOReg was close to effective for the Energy objective when improved by hyperparameter tuning and somewhat effective when improved by feature selection. It was close to effective for the Illuminance objective when improved by hyperparameter tuning. The Value Function was also close to effective when improved by hyperparameter tuning.

RF was close to effective for the Energy objective when improved by sensitivity analysis and hyperparameter tuning. It was somewhat effective under feature selection. Illuminance was close to effective when improved by hyperparameter tuning and somewhat effective when improved by sensitivity analysis. The Value Function was somewhat effective when improved by sensitivity analysis.

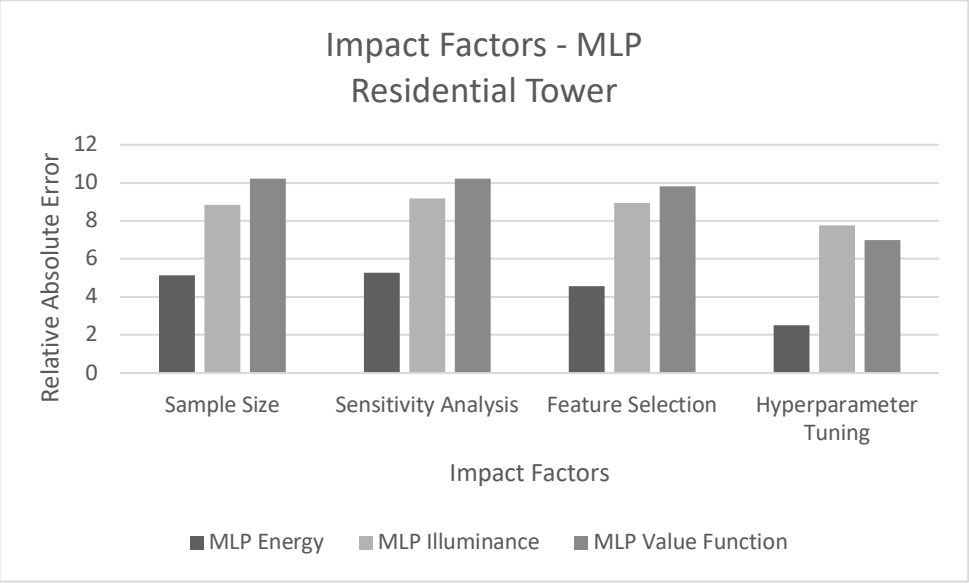


Figure 6:22 Residential Tower effect of impact factors on MLP.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
MLP	**Energy**	5.14	5.27	4.56	2.5
	Illuminance	8.83	9.35	8.93	7.76
	Value Function	10.23	10.78	9.81	6.99

Table 6:16 Residential Tower effect of impact factors on MLP.

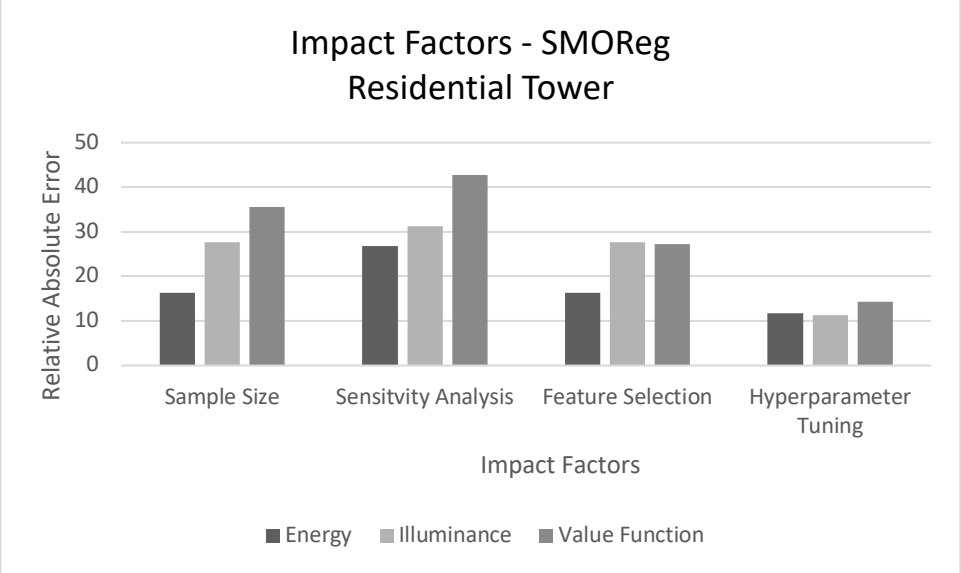


Figure 6:23 Residential Tower effect of impact factors on SMOReg.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
SMOReg	**Energy**	16.38	16.38	16.3	11.7
	Illuminance	27.27	27.63	27.7	11.2
	Value Function	27.23	27.24	27.2	14.3

Table 6:17 Residential Tower effect of impact factors on SMOReg.

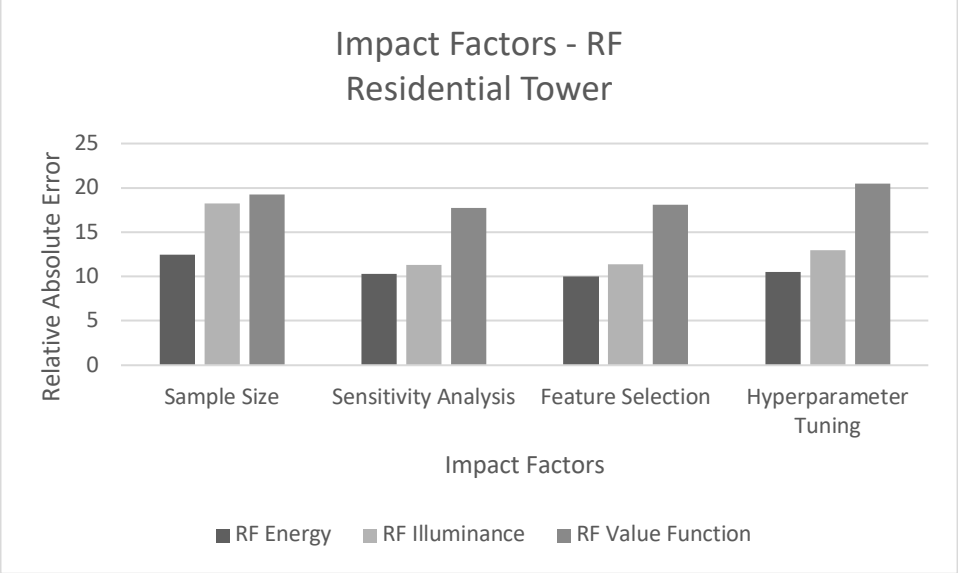


Figure 6:24 Residential Tower effect of impact factors on RF.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
RF	**Energy**	12.5	10.31	9.98	10.5
	Illuminance	18.23	11.31	11.36	13
	Value Function	19.27	17.76	18.11	20.5

Table 6:18 Residential Tower effect of impact factors on RF.

Chapter 7 Mid-Rise Towers Experiment

This chapter performs machine learning experiments on data from the Mid-Rise Towers reference project. We first provide a project description based on DSC and then make use of the general experimental method discussed in Chapter Five. Given the design space complexity of the project, due primarily to different geometry conditions, we investigate the impact of this design space complexity on the performance of machine learning algorithms. We also investigate the influence of impact factors (sample size, sensitivity analysis, feature selection, hyperparameter tuning) on the accuracy of the algorithms. We found that MLP was the best performing algorithm and hyperparameter tuning was the most influential factor for the MLP and SMOReg algorithms.

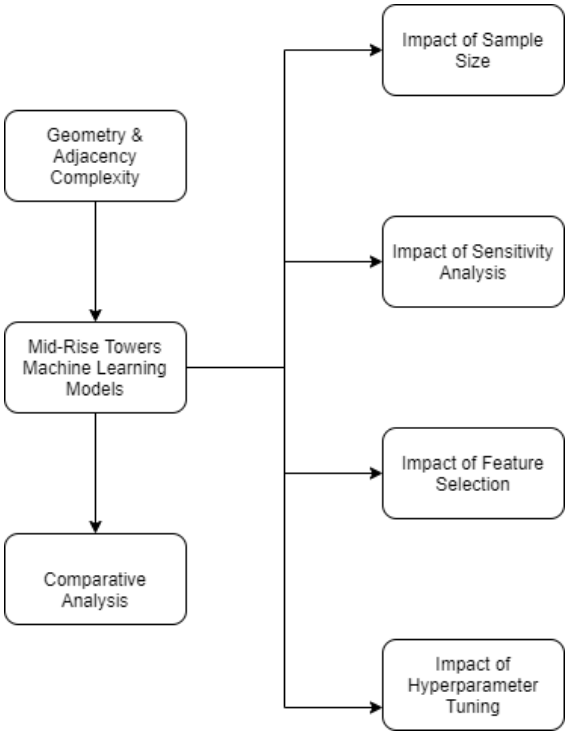


Figure 7:1 Diagram Summary of Chapter Seven.

7.1 Introduction

Mid-Rise Towers (Figure 7:2) consists of three towers from 14 to 18 levels in height standing on two podiums. They are located in a growing, transit-oriented, affordable housing community in Vancouver. The project consisted of two separate buildings on two adjacent sites with a total site area of 63,000 sqft. Two of the towers sit on a connected podium while the third sat on the podium of an adjacent site.

The design rationale was based on community development planning principles. These included achieving a green, environmentally sustainable urban pattern; supporting a range of affordable housing options to meet the diverse needs of the community; fostering a robust resilient economy; enhancing culture, heritage and creativity within the city; providing and supporting a range of sustainable transport options; protecting and enhancing public open spaces, parks and green linkages, and fostering a resilient, safe and healthy community.

In order to achieve the community planning principles, and the developer client's objectives, the designers proposed a range of accommodations including retail units, market housing units, co-op housing units, town houses, underground parking, rooftop and outdoors amenity spaces and an upper level courtyard. The physical form of the design was influenced by topography, creating a stepped ground floor; the need for

solidity and privacy on long street facing facades; large balconies and living spaces oriented towards streetscapes, and acoustic and visual privacy on the transit facing façades.



Figure 7:2 (Perkins&Will, 2019) Mid-Rise Towers project.

7.2 DSC Process

The designers engaged an energy modeling consultant to perform compliance modeling for development permit submissions. The purpose of the DSC process was to validate and build on the findings of the consultant. Given the complexity of the project the design team was interested in more detailed design guidance on different aspects of the design than was available through the consultant's report. Further, the consultant was exclusively an energy consultant. The designers wished to understand the impact of daylighting on the design and achieve a design balance with energy performance.

7.2.1 Objective Definition

The DSC team addressed three questions. First, what is the best performance strategy for each objective, what are the best energy and daylight performance strategies, and what is the best strategy for balancing energy and daylighting? Second, are there significant differences in optimum WWR on the different facades? While the design team understood that there would be differences in building performance and response on different orientations they did not have as much latitude to vary many of the other input parameters. Third, what is the optimized performance for each objective at each of the slices (or design conditions) and what are the corresponding optimized input parameters?

The designers sought detailed design guidance on different aspects of their design in order to meet the City of Vancouver sustainability requirements. The required energy performance limits were TEUI of 120 kWh/m^2/year, TEDI of 30 kWh/m^2/year and GHGI of 6 kgCO_2/m^2. To achieve these minimum performance targets the team proposed air source heat pumps with condensing backup boilers; hydronic fan coil and electric base board heating; in-suite heat recovery ventilation; triple-glazed windows with 0.25 U-Value and 0.35 SHGC; WWR of 35% on the North and South facades and 65% on the East and West facades; an effective R-Value of R-7 for exterior walls and R-40 for the roof. These assumptions were the inputs of the DSC parametric analysis process with parameter ranges centered on these assumed values.

7.2.2 Alternative Generation

Due to the complexity of the project the energy modeler proposed ten slices to capture all the different adjacency conditions of the design. These were the roof, which had an exposed upper surface; the typical floor, which was sandwiched between two typical floors and the podium adjacent floor, which shared a surface with the podium itself (Figure 7:3). In practice, due to different program requirements at the podium level (such as retail and commercial spaces requiring substantial mechanical systems and equipment) the DSC process focused only on the eight non-podium slices.

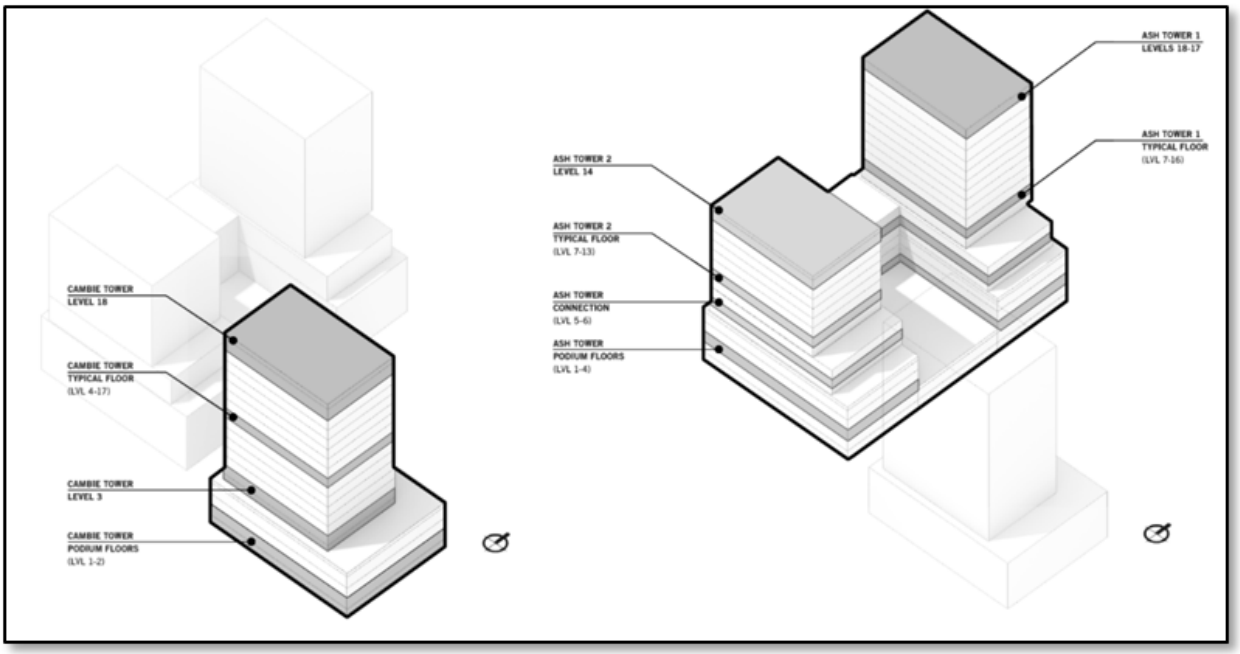


Figure 7:3 (Perkins&Will, 2019) Mid-Rise Towers project slices.

The design space of the analysis is shown in Table 7:3 and for each slice there was a design space size of 1296 instances for a total of 10,368 simulations. The zoning was related to the project slicing and suite layouts (Figure 7:4). In addition to zoning other inputs into the parametric analysis process included glazing information such as SHGC values; envelope information such as Wall U-Value; lighting information such as lighting controls; equipment information such as equipment loads; design conditions such as thermostat set-back temperatures, and air system information such as heat recovery system. Zone inputs included program type, lighting power density, occupancy, metabolic rate, ventilation and equipment.

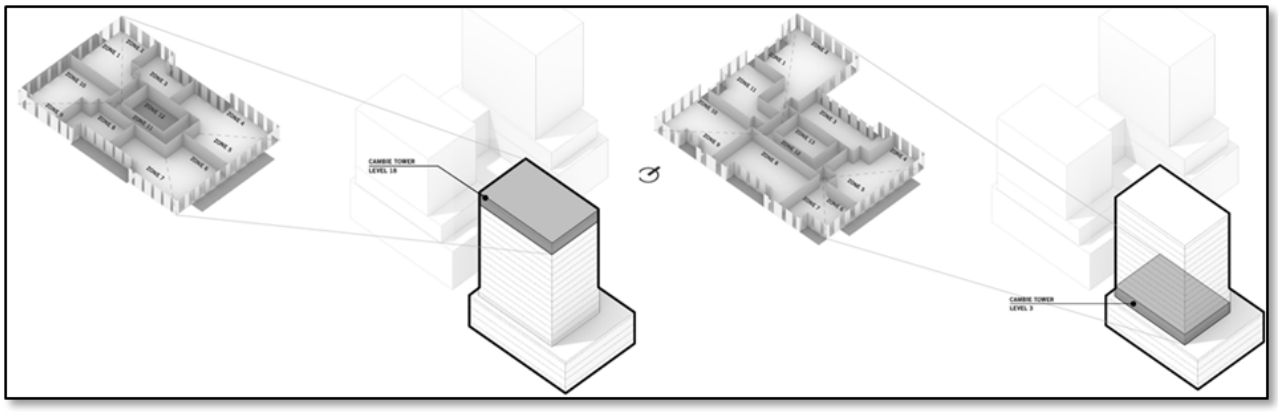


Figure 7:4 (Perkins&Will, 2019) Mid-Rise Towers zoning approach

The 10,368 simulations would take 14.4 days to simulate which was not practical. The DSC team therefore used a sampling approach to reduce the design space. Using a DOE method, they sampled 200 instances from the proposed 1296 instance design space. They simulated this reduced design space for each of the eight slices (1600 instances) and then aggregated the simulations into one set of results. Simulation took about 2 days.

7.2.3 Impact Analysis

Impact analysis was presented in a comprehensive report that detailed optimal parameter values for the different slices and adjacency conditions of the design. The report included a whole building summary section, a performance matrix section for each slice of the building and a detailed breakdown section for each slice.

The whole building summary showed the optimized Energy and Illuminance for each slice of the building (Figure 7:5). The performance matrix for each slice of the building showed the optimized Site Energy and its implication for Illuminance; the optimized Illuminance and its implications for Energy, and the Balanced case (Figure 7:6). The detailed breakdown for each slice showed the best, the average and the worst Site Energy and Illuminance and their corresponding Illuminance and Site Energy (Figure 7:7). It also showed a Site Energy breakdown of the key sources of Energy consumption at the slice.

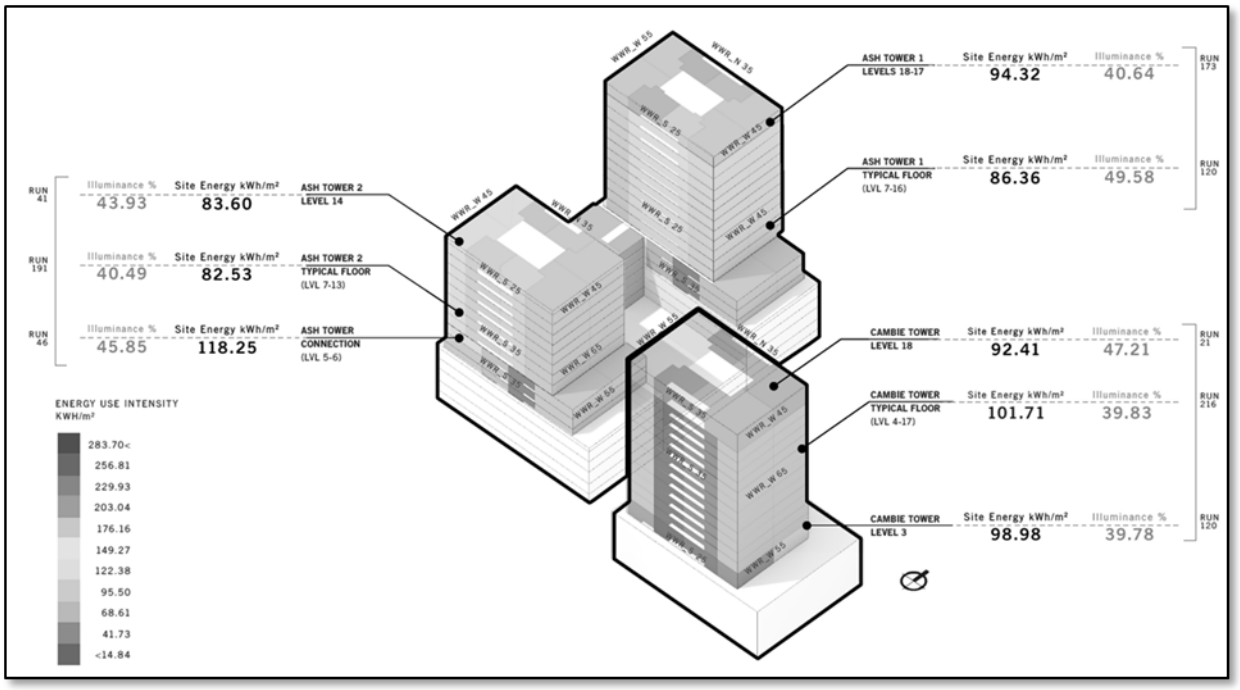


Figure 7:5 (Perkins&Will, 2019) Mid-Rise Towers whole building summary report.

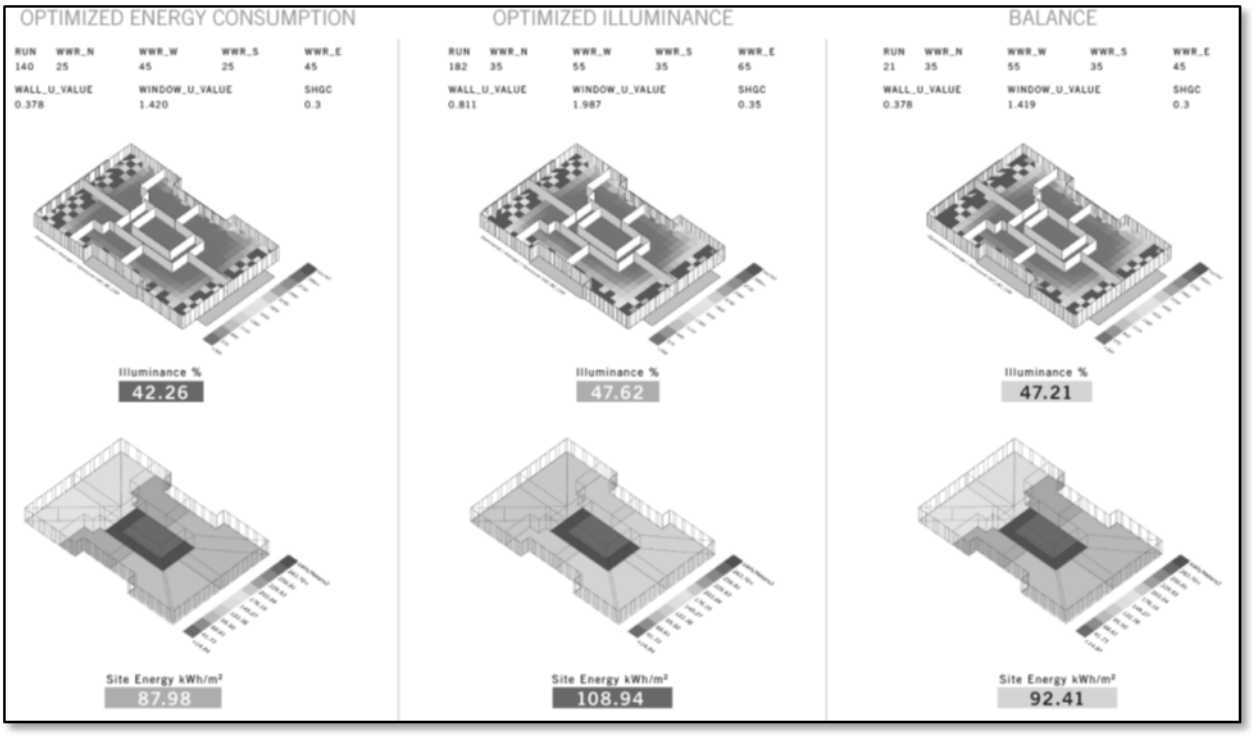


Figure 7:6 (Perkins&Will, 2019) Mid-Rise Towers performance matrix report.

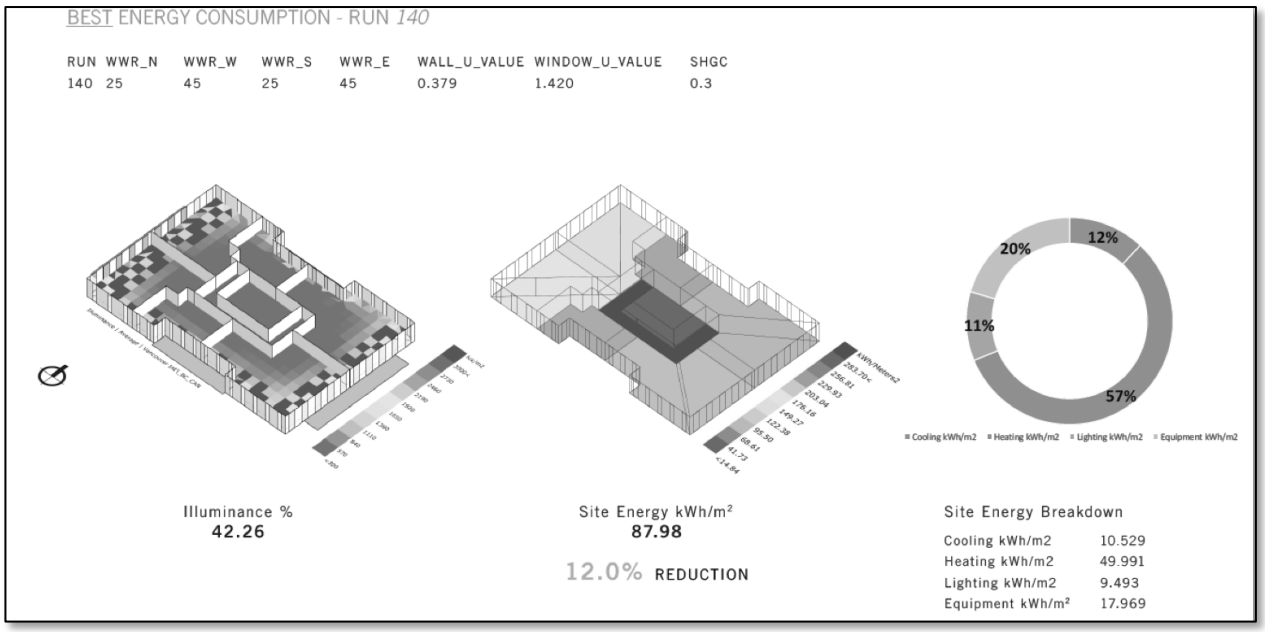


Figure 7:7 (Perkins&Will, 2019) Mid-Rise Towers detailed breakdown report.

7.2.4 Value Assessment

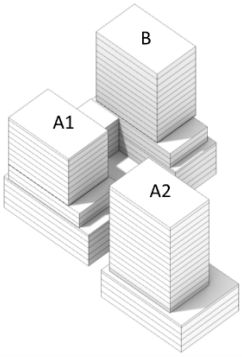


Figure 7:8 Mid-Rise Towers identification key.

Discipline	Value	Level	Comparison	WWR	Win U-Value	Wall U-Value	SHGC
Energy	77.63 (kWh/m^2)	A2 - Typical	37.19% (Illum.)	25% - N,S 45% - E,W	1.42	0.38	0.3
Daylight	56.12%	B-Podium Adj.	113.70 (kWh/m^2)	25% - N,S 55%-E, 65%-W	1.7	1.135	0.3
Balance	82.53 (kWh/m^2)	A2 - Typical	40.49%	25%-N, 35%-S 65%-E, 45%-W	1.7	0.38	0.3

Table 7:1 Mid-Rise Tower summary of optimal values.

The optimum findings from the Mid-Rise Towers are summarized in Table 7:1. The best energy performance occurs on the typical floor of tower A2 (77.63 kWh/m^2/year), the best daylight illuminance occurs on the podium adjacent level of tower B (56.12%) while the balanced case occurs at the typical floor level of tower A2. The comparative value of the opposing discipline is also shown together with the input parameter values at optimum.

In addition to the comprehensive performance reporting a PCP was made available to the design team for any further interactive interrogation of the design parameters (Figure 7:9). Lastly a sensitivity analysis of heating, cooling and illuminance was carried out to inform the designers which input parameters were most impactful (Figure 7:10).

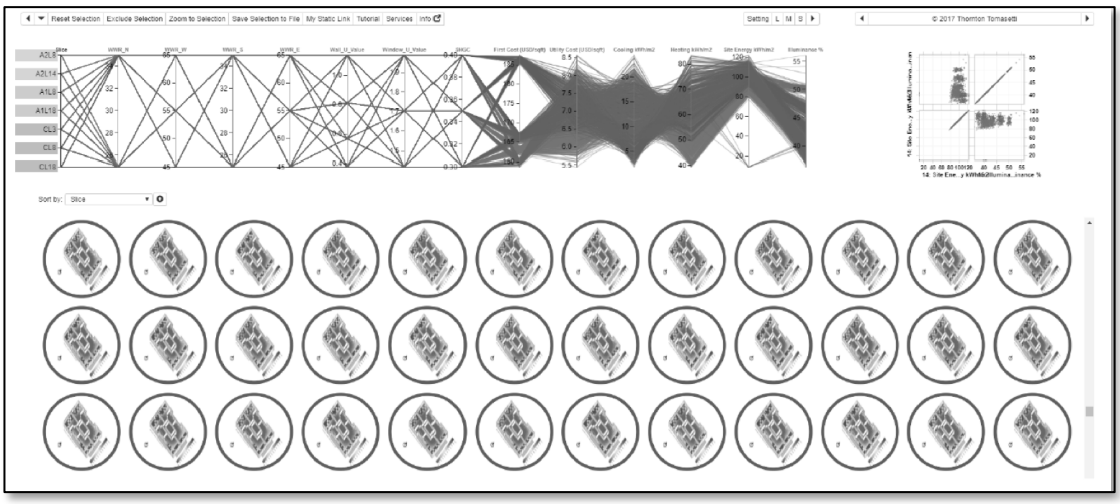


Figure 7:9 (Perkins&Will, 2019) Mid-Rise Towers parallel coordinates plot.

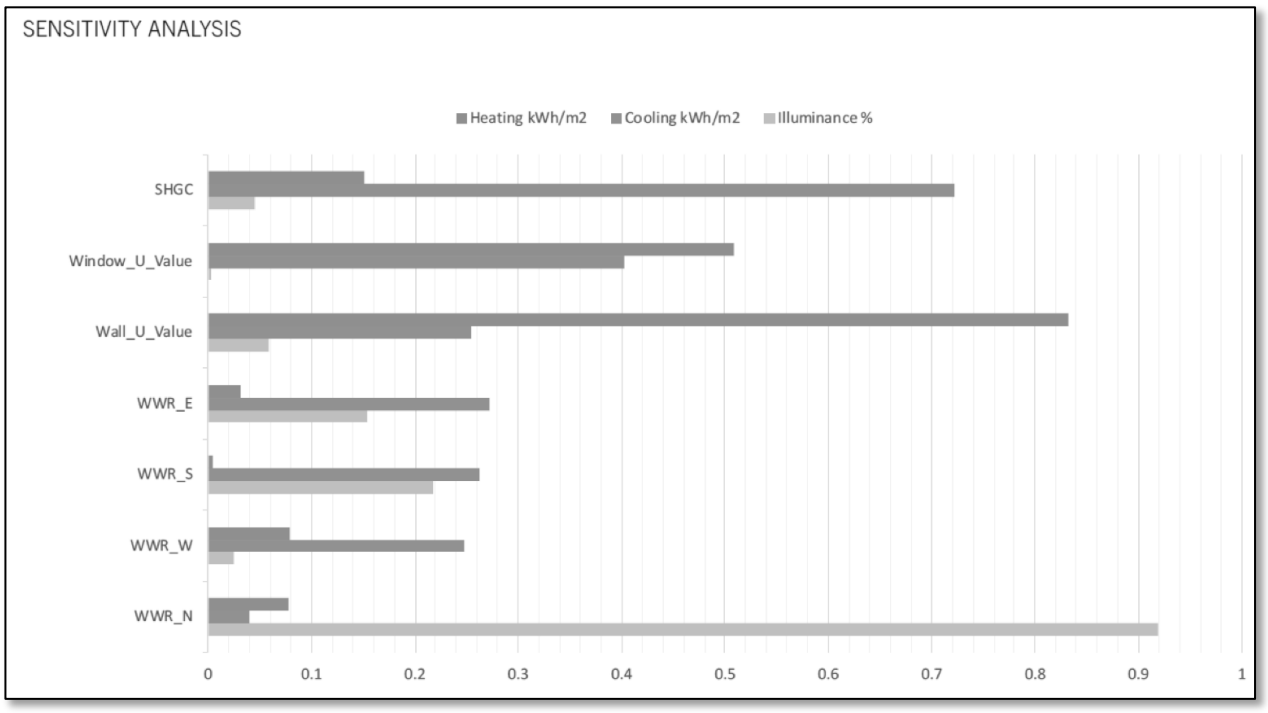


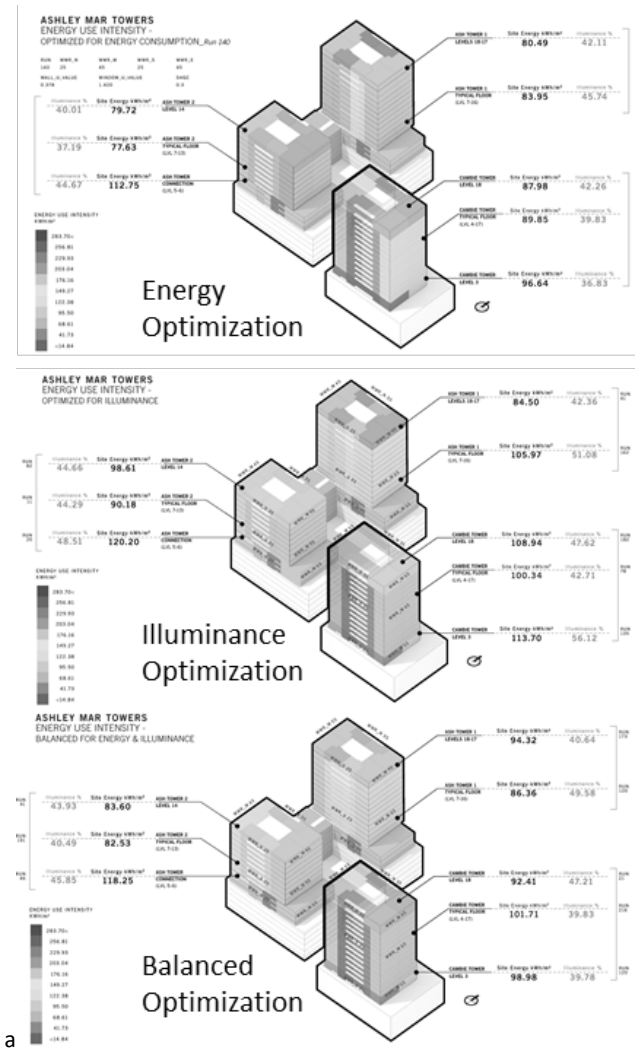
Figure 7:10 (Perkins&Will, 2019) Mid-Rise Towers sensitivity analysis.

7.2.5 Discussion

Parametric Analysis on the Mid-Rise Towers responded to the questions raised by the DSC team. With respect to the best performance strategies a whole building report was developed for Energy, Illuminance and the Balanced case (Figure 7:11). With respect to optimum values for WWR it is clear from the whole building reports that different locations and orientations have different optimum values. For example, based on the Balanced whole building energy report we can see that optimum WWR on the West façade of A1 is 55%, of A2 is 45% and of B is 55% (Figure 7:11). In addition, the East façade of Tower C has 45% WWR at the top, 55% WWR at the typical levels and 45% at the bottom. With respect to the best performing alternatives at each slice (design condition) this can be seen from the performance matrix reports. For example, the best performing alternatives for each objective, as well as the balanced case, for the typical level on Tower B are shown in Figure 7:13. In addition, sensitivity analysis indicates which parameters have the most influence on each of the objectives (Figure 7:10).

However, the sampling method used for the exercise raises some issues related to sampling accuracy. There is a loss of accuracy when optimizing from a small sample compared to the full design space. The sampling method does not report this error. This means we do not know what level of confidence we can have in the results. In addition, the sensitivity analysis is also performed with less data than the full design space which affects accuracy. These accuracy related issues incentivize us to investigate developing a design space reduction model using machine learning to allow us to predict the full design space.

Mid-Rise Towers also exhibits geometric and adjacency complexity. Machine learning models must account for this complexity. We need to investigate whether algorithms can still be effective under these complex design space conditions.



a

Figure 7:11 (Perkins&Will, 2019) Mid-Rise Towers best performing strategies.

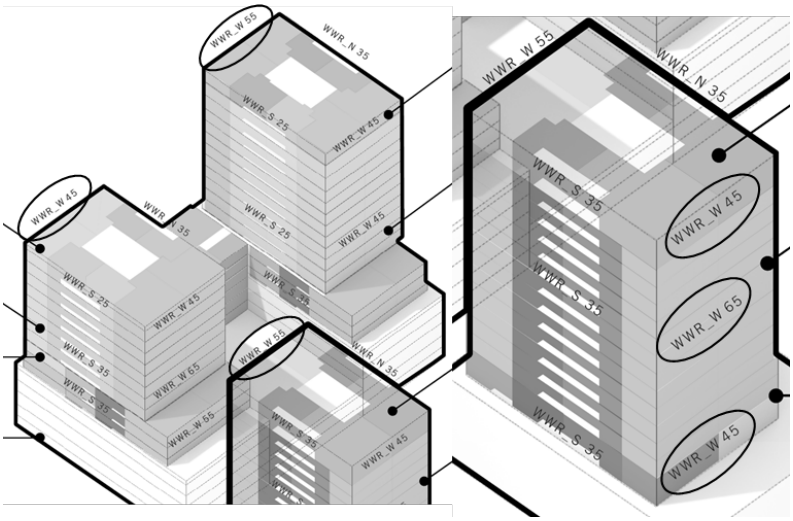


Figure 7:12 (Perkins&Will, 2019) Mid-Rise Towers optimum WWR ratios.

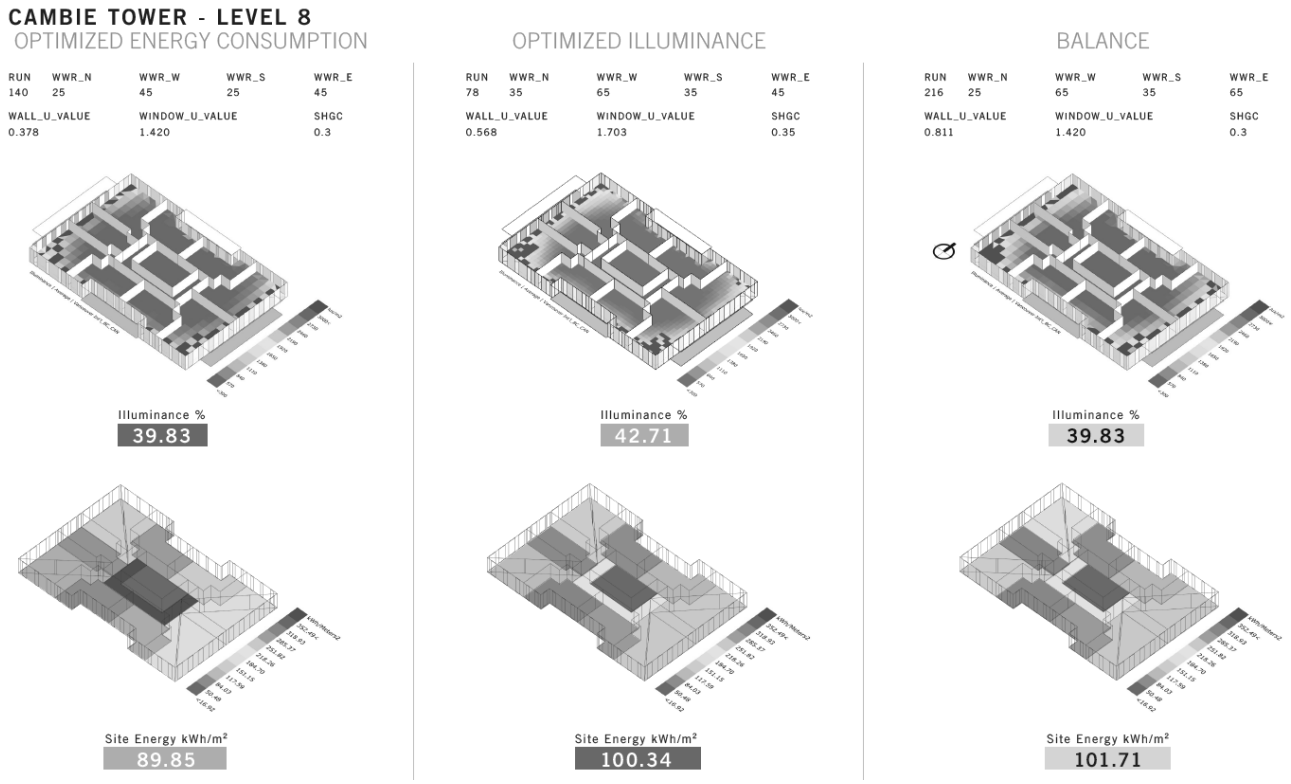


Figure 7:13 (Perkins&Will, 2019) MRT best performing alternatives for Tower B typical levels.

7.3 Method

The general experimental method used for this reference project has been described in Chapter Five. In this section we will describe the method for performing machine learning on multiple slices of the project. We will also present the specific data description and design space definition of the Mid-Rise Towers.

As discussed in Section 7.2.2 due to the complexity of the design ten slices were initially identified for simulation according to the distinct adjacency conditions of the design. However, the podium levels were not included in the DSC process because it was felt that they would need substantial input from mechanical engineering consultants to be accurately analyzed. Therefore, eight slices were simulated with 200 simulations per slice. From the combined sample of about 1600 instances 1000 instances were chosen at random for machine learning.

The machine learning training set consisted of all the simulation input parameters with an additional categorical parameter for the particular slice of the design. This Slice ID parameter had eight possible values. In performing the simulations for the Mid-Rise Towers, the same design space was used for each slice. This would enable better comparison across slices. However, this approach could potentially overfit the machine learning model. In Chapter Eight we will examine the effect of using random samples for each slice compared to using an identical design space for each slice with only the slice identifier changing.

7.3.1 Data Description

The data used for the Mid-Rise Towers machine learning experiments is described in Table 7:2.

Input	Data Type	Units	Description
WWR_N	Numerical - Discrete	%	Window to wall ratio on the North façade
WWR_W	Numerical - Discrete	%	Window to wall ratio on the West façade
WWR_S	Numerical - Discrete	%	Window to wall ratio on the South façade
WWR_E	Numerical - Discrete	%	Window to wall ratio on the East façade
Wall_U_Value	Numerical - Discrete	$W/m^2 K$	Wall thermal transmittance value
Window U_Value	Numerical - Discrete	$W/m^2 K$	Window thermal transmittance value
Chi-Value	Numerical - Discrete	W/K	Heat flow coefficient representing the added heat flow associated with thermal bridging.
SHGC	Numerical - Discrete	-	Solar Heat Gain Coefficient
Slice ID			Identification of the floor to analyze.
OUTPUT	**Data Type**	**Units**	**Description**
Cooling	Numerical - Continuous	kWh/m^2	Cooling load
Heating	Numerical - Continuous	kWh/m^2	Heating load
Site Energy	Numerical - Continuous	kWh/m^2	Total Site Energy
Illuminance %	Numerical - Continuous	%	Percentage of usable floor area with 300 - 3000 lux illuminance

Table 7:2 Mid-Rise Tower data description.

7.3.2 Design Space Definition

The design space definition of the Mid-Rise Towers is shown in Table 7:3.

Parameter	Description	Range	Size
WWR North	Window to wall ratio	25,35	2
WWR West	Window to wall ratio	45,55,65	3
WWR East	Window to wall ratio	25,35	2
WWR South	Window to wall ratio	45,55,65	3
Wall_U_Value	Wall U Value	0.379,0.568,0.811,1.136	4
Chi_Value	Point heat transmittance coefficient	0.48	1
Window_U_Value	Window U Value	1.42,1.7,1.99	3
SHGC	Solar Heat Gain Coefficient	0.3,0.35,0.4	3
Slice_ID	Project slice identifier	1-8	8
			10,368

Table 7:3 Mid-Rise Tower design space description.

7.4 Results

7.4.1 Sample Size

The impact of Sample Size on Site Energy for the Mid-Rise Towers was that all algorithms perform better with a large sample (1000 instances) than with a small sample (100 instances) by 10.84% on average. The best performing algorithm was MLP with 1000 instances (RAE 5.08%) while the poorest performing was

RF with 100 instances (RAE 41.44%). Only MLP was effective with an RAE of 9.17% at sample sizes starting at 200 instances.

The effect of Sample Size on Illuminance was that all algorithms performed better with a large sample size by 6.51% on average. The best performing algorithm was MLP with 1000 instances (RAE 17.35%) while the poorest performing algorithm was RF with 100 instances (RAE 26.55%). No algorithm was effective for Illuminance by default.

The effect of Sample Size on the multi-discipline Value Function was that all algorithms performed better with a large sample size by 9.67% on average. The best performing algorithm was MLP with 1000 instances (RAE 14.99%) while the poorest performing algorithm was RF with 100 instances (RAE 38.05%). No algorithm was effective for the Value Function by default.

On average MLP performed 8.72% better than RF and 5.26% better than SMOReg. SMOReg performed better than RF by 3.46%. Although again RF improved performance faster than SMOReg in this reference project. it did not surpass the performance of SMOReg for the Energy objective. On average the Energy objective performed better than the Illuminance objective and the Value Function objective by about 4.2%. Illuminance and the Value Function were evenly matched in performance.

Overall MLP was effective for the Energy objective and close to effective for the Illuminance objective and the Value Function objective. SMOReg was close to effective for the Energy objective but somewhat effective for the Value Function objective. RF was close to effective for the Illuminance objective and somewhat effective for the Energy objective and the Value Function objective.

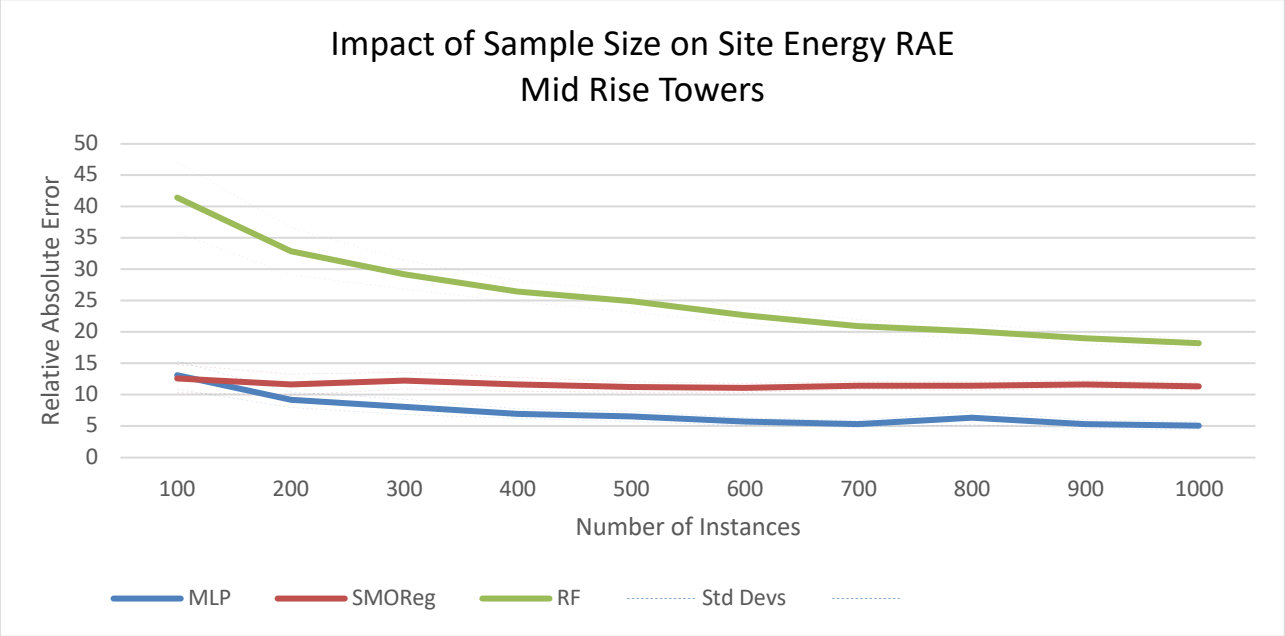


Figure 7:14 Mid-Rise Towers impact of Sample Size on the RAE of Site Energy.

Sample Size	100	200	300	400	500	600	700	800	900	1000
Multi-Layer Perceptron	13.13	9.17	8.08	6.98	6.57	5.72	5.36	6.32	5.31	5.08
SMOReg	12.61	11.61	12.22	11.67	11.26	11.11	11.5	11.45	11.7	11.36
Random Forest	41.44	32.86	29.19	26.43	24.89	22.67	20.91	20.15	18.98	18.22

Table 7:4 Mid-Rise Towers impact of sample size on the RAE of Site Energy.

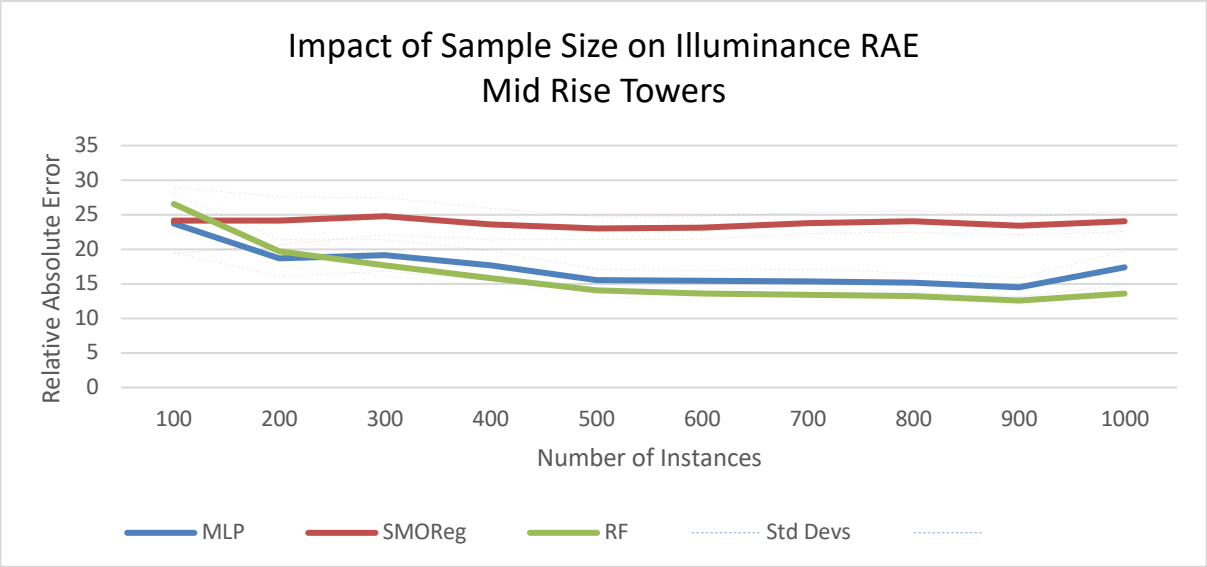


Figure 7:15 Mid-Rise Towers impact of Sample Size on the RAE of Illuminance.

Sample Size	100	200	300	400	500	600	700	800	900	1000
Multi-Layer Perceptron	23.72	18.72	19.13	17.62	15.56	15.45	15.31	15.13	14.49	17.35
SMOReg	24.17	24.11	24.78	23.63	23.01	23.12	23.76	24.04	23.41	24.01
Random Forest	26.55	19.69	17.64	15.78	14.06	13.59	13.41	13.24	12.57	13.56

Table 7:5 Mid-Rise Towers impact of sample size on the RAE of Illuminance.

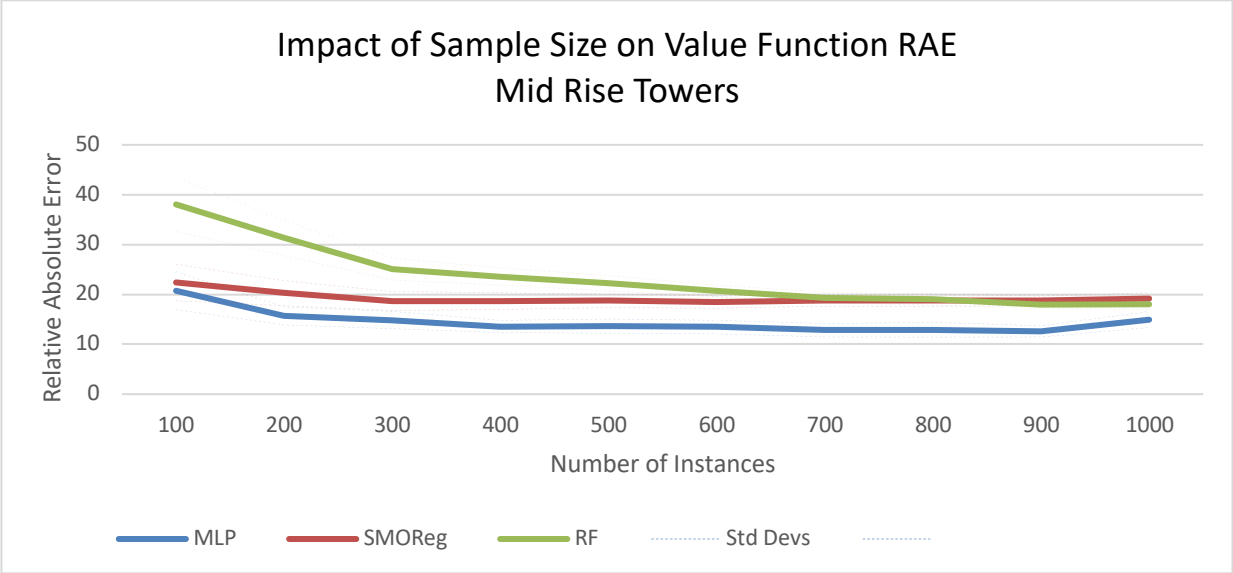


Figure 7:16 Mid-Rise Towers impact of Sample Size on the RAE of the Value Function.

Sample Size	100	200	300	400	500	600	700	800	900	1000
Multi-Layer Perceptron	20.73	15.66	14.82	13.55	13.61	13.47	12.84	12.91	12.59	14.99
SMOReg	22.4	20.3	18.67	18.6	18.73	18.47	18.79	18.84	18.74	19.13
Random Forest	38.05	31.31	25.06	23.56	22.2	20.74	19.31	19.01	17.94	18.05

Table 7:6 Mid-Rise Towers impact of sample size on the RAE of the Value Function.

7.4.2 Sensitivity Analysis

Sensitivity Analysis established that Slice ID was the most sensitive parameter impacting Site Energy. The other sensitive parameter (greater than 10% for Total Effect) was Wall U-Value. Window U-Value was somewhat sensitive (greater than 5% for Total Effect). Average performance was significantly worse with just the sensitive parameters (17.35%) compared to all parameters (11.28%). This indicated that Sensitivity Analysis was not an influential factor for Site Energy.

The best performing algorithm was MLP with 8 parameters (RAE 4.96%) while the worst performing algorithm was again MLP with a single parameter (RAE 77.95%). At best MLP performed better than RF by 12.65% and better than SMOReg by 6.3%. SMOReg performed better than RF by 6.35%. Only MLP was effective starting at 7 parameters. MLP was somewhat effective at 6 parameters. SMOReg was somewhat effective starting at 6 parameters and close to effective starting at 7 parameters. RF was somewhat effective starting at 6 parameters.

For Illuminance Sensitivity Analysis, Slice ID was the most sensitive parameter. WWR North was somewhat sensitive. Average performance was slightly better with just the sensitive parameters (18.00%) compared to all parameters (18.53%). Since the difference is very small this suggests that Sensitivity Analysis did not make much difference to performance for the Illuminance objective.

The best performing algorithm was RF with 8 parameters (RAE 13.1%) while the worst performing algorithm was MLP with a single parameter (RAE 47.27%). At best RF performed better than MLP by 3.55% and better than SMOReg by 10.6%. MLP performed better than SMOReg by 7.05%. None of the algorithms was effective under Sensitivity Analysis for Illuminance. RF was close to effective starting at 7 parameters and somewhat effective starting at 3 parameters. MLP was somewhat effective starting at 4 parameters. SMOreg was not at all effective.

For the Value Function, Slice ID was the most sensitive parameter. Wall U-Value was another sensitive parameter. Average performance was slightly worse with just the sensitive parameters (18.47%) compared to all parameters (17.19%). This indicated that Sensitivity Analysis was not an influential factor.

The best performing algorithm was MLP with 8 parameters (RAE 14.76%) while the worst performing algorithm was MLP with a single parameter (RAE 63.27%). At best MLP performed better than RF by 3.07% and better than SMOReg by 4.22%. RF performed better than SMOReg by 1.15%. None of the algorithms was effective but MLP was close to effective starting at 8 parameters and somewhat effective starting at 7 parameters, while RF was somewhat effective starting at 7 parameters. SMOReg was somewhat effective starting at 7 parameters.

Overall none of the algorithms was effective when considering only sensitive parameters. MLP was somewhat effective for all objectives, SMOReg was somewhat effective for Energy and the Value Function while RF was close to effective for Illuminance and somewhat effective for Energy and the Value Function.

Summary Report

Column	Main Effect	Total Effect
Slice	0.544	0.563
Wall_U_Val	0.28	0.299
Window_U_Val	0.049	0.068
SHGC	0.014	0.024
WWR_W	0.014	0.023
WWR_E	0.008	0.016
WWR_S	0.006	0.01
WWR_N	0.001	0.003

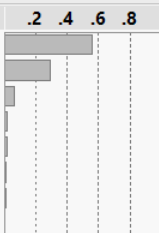


Figure 7:17 Mid-Rise Towers Sensitivity Analysis, Energy.

Summary Report

Column	Main Effect	Total Effect
Slice	0.863	0.879
WWR_N	0.079	0.095
WWR_S	0.013	0.021
WWR_E	0.001	0.003
SHGC	4e-4	0.001
Wall_U_Val	2e-4	3e-4
WWR_W	9e-5	2e-4
Window_U_Val	4e-5	1e-4

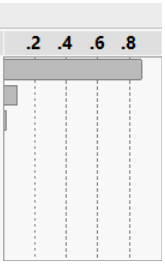


Figure 7:18 Mid-Rise Towers Sensitivity Analysis, Illuminance.

Summary Report

Column	Main Effect	Total Effect
Slice	0.742	0.775
Wall_U_Val	0.123	0.156
Window_U_Val	0.015	0.034
WWR_N	0.011	0.026
SHGC	0.008	0.019
WWR_W	0.005	0.012
WWR_E	0.001	0.004
WWR_S	2e-4	0.001

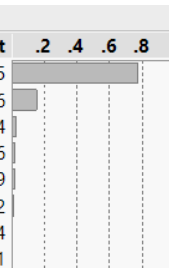


Figure 7:19 Mid Rise Tower Sensitivity Analysis, Value Function.

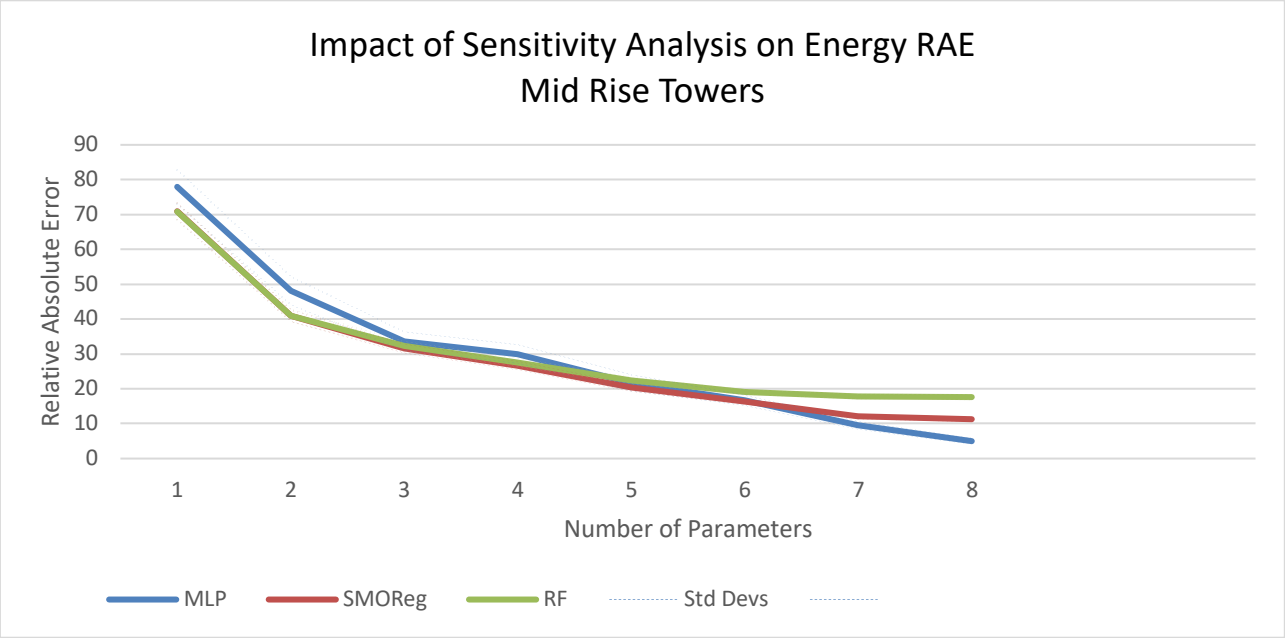


Figure 7:20 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of Energy.

		1	2	3	4	5	6	7	8
Energy	**MLP**	77.95	48.16	33.68	29.96	21.7	16.71	9.52	4.96
	SMOReg	70.95	40.95	31.56	26.7	20.36	16.3	12.02	11.26
	RF	70.85	40.98	32.25	27.59	22.33	19.03	17.86	17.61

Table 7:7 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of Energy.

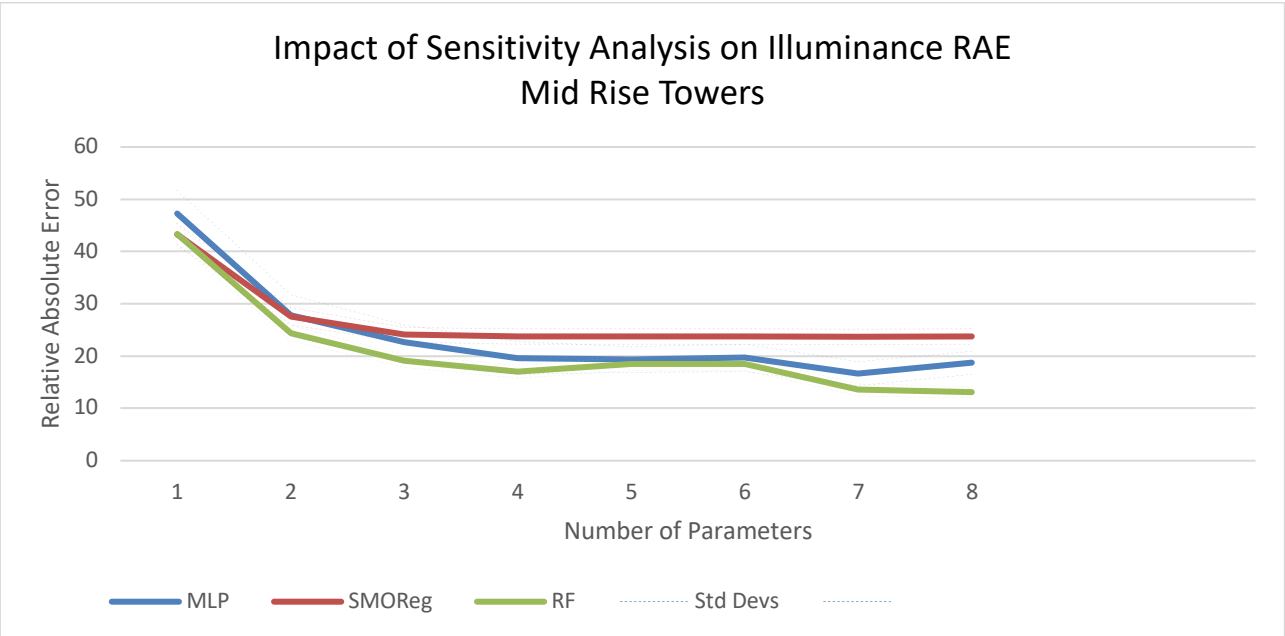


Figure 7:21 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of Illuminance.

		1	2	3	4	5	6	7	8
Illuminance	**MLP**	47.27	27.77	22.66	19.61	19.38	19.68	16.65	18.71
	SMOReg	43.29	27.56	24.07	23.76	23.78	23.77	23.7	23.79
	RF	43.35	24.39	19.06	17.04	18.53	18.54	13.54	13.1

Table 7:8 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of Illuminance.

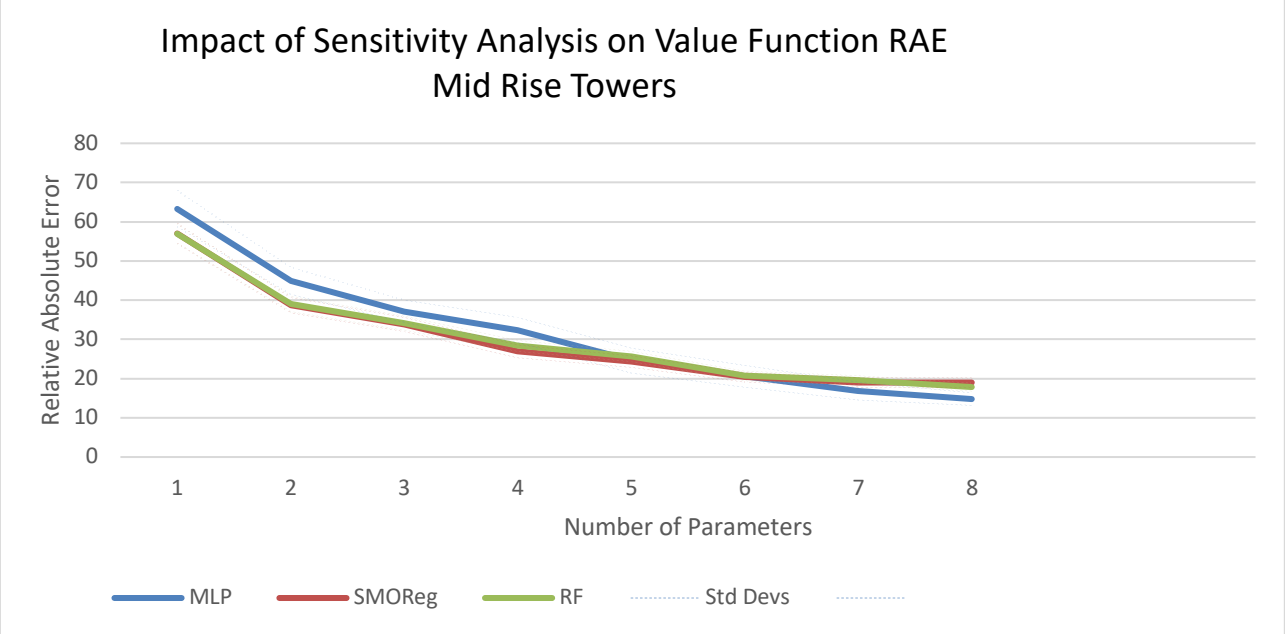


Figure 7:22 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of the Value Function.

		1	2	3	4	5	6	7	8
Value Function	**MLP**	63.27	44.92	37.13	32.39	24.58	20.59	16.82	14.76
	SMOReg	57	38.68	33.76	26.92	24.34	20.38	18.98	18.98
	RF	56.93	39	34.09	28.34	25.61	20.66	19.62	17.83

Table 7:9 Mid-Rise Towers impact of Sensitivity Analysis on the RAE of the Value Function.

7.4.3 Feature Selection

For the Mid-Rise Towers, Table 7:10 established the best features for each algorithm and for each objective. The best performing algorithm after Feature Selection was MLP for the Energy objective (RAE 5.08%). The poorest performing algorithm was SMOReg for the Illuminance objective (RAE 23.72%). On average, for all objectives, MLP performed 3.75% better than RF and 5.55% better than SMOReg. RF performed 1.8% better than SMOReg.

For both Energy and the Value Function all algorithms returned the full parameter space as the best feature selection. In this case there was no difference in performance compared to the default. In the case of Illuminance there were insignificant differences for MLP (0.15%), SMOReg (0.28%) and RF (1.01%).

Overall, MLP was effective for Energy, somewhat effective for Illuminance and close to effective for the Value Function after Feature Selection. SMOReg was close to effective for Energy and somewhat effective for the Value Function. RF was somewhat effective for Energy, close to effective for Illuminance and somewhat effective for the Value Function.

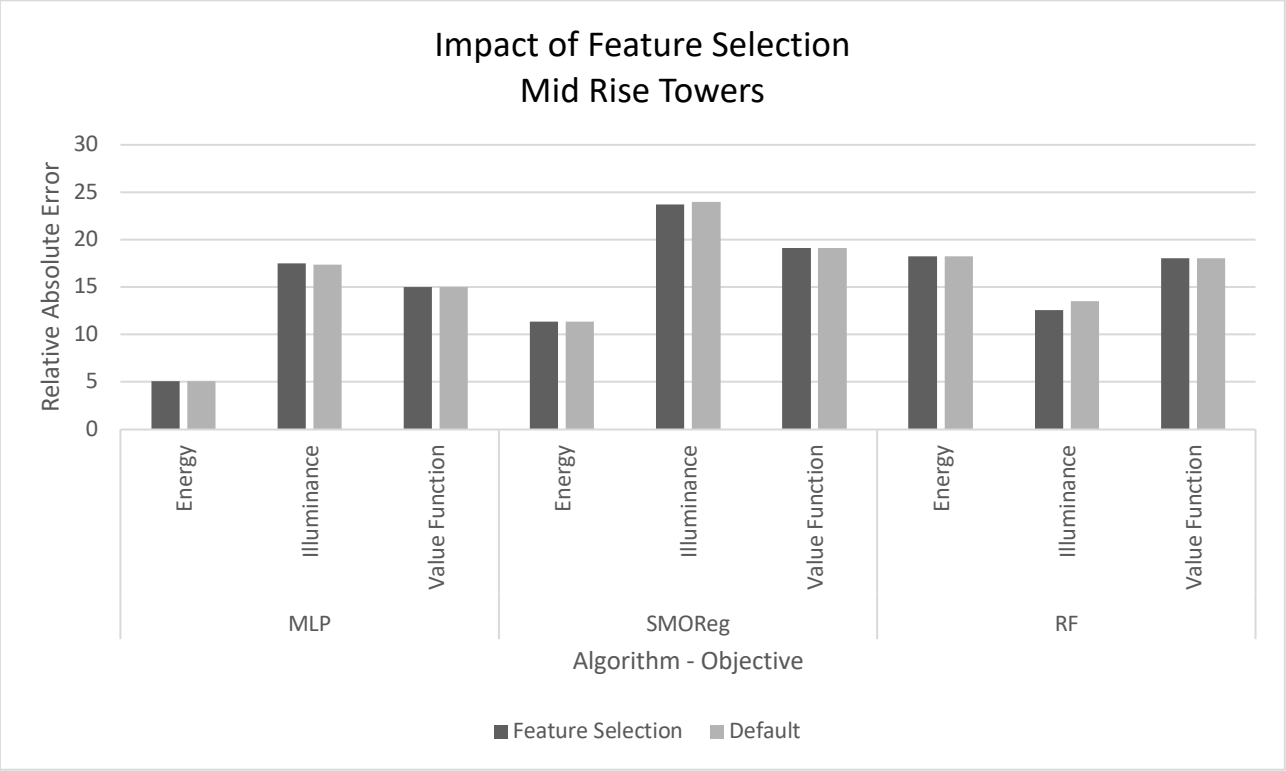


Figure 7:23 Mid Rise Towers impact of Feature Selection on the RAE.

ALGORITHM	OBJECTIVE	BEST FEATURES	RAE
MLP	Energy	Slice_ID, WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Window_U_Value, SHGC	5.08
	Illuminance	Slice_ID, WWR_N, WWR_W, WWR_S, WWR_E, SHGC	17.5
	Value Function	Slice_ID, WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Window_U_Value, SHGC	14.99
SMOReg	Energy	Slice_ID, WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Window_U_Value, SHGC	11.36
	Illuminance	Slice_ID, WWR_N, WWR_S, WWR_E, Wall_U_Value	23.72
	Value Function	Slice, WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Window_U_Value, SHGC	19.13
RF	Energy	Slice_ID, WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Window_U_Value, SHGC	18.22
	Illuminance	Slice_ID, WWR_N, WWR_W, WWR_S, WWR_E, Window_U_Value	12.55
	Value Function	Slice_ID, WWR_N, WWR_W, WWR_S, WWR_E, Wall_U_Value, Window_U_Value, SHGC	18.05

Table 7:10 Mid-Rise Towers impact of Feature Selection on the RAE.

7.4.4 Hyperparameter Tuning

For the MLP algorithm the most impactful hyperparameter for Energy was hiddenLayers while the most impactful hyperparameter for Illuminance and the Value Function were learningRate. Two hidden layers with 30 neurons each gave the best performance for Energy. For Illuminance a learningRate of 0.01 gave the best performance while for the Value Function a learningRate of 0.1 gave the best performance.

Hyperparameter Tuning for MLP improved performance for the Energy objective by 1.17%, for the Illuminance objective by 2.73% and for the Value Function objective by 1.81%. On average Hyperparameter Tuning for MLP improved performance by 1.9%.

For the SMOReg algorithm the most impactful hyperparameter for Energy, Illuminance and the Value Function was kernel. In all cases the normalizedPolykernel gave the best performance. Performance for the Energy objective improved by 2.61%, for the Illuminance objective by 8.11% and for the Value Function objective by 5.9%. On average Hyperparameter Tuning for SMOReg improved performance by 5.54%.

For the RF algorithm the most impactful hyperparameter for Energy and Illuminance was numFeatures. The most impactful hyperparameter for the Value Function was iterations. For both Energy and Illuminance, the best value of numFeatures was 6, while for the Value Function the best value of iterations was 500 and 700. Performance for the Energy objective improved by 0.97%, for the Illuminance objective by 0.74% and for the Value Function by 0.26%. On average Hyperparameter Tuning for RF improved performance by 0.66%.

Overall Hyperparameter Tuning was effective for MLP the Energy objective. It was close to effective for MLP for the Illuminance and Value Function objectives. It was effective for SMOReg for the Energy objective and close to effective for the Illuminance and Value Function objectives. It was close to effective for the RF for the Illuminance objective and somewhat effective for the Energy objective and the Value Function objective. As with the Residential Towers reference project Hyperparameter Tuning was the most influential of the impact factors after Sample Size.

MLP	hiddenLayers					learningRate			
	(10,10)	(20,20)	(30,30)	(40,40)	(50,50)	0.001	0.01	0.1	0.2
Energy	5.59	4.13	3.91	3.95	4.19	11.23	8.03	4.39	4.67
Illuminance	17.26	18.73	19.43	18.8	18.13	23.66	14.62	15.05	15.66
Value Function	14.26	14.51	14.63	14.43	14	19.66	16.88	13.18	13.65

	momentum				Best Value	Hyper-parameter
	0	0.4	0.6	0.8		
Energy	4.98	5.08	5.41	6.54	3.91	hiddenLayers
Illuminance	16.16	17.35	18.29	23.38	14.62	learningRate
Value Function	14.11	14.99	17.03	21.55	13.18	learningRate

Table 7:11 Mid-Rise Towers impact of Hyperparameter Tuning on MLP.

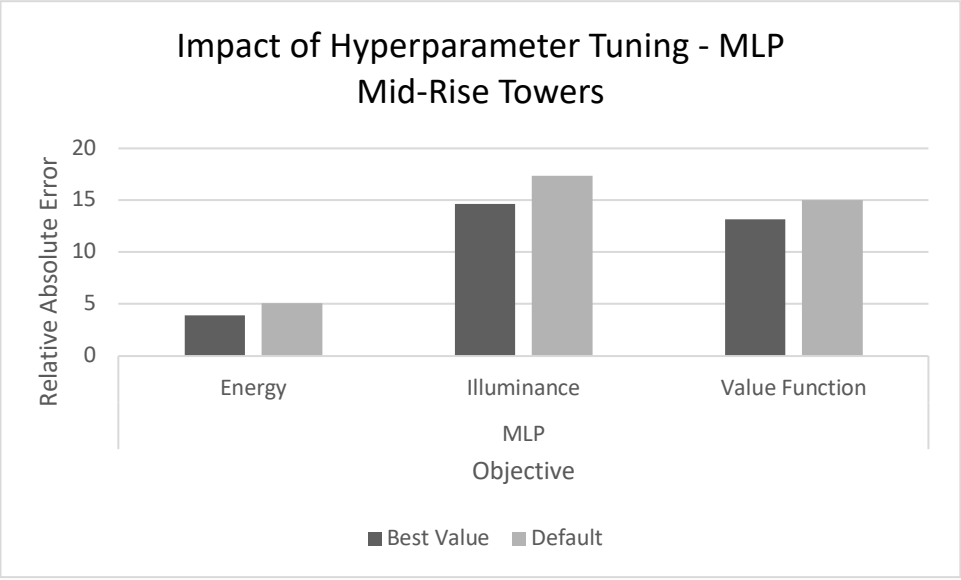


Figure 7:24 Mid-Rise Towers impact of Hyperparameter Tuning on MLP.

		Best Value	Default
MLP	**Energy**	3.91	5.08
	Illuminance	14.62	17.35
	Value Function	13.18	14.99

Table 7:12 Mid-Rise Towers comparative impact of Hyperparameter Tuning on MLP.

SMOReg	complexity			kernel			epsilonParameter			Best Value	Hyper-parameter
	0.001	*0.01*	*0.1*	*10*	*normalized-Polykernel*	*rbfKernel*	*0.01*	*0.1*	*1*		
Energy	49.58	11.5	11.36	11.36	8.75	9.66	11.3	28.9	107.95	8.75	Kernel
Illuminance	51.61	24.22	24.01	24.01	15.9	20.98	23.9	30.59	159.64	15.9	Kernel
Value Function	53.3	19.42	19.14	19.13	13.23	16.55	19.07	26.23	104.95	13.23	Kernel

Table 7:13 Mid-Rise Towers impact of Hyperparameter Tuning on SMOReg.

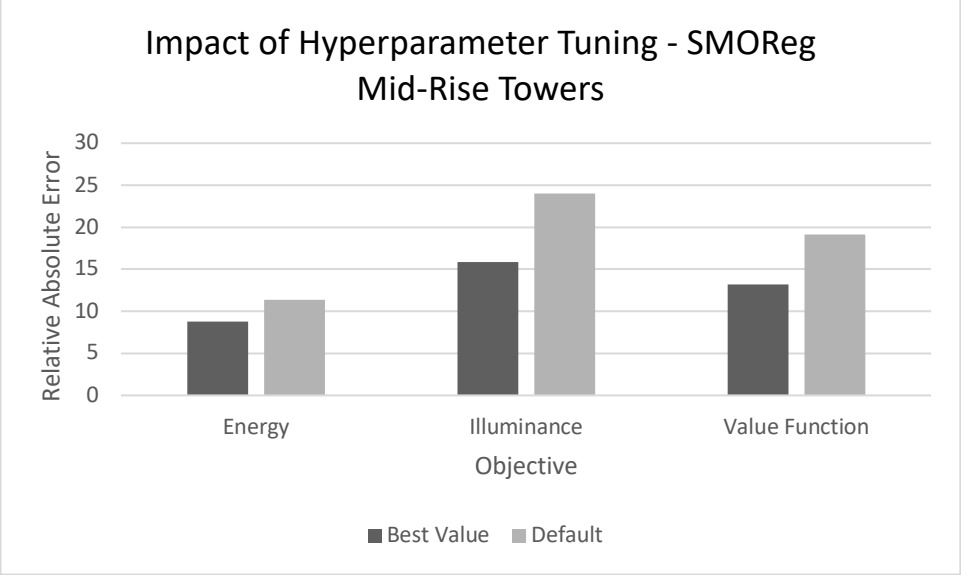


Figure 7:25 Mid-Rise Towers impact of Hyperparameter Tuning on SMOReg.

		Best Value	Default
SMOReg	**Energy**	8.75	11.36
	Illuminance	15.9	24.01
	Value Function	13.23	19.13

Table 7:14 Mid-Rise Towers comparative impact of Hyperparameter Tuning on SMOReg.

RF	iterations				numFeatures			
	100	***300***	***500***	***700***	***2***	***4***	***6***	***8***
Energy	18.22	17.95	17.87	17.83	25.38	18.22	17.25	17.73
Illuminance	13.56	13.28	13.26	13.24	21.88	13.56	12.82	13.14
Value Function	18.05	17.82	17.79	17.79	23.84	18.05	17.92	18.47

	maxDepth				Best Value	Hyperparameter	
	50	***100***	***150***	***200***			
Energy	18.22	18.22	18.22	18.22	17.25	**numFeatures**	
Illuminance	13.56	13.56	13.56	13.56	12.82	**numFeatures**	
Value Function	18.05	18.05	18.05	18.05	17.79	**iterations**	

Table 7:15 Mid-Rise Towers impact of Hyperparameter Tuning on RF.

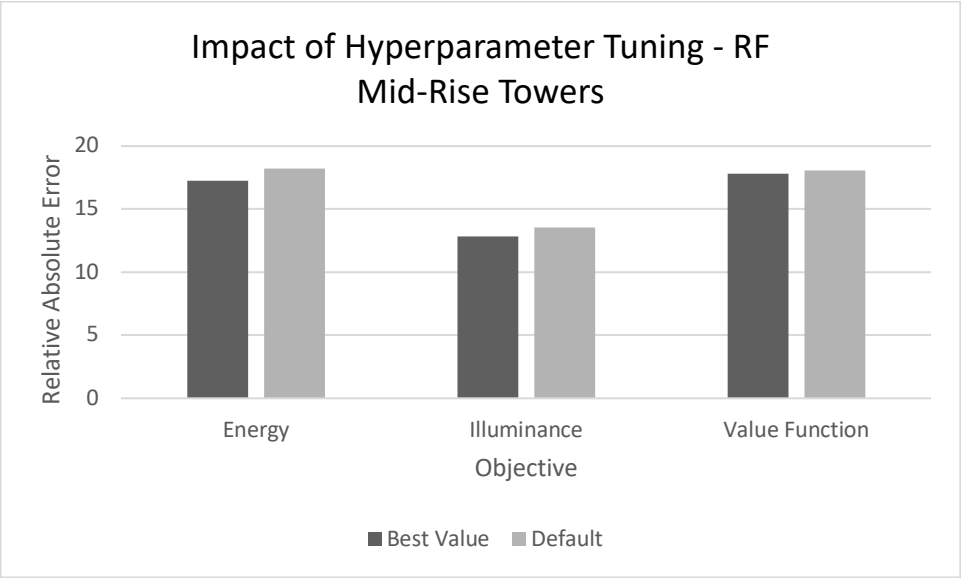


Figure 7:26 Mid-Rise Towers impact of Hyperparameter Tuning on RF.

		Best Value	Default
RF	Energy	17.25	18.22
	Illuminance	12.82	13.56
	Value Function	17.79	18.05

Table 7:16 Mid-Rise Towers comparative impact of Hyperparameter Tuning on RF.

7.5 Discussion

The best performing algorithm for all objectives was MLP. MLP improved to RAE 3.91% for the Energy objective and defaulted to RAE 14.62% for Illuminance and RAE 12.59% for the Value Function. At its best MLP performed 4.84% better than SMOReg and 13.34% better than RF for Energy; it performed 8.52% better than SMOReg and 1.92% worse than RF for Illuminance, and it performed 5.88% better than SMOReg and 5.35% better than RF for the Value Function. The best performing objective for MLP was Energy. On average it was 6.66% better than Illuminance and 9.48% better than the Value Function. Illuminance was 2.83% better than the Value Function.

The most important impact factor for MLP was Hyperparameter Tuning which, on average, improved by 0.15% compared to the default. The most important factor for SMOReg was Hyperparameter Tuning which improved by 4.90% compared to the default. The most important impact factor for RF was also Feature Selection which improved by 0.15% compared to the default. Across all algorithms the most important factor was Hyperparameter Tuning which was 1.73% better than the default. Hyperparameter Tuning was 22.02% better than Sensitivity Analysis and 2.55% better than Feature Selection.

Overall, MLP was effective for the Energy objective when the algorithm was improved by Sample Size, Feature Selection and Hyperparameter Tuning. It was close to effective for the Illuminance objective when improved by Sample Size and Hyperparameter Tuning, and somewhat effective when improved by Feature Selection. The algorithm was close to effective for the Value Function objective when it was improved by Sample Size, Feature Selection and Hyperparameter Tuning.

SMOReg was effective for the Energy objective when the algorithm was improved by Hyperparameter Tuning. It was close to effective for the Energy objective when improved by Sample Size and Feature Selection. It was somewhat effective for the Illuminance objective when improved by Hyperparameter Tuning. The Value Function was also close to effective when improved by Hyperparameter Tuning and somewhat effective when improved by Sample Size and Feature Selection.

RF was somewhat effective for the Energy objective when improved by Sample Size, Feature Selection and Hyperparameter Tuning. Illuminance was close to effective when improved by Sample Size, Feature Selection and Hyperparameter Tuning. The Value Function was somewhat effective when improved by Sample Size, Feature Selection and Hyperparameter Tuning.

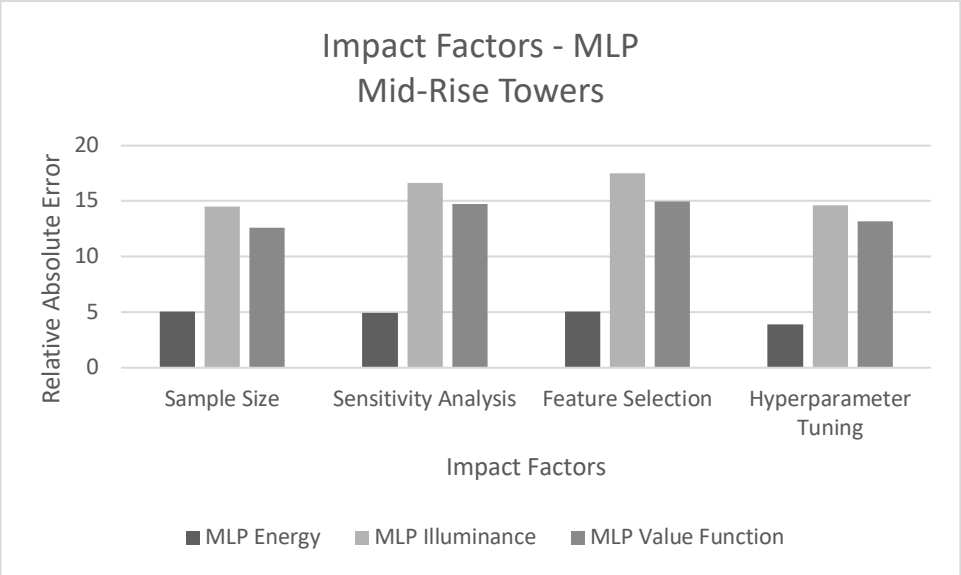


Figure 7:27 Mid-Rise Towers impact factors for MLP algorithm.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
MLP	**Energy**	5.08	9.52	5.08	3.91
	Illuminance	14.49	16.65	17.50	14.62
	Value Function	12.59	16.82	14.99	13.18

Table 7:17 Mid-Rise Towers impact factors for MLP algorithm.

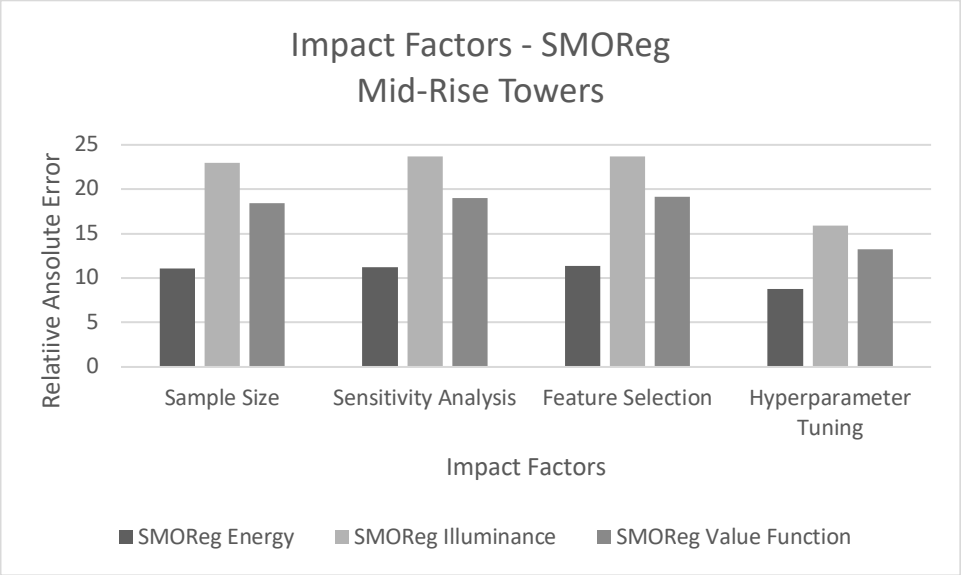


Figure 7:28 Mid-Rise Towers impact factors for SMOReg algorithm.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
SMOReg	**Energy**	11.11	12.02	11.36	8.75
	Illuminance	23.01	23.7	23.72	15.90
	Value Function	18.47	18.98	19.13	13.23

Table 7:18 Mid-Rise Towers impact factors for SMOReg algorithm.

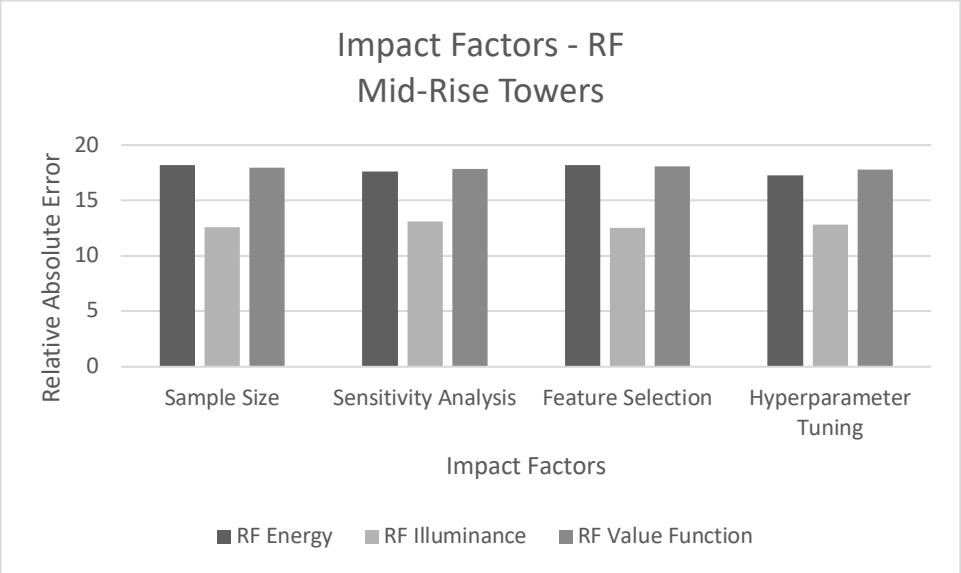


Figure 7:29 Mid-Rise Towers impact factors for Random Forests algorithm.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
RF	**Energy**	18.22	32.25	18.22	8.75
	Illuminance	12.57	13.54	12.55	15.9
	Value Function	17.94	19.62	18.05	13.23

Table 7:19 Mid-Rise Towers impact factors for Random Forests algorithm.

Chapter 8 Students' Residence Experiment

This chapter performs machine learning experiments on data from the Students' Residence reference project. We describe the project in terms of DSC and make use of the general experimental method of Chapter Five. However, the goal of DSC in this project is to help establish which of three design alternatives was most suitable in terms of building performance. Therefore, we investigate the performance of machine learning algorithms on a parametric analysis problem where the design geometry is variable, but the input parameters are constant across the alternatives. We assess the impact of sample size, sensitivity analysis, feature selection and hyperparameter tuning on algorithm performance. We find that machine learning algorithms are effective for all objectives. MLP is the best performing algorithm and hyperparameter tuning is the most influential impact factor.

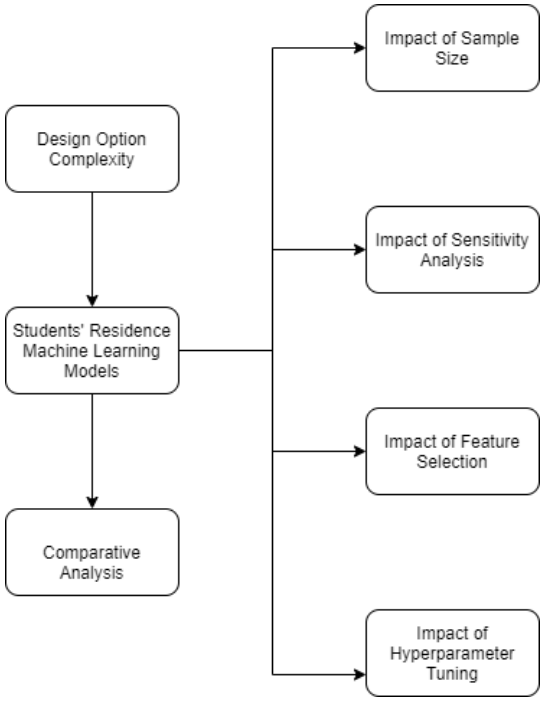


Figure 8:1 Diagram Summary of Chapter Eight.

8.1 Introduction

The Students' Residence project is a 330,000-sqft student housing, dining and conferencing facility located at a university in British Columbia. The project is located at the intersection of two important promenades and presented an opportunity to strengthen the campus circulation network. The design sought, among other criteria, to incorporate passive design principles, and reduce energy consumption, greenhouse gas emissions and address future climate resilience, creating a showcase project for the university (Figure 8:2).

The motivating design concepts included: create a compact campus; reinforce and create new activity hubs with open spaces and a mix of uses; create an animated place for walking, cycling, lingering and socializing; create better linkages between key places; enhance cycling and transit, and conserve and enhance natural areas, minimizing impact from building development.

The site and design principles used to realize the design intent included using massing and podium to frame the site circulation; using lower heights adjacent to circulation and increasing it farther away; aligning the buildings with the context and adjacent buildings to accommodate better seating and landscaping; upper

massing of the building to be setback to minimize height impact, and primary building frontages with active programming to be oriented towards circulation routes (Figure 8:3).



Figure 8:2 (Perkins&Will, 2019) A design impression of the Students' Residence project.

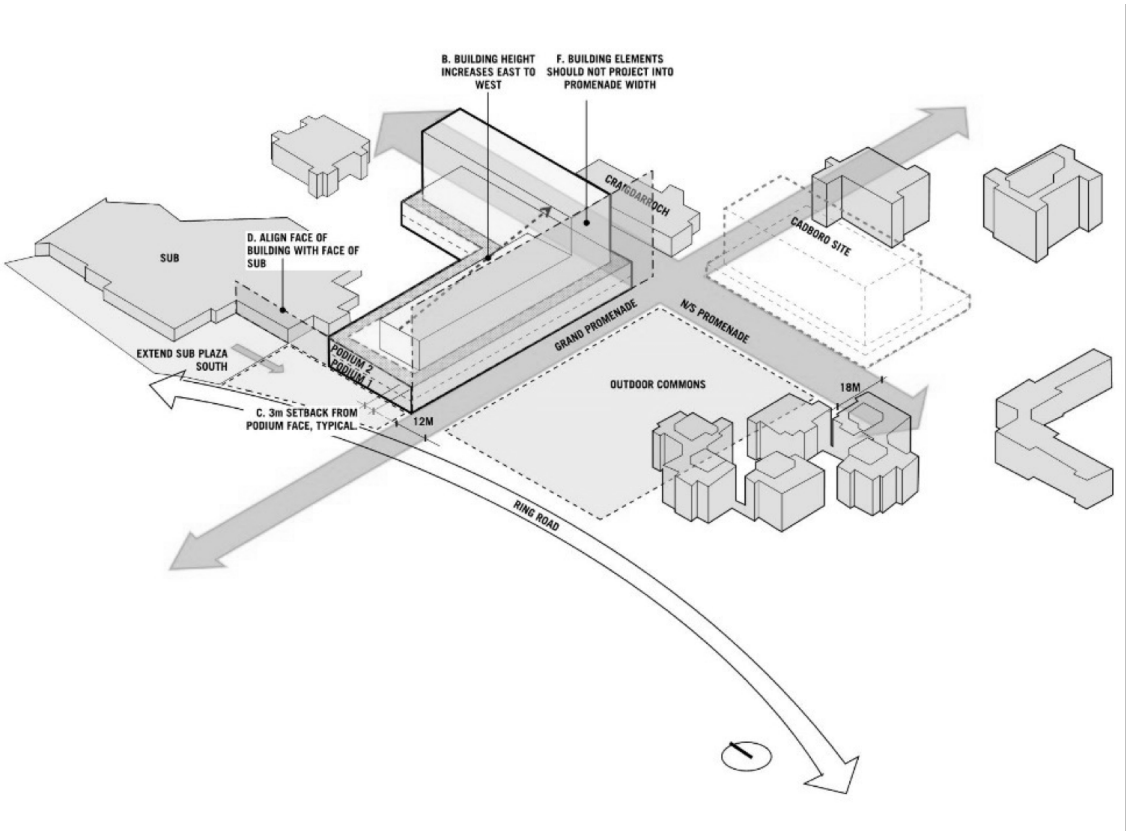


Figure 8:3 (Perkins&Will, 2019) Students' Residence site and design principles.

8.2 DSC Process

The design team engaged the DSC process in the early massing development stages. The key question was which of three massing alternatives was most suitable for design development in terms of optimal energy and daylighting performance. The investigation required three different DSC analytical models to be developed and cross-compared, shown in Figure 8:4.

The DSC process involved the four roles of Designer, Energy Analyst, Computational Designer and Data Analyst (Figure 8:5). The team developed three DSC analytical models based on a typical level from each of the design alternatives. They based the design space on the inputs shown in Table 8:1. A DOE was used to define three design spaces of 600 simulations each for the three massing options. The analysis metrics were Illuminance and Site Energy.

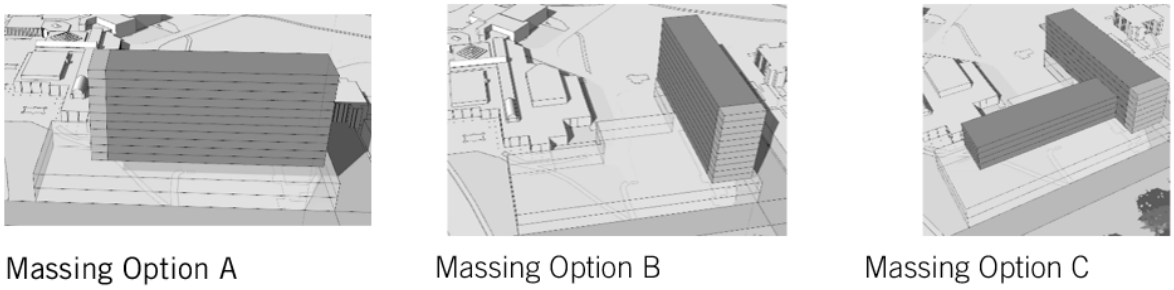


Figure 8:4 (Perkins&Will, 2019) Students' Residence massing options for DSC analysis.

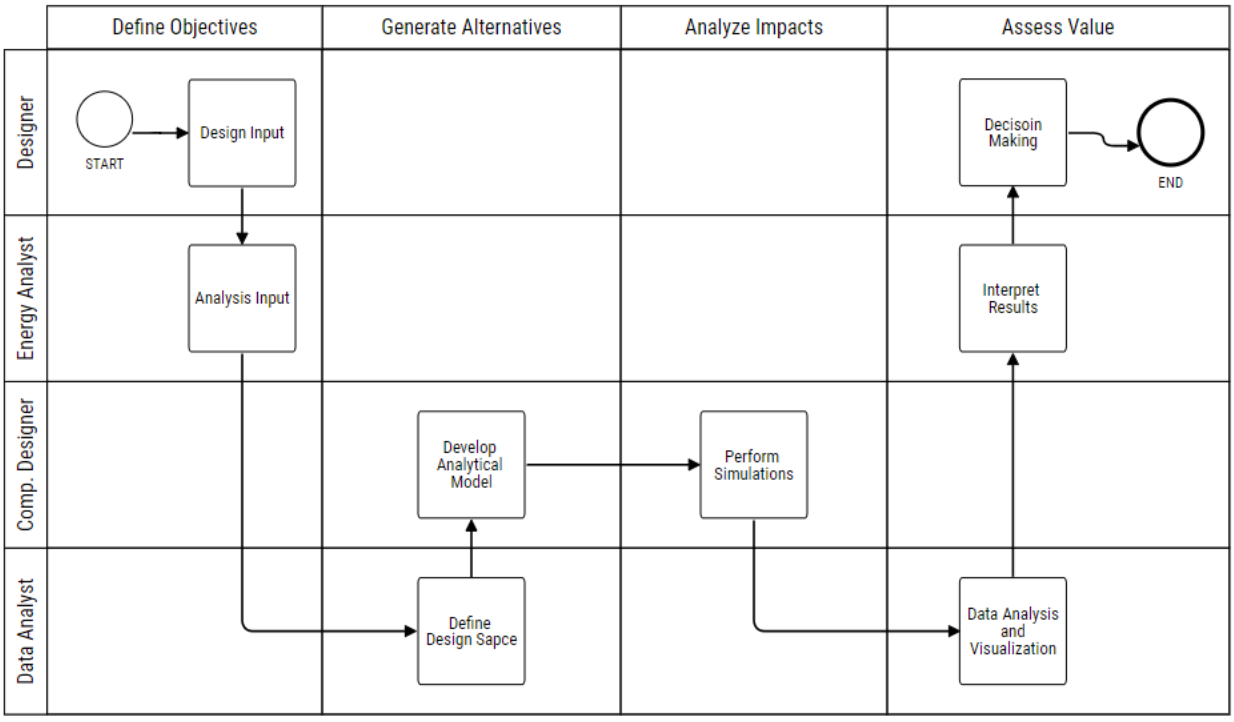


Figure 8:5 (Perkins&Will, 2019) Students' Residence DSC process.

8.2.1 Objective Definition

The main question of the Students' Residence DSC process was which of the three massing alternatives performed best in terms of energy and daylighting. Since much of the analytical data was the same for all three alternatives (location, weather files, energy standards, analysis constants, zone inputs, etc.)

it made sense, from a machine learning point of view, to create a single data model that could model all three together.

The full design space for each design option was 34,992 instances. This made for a total of 104,976 simulations which would take 145.8 days. This was not possible. Therefore, the DSC team used a sampling method to reduce the design space. They used a DOE approach to sample the design space for three independent samples of 600 instances each (for a total of 1800 instances). This was simulated in about 3 days.

8.2.2 Alternative Generation

The input parameters for the Students' Residence DSC were WWR along all facade orientations, shading device depth on all orientations, Wall U-values, Roof U-values, Window U-values and Window SHGC values (Table 8:2). The zonal inputs were lighting power density, number of occupants, total heat generated by occupants when fully occupied, and ventilation and equipment loads. Three distinct analytical models were involved (Figure 8:6) designated Options A, B and C. Option A & B had identical geometry but were oriented 90 degrees to each other while Option C had a completely different external geometry. For machine learning purposes all three were combined into one data model. This was achieved by using identical input parameters for each analytical model and distinguishing the alternatives by an *Option ID* parameter. Three 350 run simulations per design option were selected and the results combined into a single training dataset.

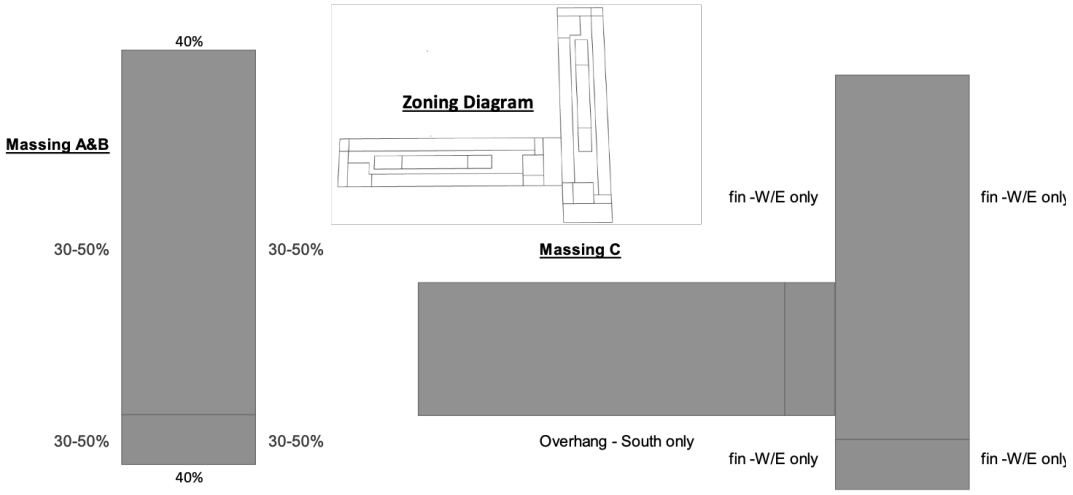


Figure 8:6 (Perkins&Will, 2019) Students' Residence massing options and zoning diagram.

8.2.3 Impact Analysis

Since the goal of the analysis was to establish which design alternative performs best for energy and daylighting, we performed a one-way MANOVA test with *Option ID* as the independent variable and Site Energy and Illuminance as the dependent variables. We used SPSS statistical software for the tests. Using the Wilk's Lambda test we observe that at 95% confidence levels ($p<0.05$) the options are significantly different from each other (Figure 8:7). Looking at the Test of Between-Subjects Effects we further observe that the options are significantly different from each other for both the Site Energy as well as the Illuminance objective (Figure 8:8). Since MANOVA does not report which option in particular is the best performer we perform a post-processing Tukey HSD test (Figure 8:9). We observe that that there are no significant differences between A and B for Site Energy and Illuminance. However, there are significant differences between A and C for both objectives as well as between B and C for both objectives. Option C performs significantly worse than A and B for Energy but performs significantly better for Illuminance. This is not conclusive.

Effect		Value	F	Hypothesis df	Error df	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power[d]
Intercept	Pillai's Trace	0.990	48707.714[b]	2.000	994.000	0.000	0.990	97415.427	1.000
	Wilks' Lambda	0.010	48707.714[b]	2.000	994.000	0.000	0.990	97415.427	1.000
	Hotelling's Trace	98.003	48707.714[b]	2.000	994.000	0.000	0.990	97415.427	1.000
	Roy's Largest Root	98.003	48707.714[b]	2.000	994.000	0.000	0.990	97415.427	1.000
Option	Pillai's Trace	0.154	41.377	4.000	1990.000	0.000	0.077	165.509	1.000
	Wilks' Lambda	0.848	42.730[b]	4.000	1988.000	0.000	0.079	170.919	1.000
	Hotelling's Trace	0.178	44.083	4.000	1986.000	0.000	0.082	176.333	1.000
	Roy's Largest Root	0.167	83.080[c]	2.000	995.000	0.000	0.143	166.159	1.000

Figure 8:7 Students' Residence one-way MANOVA Wilk's Lambda test.

Source		Type III Sum of Squares	df	Mean Square	F	Sig.	Partial Eta Squared	Noncent. Parameter	Observed Power[c]
Corrected Model	Site Energy	16596.416[a]	2	8298.208	72.891	0.000	0.128	145.782	1.000
	Illuminance	5045.780[b]	2	2522.890	52.919	0.000	0.096	105.839	1.000
Intercept	Site Energy	1838409.890	1	1838409.890	16148.545	0.000	0.942	16148.545	1.000
	Illuminance	4593431.306	1	4593431.306	96350.388	0.000	0.990	96350.388	1.000
Option	Site Energy	16596.416	2	8298.208	72.891	0.000	0.128	145.782	1.000
	Illuminance	5045.780	2	2522.890	52.919	0.000	0.096	105.839	1.000
Error	Site Energy	113274.466	995	113.844					
	Illuminance	47435.867	995	47.674					
Total	Site Energy	1978663.983	998						
	Illuminance	4665133.175	998						
Corrected Total	Site Energy	129870.882	997						
	Illuminance	52481.647	997						

Figure 8:8 Students' Residence Test of Between-Subjects Effects.

Dependent Variable			Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
						Lower Bound	Upper Bound
Site Energy	A	B	-1.38774588	0.835441588	0.221	-3.34871076	0.57321900
		C	-9.15867067*	0.814153995	0.000	-11.06966889	-7.24767244
	B	A	1.38774588	0.835441588	0.221	-0.57321900	3.34871076
		C	-7.77092479*	0.834863944	0.000	-9.73053381	-5.81131576
	C	A	9.15867067*	0.814153995	0.000	7.24767244	11.06966889
		B	7.77092479*	0.834863944	0.000	5.81131576	9.73053381
Illuminance	A	B	0.86977461	0.540634180	0.242	-0.39921254	2.13876177
		C	-4.25927625*	0.526858471	0.000	-5.49592878	-3.02262371
	B	A	-0.86977461	0.540634180	0.242	-2.13876177	0.39921254
		C	-5.12905086*	0.540260373	0.000	-6.39716060	-3.86094112
	C	A	4.25927625*	0.526858471	0.000	3.02262371	5.49592878
		B	5.12905086*	0.540260373	0.000	3.86094112	6.39716060

Figure 8:9 Students' Residence MANOVA, Tukey HSD test.

8.2.4 Value Assessment

Can we perhaps break the deadlock by observing the behavior of the options for the Value Function objective that balances both Site Energy and Daylight? To do this we perform a one-way ANOVA with the *Option ID* as the dependent variable and the *Value Function* as the independent variable. This showed that there were significant differences between the options at 95% confidence levels ($p<0.05$). Once again post processing using the Tukey HSD test showed that there were significant differences between A and B as well as between B and C. A (mean=75.87) performed significantly better than B (mean=72.89) which in turn performed significantly better than C (mean=68.64). This means that the design team should select design option A.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	9050.369	2	4525.185	29.852	0.000
Within Groups	150980.022	996	151.586		
Total	160030.392	998			

Figure 8:10 Students' Residence one-way ANOVA.

(I) Option		Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
A	B	2.975200650993742*	0.963221940275609	0.006	0.714310627782088	5.236090674205400
	C	7.223604373903513*	0.939468365198705	0.000	5.018469123980780	9.428739623826250
B	A	-2.975200650993741*	0.963221940275609	0.006	-5.236090674205400	-0.714310627782088
	C	4.248403722909771*	0.962554823837695	0.000	1.989079566125920	6.507727879693620
C	A	-7.223604373903513*	0.939468365198705	0.000	-9.428739623826250	-5.018469123980780
	B	-4.248403722909771*	0.962554823837695	0.000	-6.507727879693620	-1.989079566125920

Figure 8:11 Students' Residence ANOVA, Tukey HSD test.

8.2.5 Discussion

Once again, the sampling method raises sampling accuracy questions. In identifying the best performing option statistical methods have been used. It is possible that compared to sample statistics, full design space parameters could transform significant differences to insignificant and vice versa. This can have ramifications for design decision making. If what was thought a significant difference is in fact insignificant then designers have more latitude to pursue priorities other than performance. Conversely, if what was thought insignificant is actually a significant difference then designers must now prioritize performance above other considerations. We therefore need to investigate machine learning as a predictive model for developing the full design space.

8.3 Method

The general experimental method of this reference project has been described in Chapter Five. In this section we will present the specific data description and design space definition of the Students' Residence project. We also describe the method for applying machine learning to multiple geometrically distinct design alternatives.

Three distinct design spaces were developed for each of the design alternatives. Two of the design alternatives had similar geometry but different orientations while the third alternative had different geometry. Identical input parameters were used for all the design alternatives but, unlike for the Mid-Rise Tower reference project, different input parameter values were used for each design space. This allowed us to observe if introducing greater variability into the training dataset would significantly impact algorithm performance.

The machine learning training dataset consisted of all the simulation input parameters (Table 8:1) with an additional categorical parameter, *Option ID,* for the particular design alternative. The *Option ID* parameter had three possible values. The training dataset of 1,000 instances was a combination of three smaller datasets with about 333 instances each from the simulated design alternatives.

8.3.1 Data Description

The data used for the Students' Residence machine learning experiments is described in Table 8:1.

Input	**Data Type**	**Units**	**Description**
WWR_N	Numerical - Discrete	%	Window to wall ratio on the North façade
WWR_W	Numerical - Discrete	%	Window to wall ratio on the West façade
WWR_S	Numerical - Discrete	%	Window to wall ratio on the South façade
WWR_E	Numerical - Discrete	%	Window to wall ratio on the East façade
Wall U-Value	Numerical - Discrete	$W/m^2 K$	Wall thermal transmittance value
Roof U-Value	Numerical - Discrete	$W/m^2 K$	Roof thermal transmittance value
Shading N	Numerical - Discrete	feet	Balcony depth on the North façade
Shading W	Numerical - Discrete	feet	Balcony depth on the West façade
Shading S	Numerical - Discrete	feet	Balcony depth on the South façade
Shading E	Numerical - Discrete	feet	Balcony depth on the East façade
Window U-Value	Numerical - Discrete	$W/m^2 K$	Window thermal transmittance value
SHGC	Numerical - Discrete	-	Solar Heat Gain Coefficient
Option ID			Identification of the design option
Output	**Data Type**	**Units**	**Description**
Cooling	Numerical - Continuous	kWh/m^2	Cooling load
Heating	Numerical - Continuous	kWh/m^2	Heating load
Site Energy	Numerical - Continuous	kWh/m^2	Total Site Energy
Illuminance %	Numerical - Continuous	%	Percentage of usable floor area with 300 - 3000 lux illuminance

Table 8:1 Students' Residence data description.

Parameter	**Description**	**Range**	**Size**
WWR North	Window to wall ratio	40,50	2
WWR West	Window to wall ratio	40,50	2
WWR South	Window to wall ratio	40,50	2
WWR East	Window to wall ratio	40,50	2
Shade N	Shading Depth	0	1
Shade W	Shading Depth	0, 1, 2	3
Shade S	Shading Depth	0, 1, 2	3
Shade E	Shading Depth	0, 1, 2	3
Wall U-Value	Wall U Value	0.12, 0.15, 0.2	3
Roof U-Value	Roof U Value	0.1, 0.15, 0.18	3
Window U-Value	Window U Value	0.65, 0.85, 1.1	3
SHGC	Solar Heat Gain Coefficient	0.3, 0.5 ,0.7	3
Option ID	Design Option	A, B, C	3
			104,976

Table 8:2 Students' Residence design space definition.

8.3.2 Design Space Definition

The design space definition of the Students' Residence is shown in Table 8:2.

8.4 Result

8.4.1 Sample Size

The impact of Sample Size on Site Energy for the Students' Residence was that all algorithms perform better with a large sample (1000 instances) than with a small sample (100 instances) by 5.89% on average. The best performing algorithm was MLP with 1000 instances (RAE 1.15%) while the poorest performing was RF with 100 instances (RAE 22.94%). All algorithms were effective; MLP and SMOReg were effective with as few as 100 instances while RF was effective from 600 instances up.

The effect of Sample Size on Illuminance was that all algorithms performed better with a large sample size by 21.79% on average. The best performing algorithm was MLP with 900 instances (RAE 14.11%) while the poorest performing algorithm was SMOReg with 300 instances (RAE 62.07%). MLP was close to effective for Illuminance with 900 instances (RAE 14.11%). MLP was at least somewhat effective for Illuminance starting at 500 instances.

The effect of Sample Size on the multi-discipline Value Function was that all algorithms performed better with a large sample size by 12.56% on average. The best performing algorithm was MLP with 1000 instances (RAE 8.17%) while the poorest performing algorithm was RF with 100 instances (RAE 35.09%). MLP was effective for the Value Function starting at 500 instances (8.92% RAE) close to effective starting at 300 instances (13.29% RAE) and somewhat effective at 200 instances (15.19% RAE). RF was close to effective for the Value Function starting at 700 instances (14.13% RAE) and somewhat effective starting at 400 instances (18.43% RAE). SMOReg was not effective for the Value Function.

On average MLP performed 8.33% better than RF and 19% better than SMOReg. RF performed better than SMOReg by 10.68%. Overall MLP was effective for Energy and Value Function objectives. SMOReg was effective for the Energy objective but not for the Illuminance objective and the Value Function objective. RF was effective for the Energy objective and close to effective for the Value Function objective.

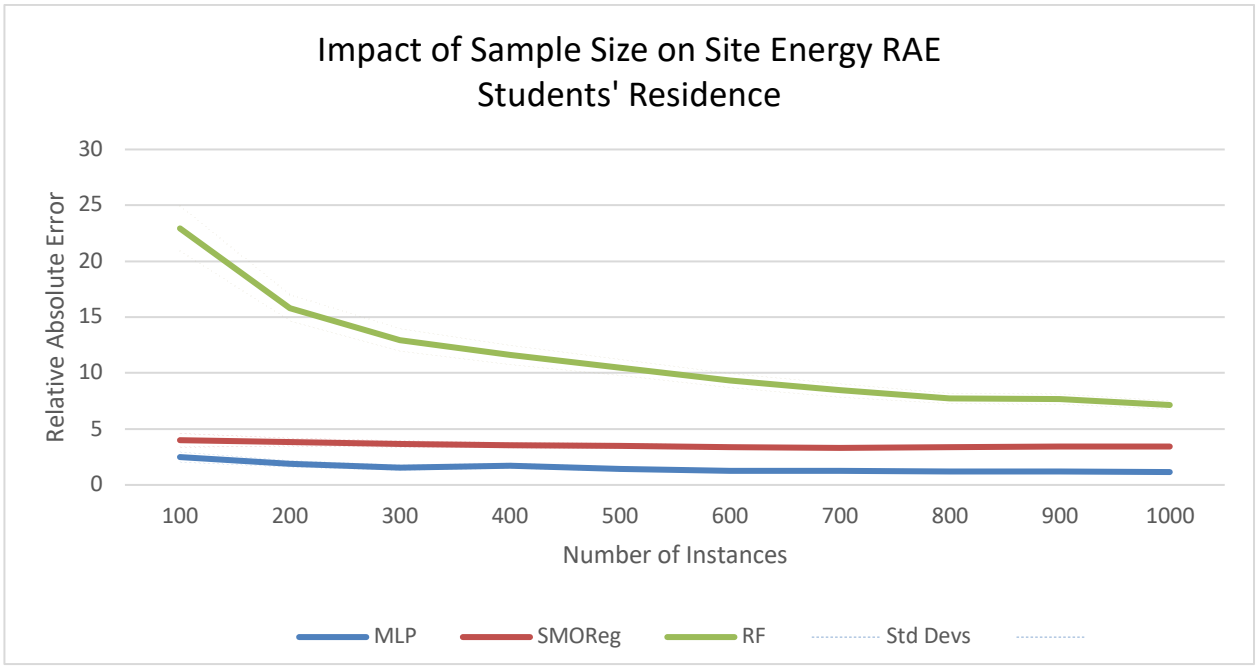


Figure 8:12 Students' Residence impact of Sample Size on the RAE of Site Energy.

Sample Size	100	200	300	400	500	600	700	800	900	1000
MLP	2.49	1.89	1.55	1.72	1.43	1.28	1.28	1.22	1.2	1.15
SMOReg	4	3.83	3.66	3.54	3.49	3.39	3.31	3.35	3.44	3.46
RF	22.94	15.83	12.95	11.63	10.49	9.35	8.47	7.73	7.65	7.14

Table 8:3 Students' Residence impact of Sample Size on the RAE of Site Energy.

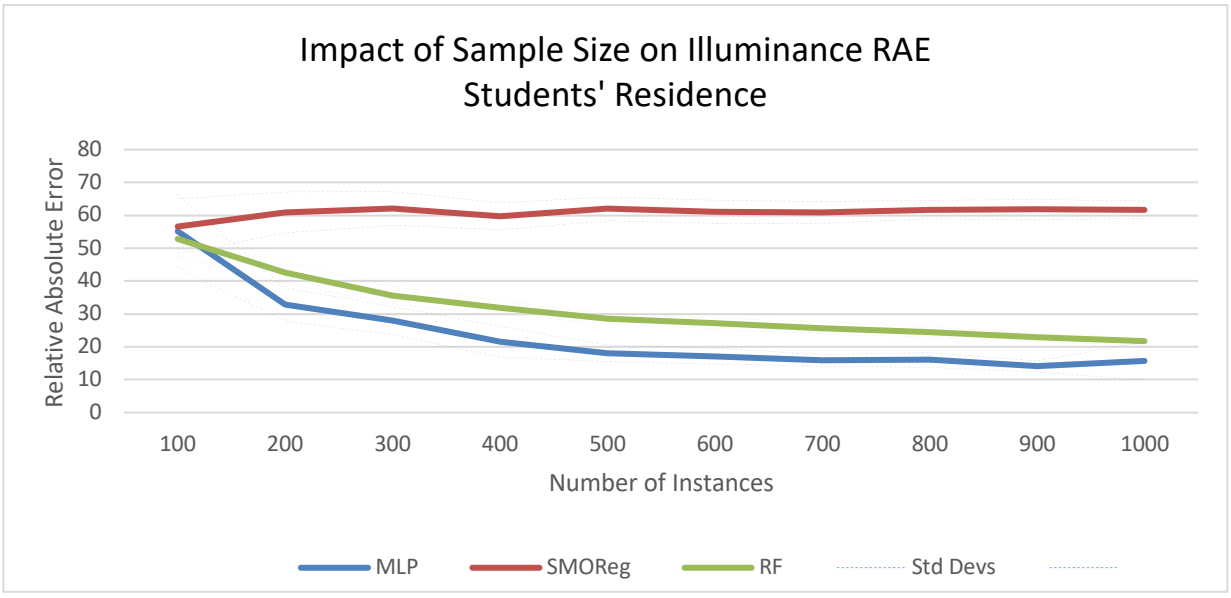


Figure 8:13 Students' Residence impact of Sample Size on the RAE of Illuminance.

Sample Size	100	200	300	400	500	600	700	800	900	1000
MLP	55.15	32.78	28.02	21.51	18.05	17.12	15.95	16.02	14.11	15.75
SMOReg	56.58	60.93	62.07	59.68	62.01	61.05	60.9	61.68	61.79	61.68
RF	52.81	42.63	35.59	31.92	28.52	27.1	25.72	24.45	22.86	21.72

Table 8:4 Students' Residence impact of Sample Size on the RAE of Illuminance.

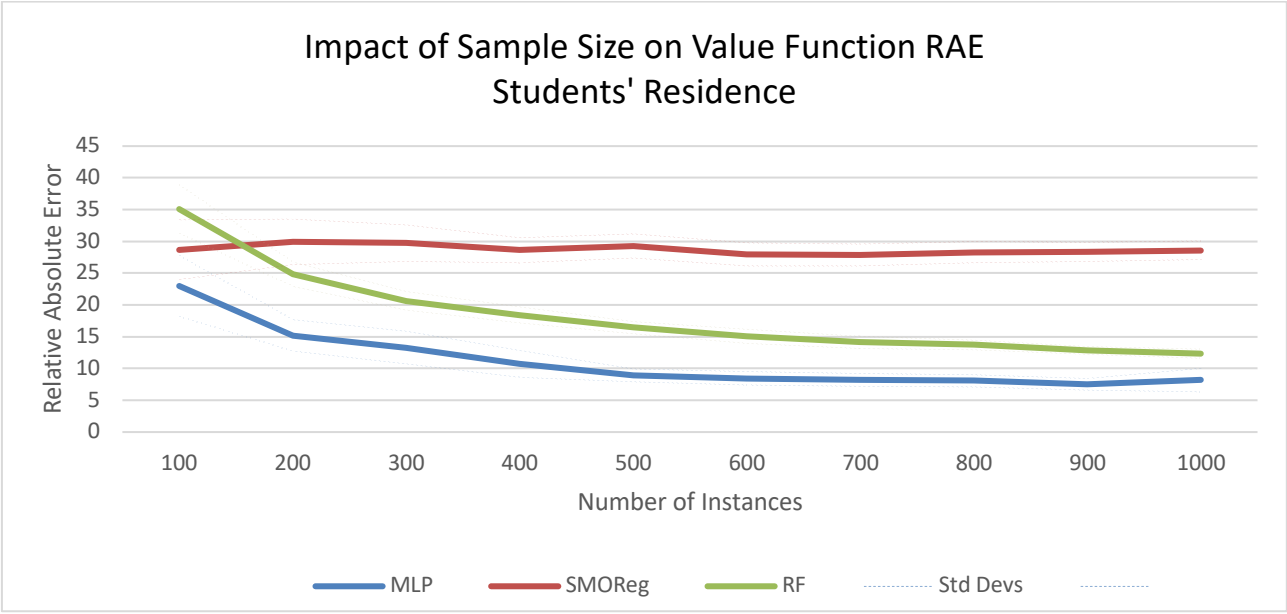


Figure 8:14 Students' Residence impact of Sample Size on the RAE of the Value Function

Sample Size	100	200	300	400	500	600	700	800	900	1000
MLP	22.99	15.19	13.29	10.71	8.92	8.44	8.21	8.09	7.51	8.17
SMOReg	28.7	29.94	29.76	28.63	29.31	27.98	27.87	28.28	28.39	28.61
RF	35.09	24.86	20.65	18.43	16.43	15.06	14.13	13.71	12.87	12.33

Table 8:5 Students' Residence impact of Sample Size on the RAE of the Value Function.

8.4.2 Sensitivity Analysis

Sensitivity Analysis established that *Option* was overwhelmingly the most sensitive parameter impacting Site Energy. This was because each of the options had a distinctive range of Energy values. The best performing algorithm for Energy RAE was MLP with 12 parameters (RAE 1.16%) while the worst performing algorithm was again MLP with a single parameter (RAE 13.39%). MLP did not improve Energy RAE under sensitivity analysis. SMOReg did not improve Energy RAE (0.01%) while RF improved Energy RAE by 4.51% using 8 parameters. MLP was effective starting at 2 parameters and close to effective with 1 parameter. SMOReg was effective starting at 2 parameters and close to effective with 1 parameter. RF was also effective starting at 2 parameters and close to effective with 1 parameter.

For Illuminance Sensitivity Analysis, Shading on the East and West facades were the most sensitive parameters. The order of sensitivity is shown in Figure 8:16. The best performing algorithm was RF with 8 parameters (RAE 9.96%) while the worst performing algorithm was MLP with a single parameter (RAE 94.99%). MLP did not improve Illuminance RAE under sensitivity analysis (0.22%). SMOReg also did not improve Illuminance RAE (-0.01%). RF, however, improved Illuminnace RAE by 11.76% using 8 parameters. RF was effective under Illuminance with 8 parameters. It was close to effective with 9 parameters and somewhat effective with 10 and 11 parameters. MLP was somewhat effective starting at 8 parameters. SMOReg was not effective.

For the Value Function, Option was the only sensitive parameter. The best performing algorithm was MLP with 11 parameters (RAE 8.08%) while the worst performing algorithm was MLP with a single parameter (RAE 49.37%). MLP did not improve Value Function RAE (0-0.09%) and SMOReg also did not improve Value Function RAE (0.0%). RF improved Value Function by 4.09% using 10 parameters. MLP was effective starting at 10 parameters, close to effective starting at 7 parameters and somewhat effective with 6 parameters. RF was effective starting at 10 parameters, close to effective starting at 7 parameters and somewhat effective at 6 parameters. SMOReg was not effective.

MLP was effective for Energy and the Value Function and somewhat effective for Illuminance. SMOReg was effective for Energy. RF was effective for all objectives.

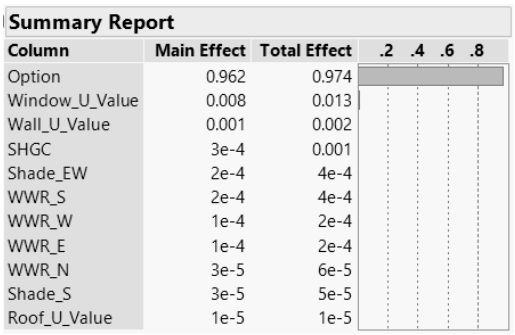


Figure 8:15 Students' Residence Sensitivity Analysis, Energy.

Summary Report

Column	Main Effect	Total Effect
Shade_EW	0.306	0.321
Shade_S	0.29	0.305
WWR_W	0.104	0.119
WWR_N	0.096	0.111
Option	0.074	0.088
WWR_S	0.019	0.029
WWR_E	0.009	0.015
Roof_U_Value	0.001	0.002
Wall_U_Value	4e-4	0.001
SHGC	3e-4	0.001
Window_U_Value	3e-4	0.001

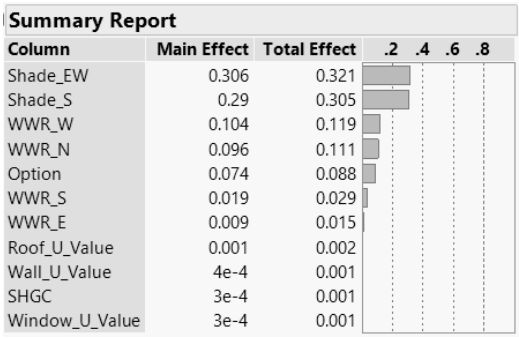


Figure 8:16 Students' Residence Sensitivity Analysis, Illuminance.

Summary Report

Column	Main Effect	Total Effect
Option	0.852	0.867
Shade_EW	0.035	0.048
Shade_S	0.023	0.033
WWR_N	0.01	0.016
WWR_W	0.006	0.011
Window_U_Value	0.005	0.009
WWR_S	0.002	0.003
Wall_U_Value	0.001	0.002
WWR_E	0.001	0.002
SHGC	3e-4	0.001
Roof_U_Value	1e-4	3e-4

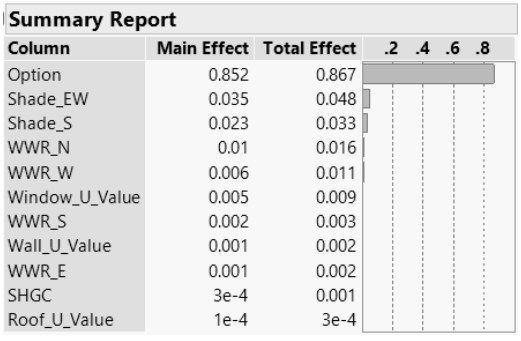


Figure 8:17 Students' Residence Sensitivity Analysis, Value Function.

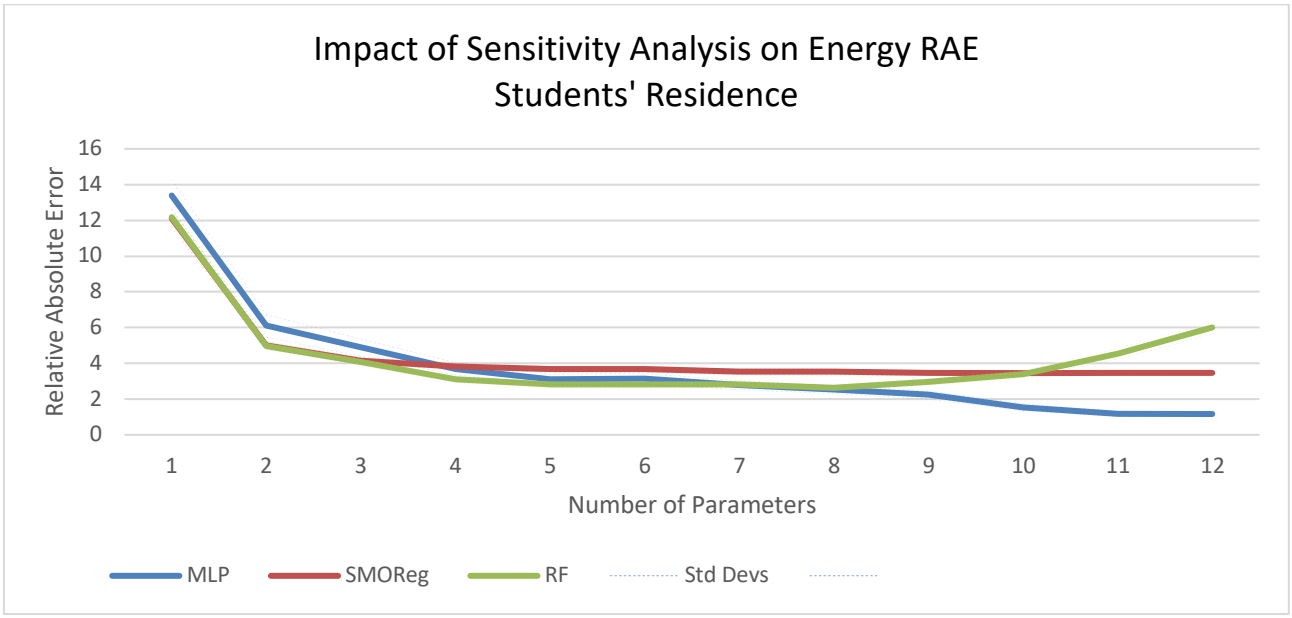


Figure 8:18 Students' Residence impact of Sensitivity Analysis on the RAE of Energy.

	1	2	3	4	5	6	7	8	9	10	11	12
MLP	13.39	6.11	4.89	3.68	3.1	3.13	2.79	2.53	2.24	1.54	1.17	1.16
SMOReg	12.11	5.01	4.15	3.82	3.66	3.66	3.52	3.52	3.46	3.45	3.46	3.46
RF	12.18	4.98	4.08	3.09	2.81	2.82	2.81	2.63	2.96	3.4	4.55	5.99

Table 8:6 Students' Residence impact of Sensitivity Analysis on the RAE of Energy.

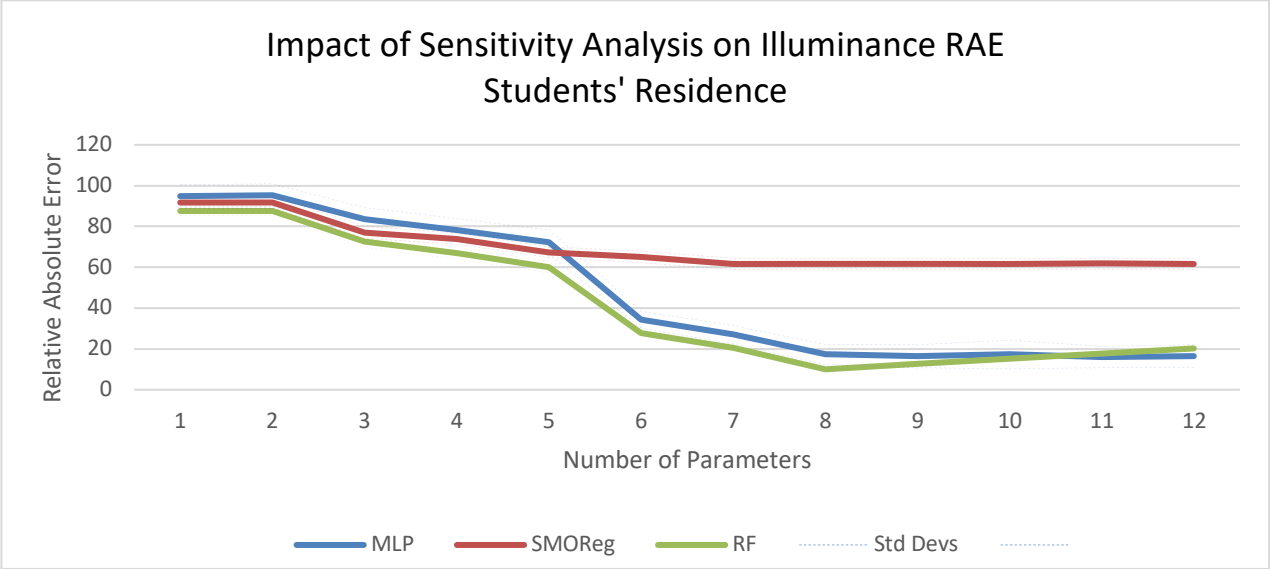


Figure 8:19 Students' Residence impact of Sensitivity Analysis on the RAE of Illuminance.

	1	2	3	4	5	6	7	8	9	10	11	12
MLP	94.99	95.3	83.53	78.32	72.3	34.28	26.98	17.23	16.41	17.35	15.97	16.48
SMOReg	91.75	91.76	77	73.97	67.35	65.06	61.69	61.67	61.67	61.72	61.78	61.68
RF	87.61	87.61	72.54	66.88	59.98	27.8	20.42	9.96	12.52	15.22	17.75	20.3

Table 8:7 Students' Residence impact of Sensitivity Analysis on the RAE of Illuminance.

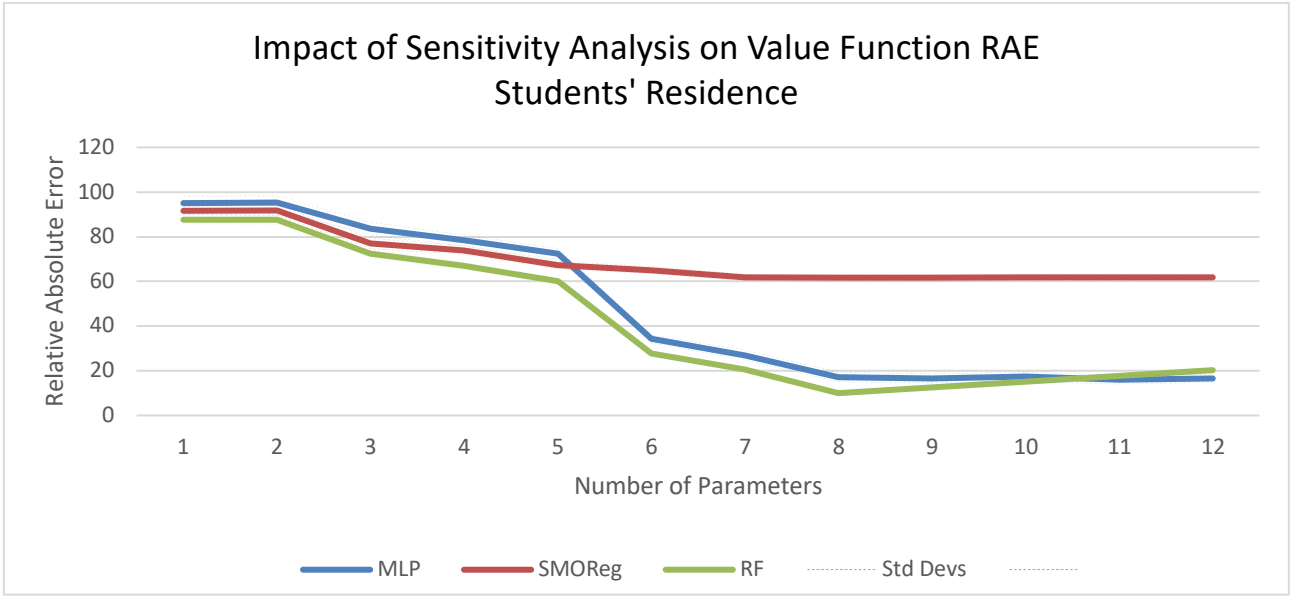


Figure 8:20 Students' Residence impact of Sensitivity Analysis on the RAE of Value Function

	1	2	3	4	5	6	7	8	9	10	11	12
MLP	49.37	41.1	40.58	31.56	31.56	18.85	15.42	13.4	12.68	9.1	8.08	8.1
SMOReg	44.73	40.36	40.36	34.82	34.82	30.82	29.86	28.85	28.85	28.79	28.61	28.61
RF	44.72	35.44	35.44	26.52	26.52	15.63	13.51	10.84	10.73	8.24	9.37	11.17

Table 8:8 Students' Residence impact of Sensitivity Analysis on the RAE of the Value Function.

8.4.3 Feature Selection

For the Students' Residence, Table 8:9 established the best features for each algorithm and for each objective. The best performing algorithm for feature selection was MLP for the Energy objective (RAE 1.18%). The poorest performing algorithm was SMOReg for the Illuminance objective (RAE 61.36%). On average, for all objectives, MLP performed 12.29% better than RF and 26.89% better than SMOReg. RF performed 14.6% better than SMOReg.

MLP had an insignificant decline in performance compared to the default baseline for Energy (0.03%), a significant decline for Illuminance (7.22%), and an insignificant decline for the Value Function (0.09%). RF improved in performance for Energy (4.68%), Illuminance (8.65%) and the Value Function (5.08%). SMOReg had no significant change for Energy (0.03%) and the Value Function (0.19%), but a significant decline for Illuminance (33.66%).

Overall MLP was effective for Energy and the Value Function after feature selection, but somewhat effective for Illuminance. RF was also effective for all objectives. SMOReg was effective for the Energy objective.

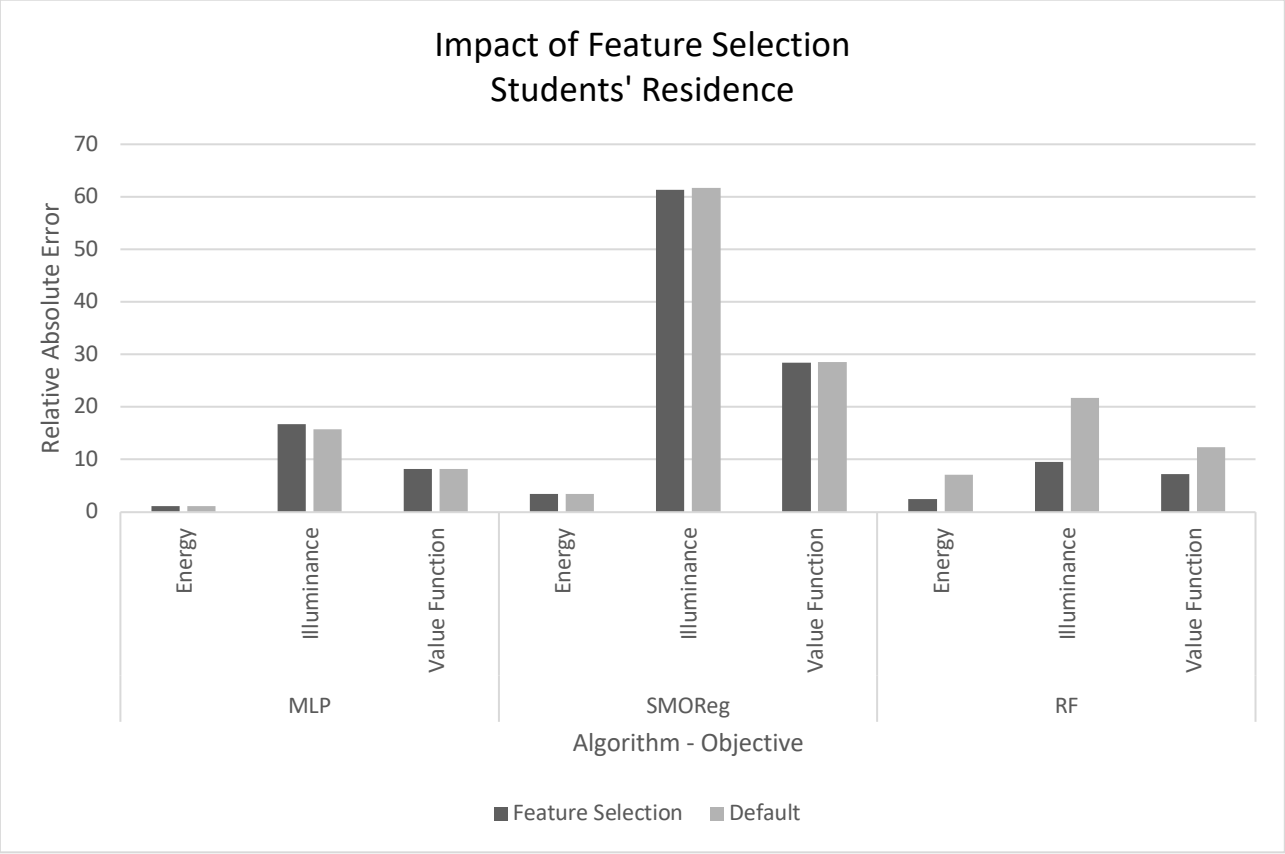


Figure 8:21 Students' Residence impact of Feature Selection on the RAE.

ALGORITHM	OBJECTIVE	BEST FEATURES	RAE
MLP	Energy	Option, WWR N, WWR W, WWR S, WWR E, Shade N, Shade W, Shade S, Wall U Value, Window U Value, SHGC	1.18
	Illuminance	Option, WWR N, WWR W, WWR S, WWR E, Shade N, Shade S, Shade E, Wall U Value, Roof U Value	16.8
	Value Function	Option, WWR N, WWR W, WWR S, WWR E, Shade W, Shade S, Shade East, Wall U Value, Roof U Value, Window U Value, SHGC	8.26
SMOReg	Energy	Option, WWR N, WWR W, WWR S, WWR E, Shade S, Shade E, Wall U Value, Window U Value, SHGC	3.43
	Illuminance	Option, WWR N, WWR W, WWR S, WWR E, Shade S, Shade E	61.36
	Value Function	Option, WWR N, WWR W, WWR S, WWR E, Shade W, Shade S, Shade E, Wall U Value, Window U Value, SHGC	28.42
RF	Energy	Option, WWR N, WWR S, Shade N, Shade E, Wall U Value, Window U Value, SHGC	2.46
	Illuminance	Option, WWR N, WWR W, WWR S, WWR E, Shade N, Shade S, Shading E	9.58
	Value Function	Option, WWR N, WWR W, WWR S, WWR E, Shade S, Shade E, Window U Value	7.25

Table 8:9 Students' Residence impact of Feature Selection on the RAE.

8.4.4 Hyperparameter Tuning

For the MLP algorithm the most impactful hyperparameter for all algorithms was the learningRate. For the Energy objective a learningRate of 0.2 gave the best performance while for the Illuminance and Value Function objectives a learningRate of 0.1 gave the best performance. Hyperparameter Tuning for MLP improved performance for the Energy objective by 0.04%, for the and for the Value Function objective by 1.29%. It decreased performance for the Illuminance objective by 3.89%.

For the SMOReg algorithm the most impactful hyperparameter for Energy, Illuminance and the Value Function was kernel. For the Energy objective the RBFKernel gave the best performance. For the Illuminance and Value Function objectives the normalizedPolykernel gave the best performance. Performance for the Energy objective improved by 2.19% and for the Value Function objective by 13.84%. The Illuminance objective reduced by 2.78%.

For the RF algorithm the most impactful hyperparameter for all objectives was numFeatures. For all objectives the best value of numFeatures was 8. Performance for the Energy objective improved by 4.75% and for the Value Function by 5.42%. For the Illuminance objective performance reduced by 4.14%.

Overall Hyperparameter Tuning was effective for MLP for the Energy objective and the Value Function objective. It was close to effective for MLP for the Illuminance objective. It was effective for SMOReg for the Energy objective and close to effective for the Value Function objective. It was effective for RF for the Energy and Value Function objectives. It was close to effective for the Value Function.

MLP	hiddenLayers					learningRate			
	(10,10)	(20,20)	(30,30)	(40,40)	(50,50)	0.001	0.01	0.1	0.2
Energy	1.14	1.24	1.14	1.15	1.31	3.61	1.7	1.16	1.11
Illuminance	14.41	15.42	16.64	17.13	17.48	61.77	13.81	13.47	14.8
Value Function	7.5	7.74	8.3	8.37	8.37	22.17	9.32	6.88	7.4

	momentum				Best Value	Hyper-parameter
	0	0.4	0.6	0.8		
Energy	1.15	1.22	1.32	2.59	1.11	learningRate
Illuminance	15.75	17.06	22.46	66.06	13.47	learningRate
Value Function	8.17	9.18	10.64	31.18	6.88	learningRate

Table 8:10 Students' Residence impact of Hyperparameter Tuning on MLP.

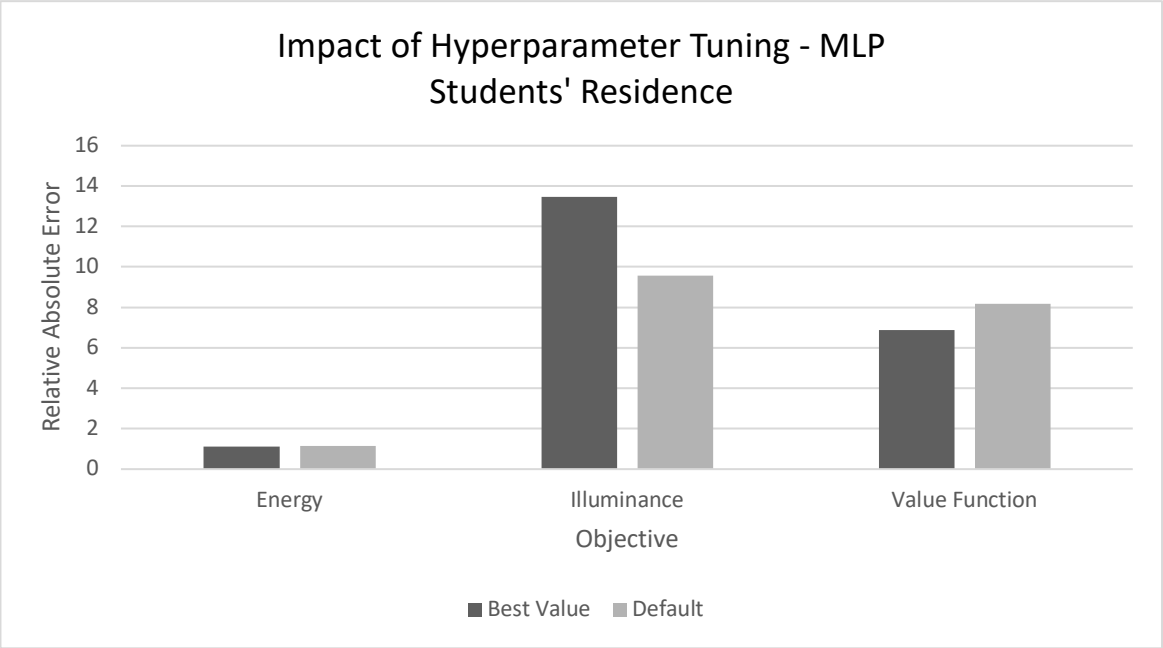


Figure 8:22 Students' Residence impact of Hyperparameter Tuning on MLP.

		Best Value	Default
MLP	**Energy**	1.11	1.15
	Illuminance	13.47	9.58
	Value Function	6.88	8.17

Table 8:11 Students' Residence comparative impact of Hyperparameter Tuning on MLP.

	complexity					kernel		epsilonParameter			Best Value	Hyper-Parameter
SMOReg	*0.001*	*0.01*	*0.1*	*1*	*10*	*normalized-Polykernel*	*RBFKernel*	*0.01*	*0.1*	*1*		
Energy	28.87	3.51	3.46	3.46	3.46	5.58	1.27	3.43	24.08	106.98	1.27	kernel
Illuminance	70.1	61.88	61.69	61.68	61.67	30.48	53.05	61.67	62.78	129.91	30.48	kernel
Value Function	40.87	28.73	28.59	28.61	28.62	14.77	22.79	28.58	30.43	100.53	14.77	kernel

Table 8:12 Students' Residence impact of Hyperparameter Tuning on SMOReg.

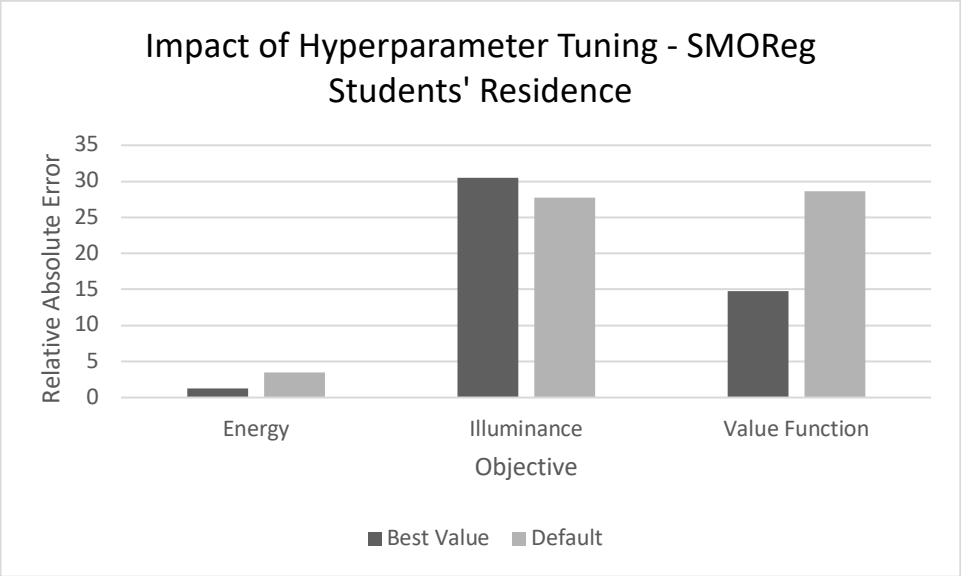


Figure 8:23 Students' Residence impact of Hyperparameter Tuning on SMOReg.

		Best Value	Default
SMOReg	**Energy**	1.27	3.46
	Illuminance	30.48	27.7
	Value Function	14.77	28.61

Table 8:13 Students' Residence comparative impact of Hyperparameter Tuning on SMOReg.

RF	iterations				numFeatures			
	100	*300*	*500*	*700*	*2*	*4*	*6*	*8*
Energy	7.14	6.91	6.85	6.82	24.45	7.14	2.93	2.37
Illuminance	21.72	21.53	21.47	21.44	36.5	21.72	15.12	12.37
Value Function	12.33	12.07	11.99	11.98	28.09	12.33	7.97	6.9
	maxDepth				**Best Value**	**Hyperparameter**		
	50	*100*	*150*	*200*				
Energy	7.14	7.14	7.14	7.14	2.37	numFeatures		
Illuminance	21.72	21.72	21.72	21.72	12.37	numFeatures		
Value Function	12.33	12.33	12.33	12.33	6.91	numFeatures		

Table 8:14 Students' Residence impact of Hyperparameter Tuning on RF.

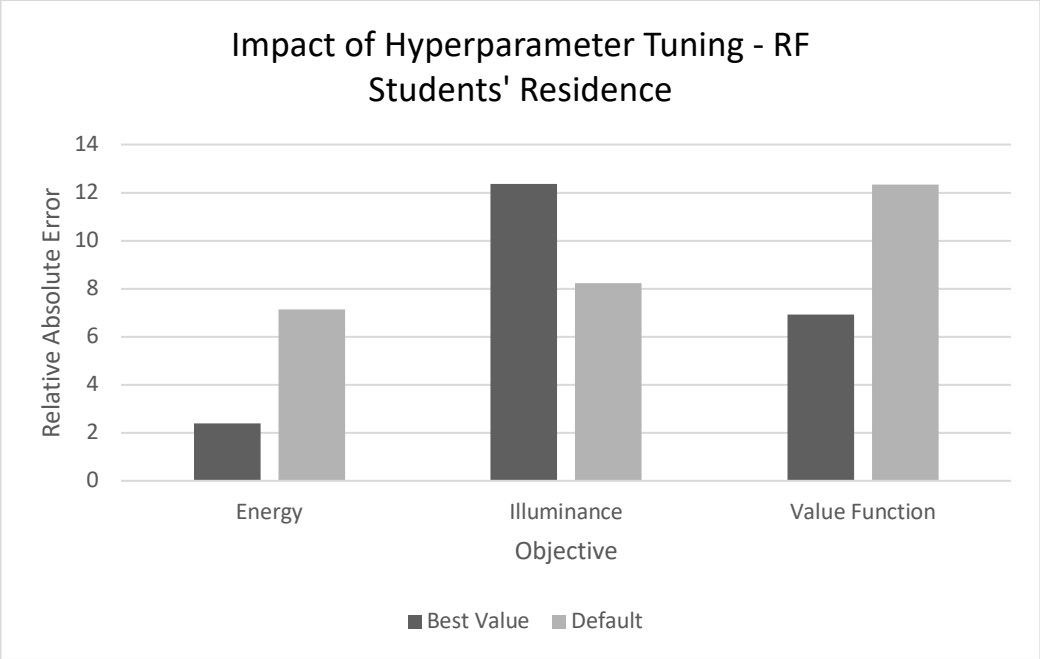


Figure 8:24 Students' Residence impact of Hyperparameter Tuning on RF.

		Best Value	Default
RF	**Energy**	2.39	7.14
	Illuminance	12.37	8.23
	Value Function	6.9	12.33

Table 8:15 Students' Residence comparative impact of Hyperparameter Tuning on RF.

8.5 Discussion

The best performing algorithm for all objectives was MLP. MLP improved to RAE 3.91% for the Energy objective and defaulted to RAE 14.62% for Illuminance and RAE 12.59% for the Value Function. At its best MLP performed 4.84% better than SMOReg and 13.34% better than RF for Energy; it performed 8.52% better than SMOReg and 1.92% worse than RF for Illuminance, and it performed 5.88% better than SMOReg and 5.35% better than RF for the Value Function. The best performing objective for MLP was Energy. On average it was 6.66% better than Illuminance and 9.48% better than the Value Function. Illuminance was 2.83% better than the Value Function.

The most important impact factor for MLP was hyperparameter tuning which, on average, improved by 0.15% compared to the default. The most important factor for SMOReg was also hyperparameter tuning which improved by 4.90% compared to the default. The most important impact factor for RF was feature selection which improved by 0.15% compared to the default. Across all algorithms the most important factor was hyperparameter tuning which was 1.73% better than the default. Hyperparameter Tuning was 22.02% better than sensitivity analysis and 2.55% better than feature selection.

Overall, MLP was effective for the Energy objective by default and when improved by feature selection and hyperparameter tuning. It was close to effective for the Illuminance objective by default and when improved by hyperparameter tuning, and somewhat effective when improved by feature selection. The algorithm was close to effective for the Value Function objective by default and when it was improved by feature selection and hyperparameter tuning.

SMOReg was effective for the Energy objective when the algorithm was improved by hyperparameter tuning. It was close to effective for the Energy objective by default and when improved by feature selection. It was somewhat effective for the Illuminance objective when improved by hyperparameter tuning. The Value Function was also close to effective when improved by hyperparameter tuning and somewhat effective by default and when improved by feature selection.

RF was somewhat effective for the Energy objective by default and when improved by feature selection and hyperparameter tuning. Illuminance was close to effective by default and when improved by feature selection and hyperparameter tuning. The Value Function was somewhat effective by default and when improved by feature selection and hyperparameter tuning.

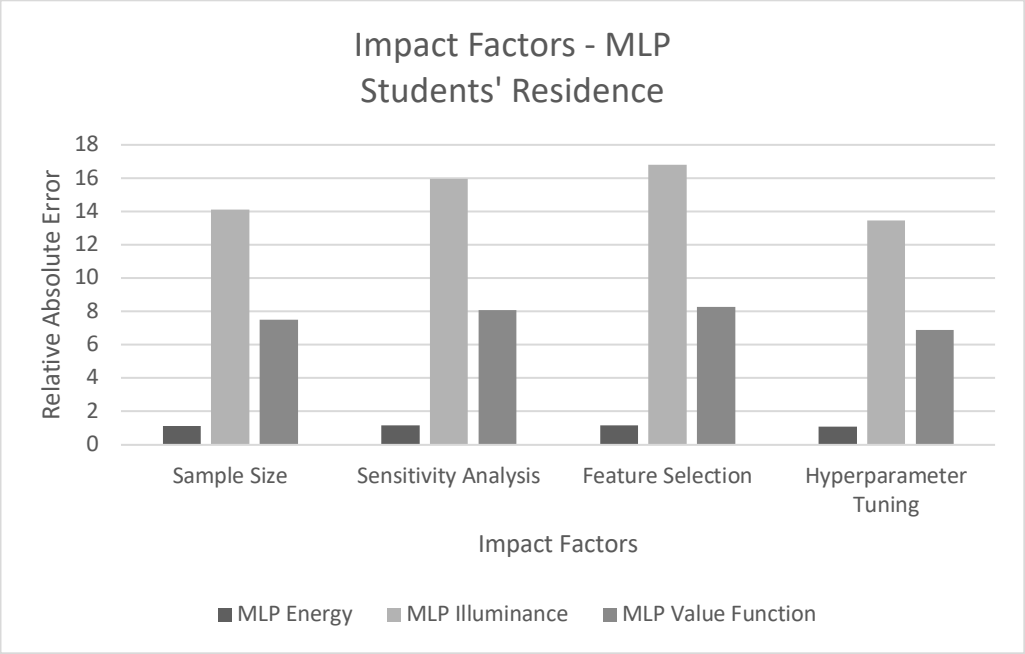


Figure 8:25 Students' Residence impact factors for MLP algorithm.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
MLP	**Energy**	1.15	1.17	1.18	1.11
	Illuminance	8.83	15.97	16.8	13.47
	Value Function	7.51	8.08	8.26	6.88

Table 8:16 Students' Residence impact factors for MLP algorithm.

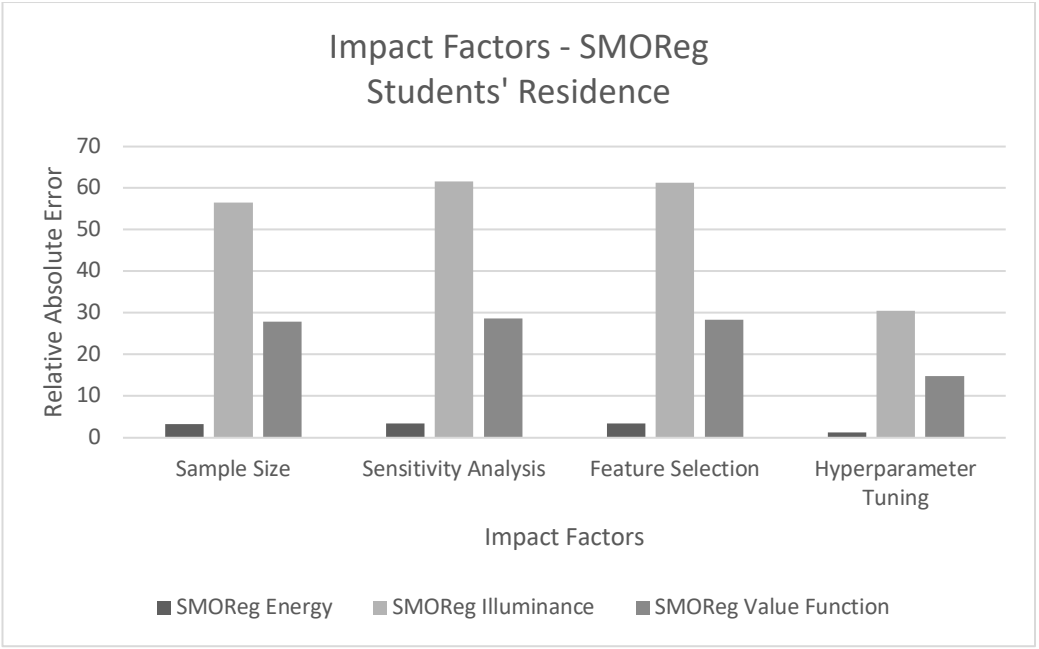


Figure 8:26 Students' Residence impact factors for SMOReg algorithm.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
SMOReg	**Energy**	3.31	3.46	3.43	1.27
	Illuminance	27.27	61.67	61.36	30.48
	Value Function	27.87	28.61	28.42	14.77

Table 8:17 Students' Residence impact factors for SMOReg algorithm.

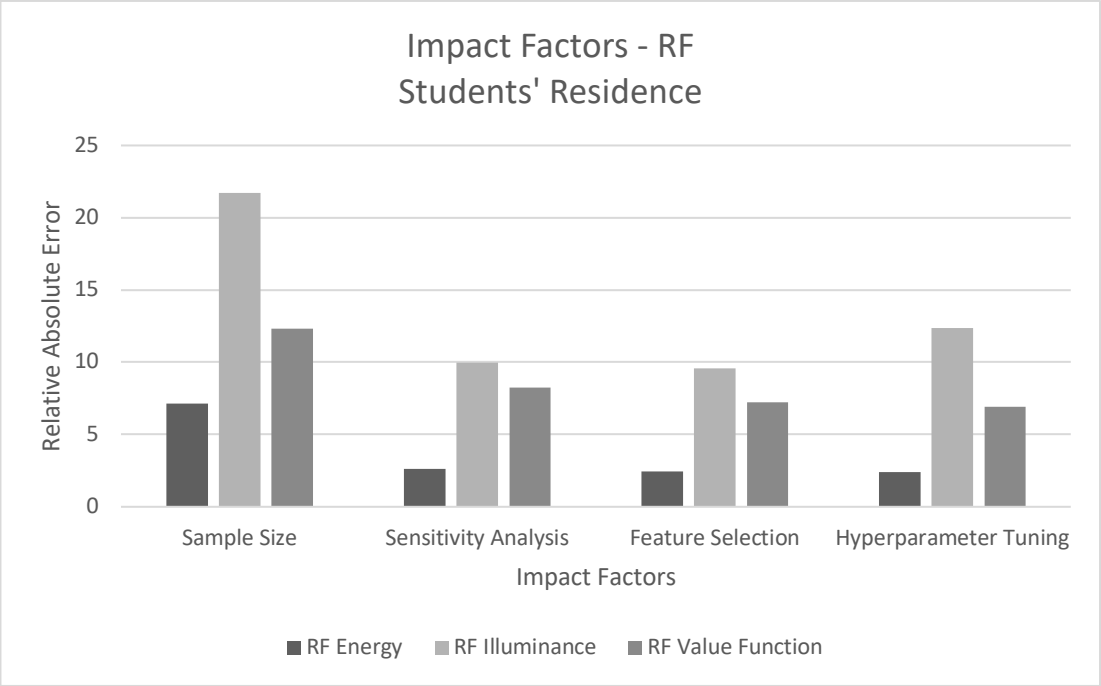


Figure 8:27 Students' Residence impact factors for Random Forest algorithm.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
RF	**Energy**	7.14	2.63	3.43	2.39
	Illuminance	18.23	9.96	61.36	12.37
	Value Function	12.33	8.24	28.42	6.9

Table 8:18 Students' Residence impact factors for Random Forest algorithm.

Chapter 9 Findings and Conclusion

In this chapter we begin by presenting answers to the research questions of the study. We then present a verification of the findings by examining a reference project. We discuss the impact of machine learning for design space reduction in the context of parametric analysis and make general concluding statements. Lastly, we discuss how to extend this work using the idea of data reuse and provide a summary of the study.

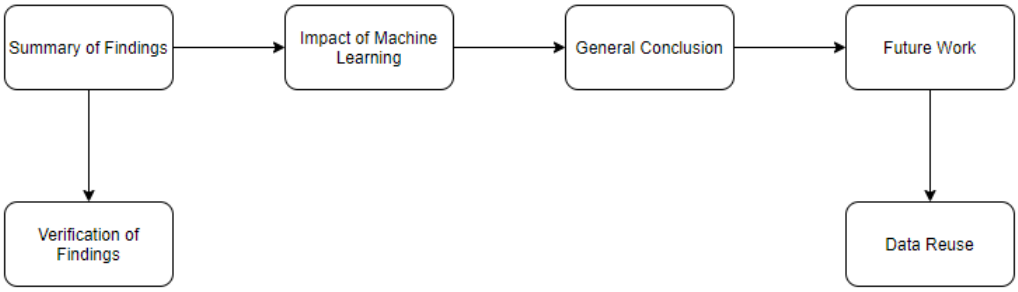


Figure 9:1 Diagram Summary of Chapter Nine.

9.1 Summary of Findings

The problem statement of this study asked one key question and three sub-questions. The key question was, in general, can machine learning algorithms be effective at multi-discipline parametric analysis? The three sub-questions were which machine learning algorithms are the most effective, what is the effect of impact factors on algorithm performance, and how does design space complexity affect the performance of algorithms?

9.1.1 Machine Learning Effectiveness

In response to the question of how effective machine learning algorithms are at multi-discipline parametric analysis we review the effectiveness of the algorithms on the three cases study projects. By effective we mean that the algorithm scored 5-10% RAE on one of the experimental samples. All samples had 1000 instances or fewer. The default had 1000 instances and out of the box settings for the algorithms. In a few cases the algorithm was not effective by default but became effective under one of the impact factors.

On the Residential Tower reference project MLP was effective for Energy and Illuminance by default. SMOReg and RF were not effective by default. Under sensitivity analysis MLP was effective for the Energy objective, SMOReg and RF were not effective. Under feature selection MLP was effective for all objectives while SMOReg and RF were not effective for any objective. Under hyperparameter tuning MLP was again effective for all objectives but SMOReg and RF were not effective for any objective.

On the Mid-Rise Towers MLP was effective for Energy under sample size considerations as well as feature selection and hyperparameter tuning. SMOReg was effective for the Energy objective under hyperparameter tuning. None of the algorithms was effective for Illuminance and the Value Function. In addition, RF was not effective for any objective under any of the impact factors.

On the Students' Residence MLP was effective for the Energy objective under all impact factors. It was also effective for the Value Function under sensitivity analysis and hyperparameter tuning. SMOReg was also effective for the Energy objective under all impact factors. It was not however effective for any other objective. RF was effective for the Energy objective under all impact factors. It was also effective for all objectives under feature selection. It was effective for The Value Function objective under hyperparameter tuning.

		Sample Size			Sensitivity Analysis			Feature Selection			Hyperparameter Tuning		
		MLP	SMOReg	RF	MLP	SMOReg	RF	MLP	SMOReg	RF	MLP	SMOReg	RF
Residential Tower	Energy	●			●			●			●		
	Illuminance	●						●			●		
	Value Function							●			●		
Mid-Rise Towers	Energy	●						●			●	●	
	Illuminance												
	Value Function												
Students' Residence	Energy	●	●	●	●	●	●	●	●	●	●	●	●
	Illuminance	●								●			
	Value Function	●			●					●	●		●

Table 9:1 Effectiveness of machine learning algorithms.

The analysis indicates that machine learning algorithms can be effective for multi-discipline parametric analysis. All samples satisfied the criteria for rapid simulation, about 33 hours, using generally accessible high specification hardware. In addition, all reference projects had at least one algorithm that was effective for at least one objective. In two of the reference projects at least one algorithm was effective for every objective.

Even where algorithms were not effective, as defined in this study, there was always at least one algorithm that was close to effective and several that were at least somewhat effective. For example, in the case of the Mid-Rise Tower's Illuminance objective RF had a lowest RAE of 12.82%. In the case of the Mid-Rise Tower's Value Function objective MLP had a lowest RAE of 13.18%. This suggests that with further optimization these algorithms can meet our definition of effectiveness. Moreover, the use of stronger algorithm implementations (such as deep learning algorithms) may well result in high performing defaults that meet the effectiveness criteria.

9.1.2 Best Performing Algorithm

In order to establish the best performing algorithms, we create a consolidated dataset from all the algorithms running on all objectives in the range 100 – 1000 instances (essentially the results from the sample size experiments). We do this across all three reference projects. The dataset has 270 instances and 4 fields: *Project ID, Objective, Algorithm, RAE* (Table 9:2). Since all the algorithms were run on the same data using the same method this consolidated dataset can be used to compare both the algorithm performance as well as the reference project performance. This latter will be useful for the design space complexity discussion (Section 9.1.4).

Project ID	Objective	Algorithm	RAE
RT	E	MLP	11.34
MRT	V	SMOReg	7.62
SR	I	RF	6.92
…	…	…	…

Table 9:2 Data format of consolidated dataset.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	8062.191	2	4031.096	32.189	0.000
Within Groups	33436.599	267	125.231		
Total	41498.790	269			

Table 9:3 Best performing algorithm one-way ANOVA.

(I) ALG		Mean Difference (I-J)	Std. Error	Sig.	90% Confidence Interval	
					Lower Bound	Upper Bound
MLP	RF	-9.79200*	1.66820	0.000	-13.2304	-6.3536
	SMOReg	-12.79900*	1.66820	0.000	-16.2374	-9.3606
RF	MLP	9.79200*	1.66820	0.000	6.3536	13.2304
	SMOReg	-3.00700	1.66820	0.171	-6.4454	0.4314
SMOReg	MLP	12.79900*	1.66820	0.000	9.3606	16.2374
	RF	3.00700	1.66820	0.171	-0.4314	6.4454

Table 9:4 Best performing algorithm Tukey HSD post processing.

Table 9:3 shows that at 90% confidence levels there are significant differences between the algorithm performances (*p<0.000)*. In addition the Tukey HSD post processing analysis (Table 9:4) shows that MLP performs significantly better than RF (9.79% better, *p<0.000*) and significantly better than SMOReg (12.5% better, *p<0.000*). There is no significant difference between the performances of RF and SMOReg. We conclude that MLP is the best performing algorithm.

9.1.3 Impact Factors

Impact factors are intended as a means to improve the machine learning algorithms beyond their default performance. In the Discussion sections of Chapters 6 – 8 we have presented a summary of the effect of all impact factors on each algorithm for each reference project. To understand the overall effect of impact factors for each algorithm we take an average of the results from the three chapters. For this purpose, we take the sample size results as a default against which to measure improvement.

For MLP (Figure 9:2) the most effective impact factor for all objectives was hyperparameter tuning. For the Energy objective hyperparameter tuning improved performance by 1.28%. For the Illuminance objective hyperparameter tuning improved performance by 0.53%. For the Value Function objective hyperparameter tuning improved performance by 1.09%. Sensitivity Analysis did not improve MLP performance. Feature Selection improved the Energy objective by 0.18%.

For SMOReg (Figure 9:3) the most effective impact factor for all objectives was hyperparameter tuning. For the Energy objective hyperparameter tuning improved performance by 3.03%. For the Illuminance objective hyperparameter tuning improved performance by 16.43%. For the Value Function objective hyperparameter tuning improved performance by 10.42%. Sensitivity Analysis and feature selection did not improve SMOReg performance.

For RF (Figure 9:4) feature selection was, on average, the most effective impact factor improving performance by 3.60%. Sensitivity Analysis improved performance by 3.46% while hyperparameter tuning improved performance by 2.93%. Sensitivity Analysis improved the performance of Energy, Illuminance and the Value Function by 2.44%, 6.05% and 1.90% respectively. Feature Selection improved the performance of Energy, Illuminance and the Value Function by 2.40%, 6.34% and 2.04% respectively. Hyperparameter Tuning improved the performance of Energy, Illuminance and the Value Function by 2.57%, 4.78% and 2.93% respectively.

We conclude that hyperparameter tuning is the most significant impact factor affecting algorithm performance (Figure 9:5). Hyperparameter Tuning, overall on average, improves performance by 4.62 better than the default. Sensitivity Analysis and Feature Selection both perform at about the same as the default improving performance by 0.6%.

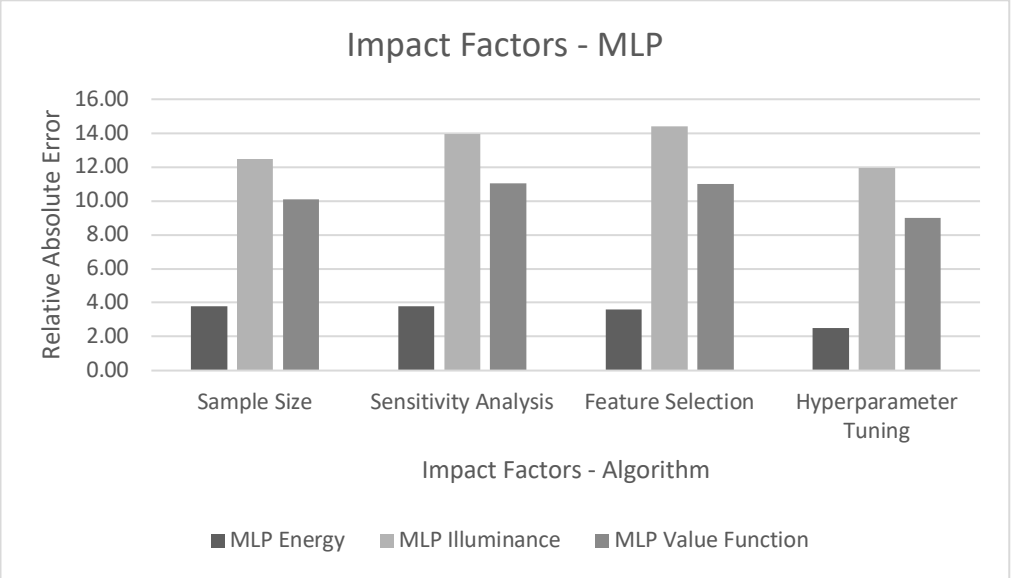


Figure 9:2 Effect of impact factors on MLP.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
MLP	**Energy**	3.79	3.80	3.61	2.51
	Illuminance	12.48	13.94	14.41	11.95
	Value Function	10.11	11.02	11.02	9.02

Table 9:5 Effect of impact factors on MLP

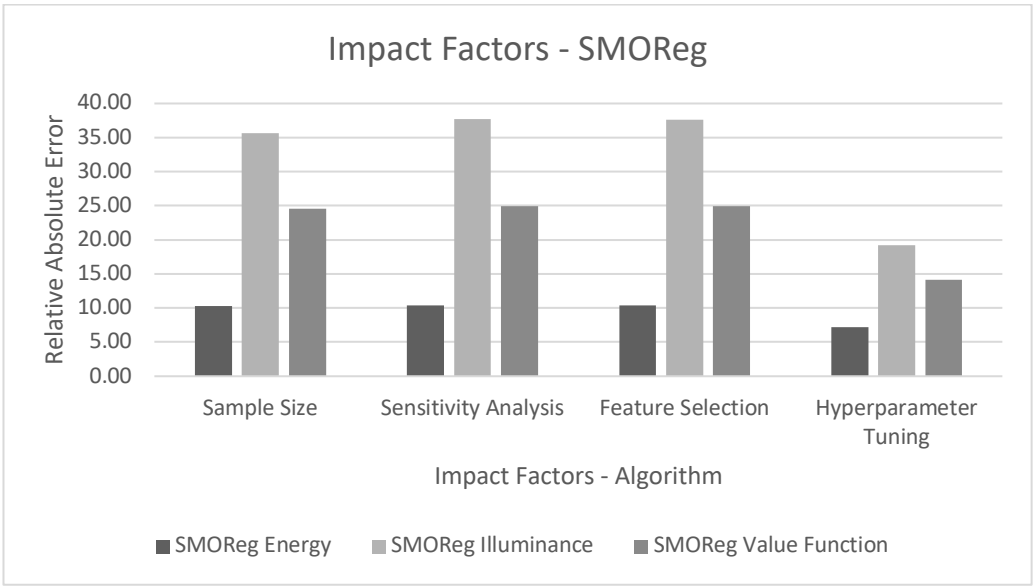


Figure 9:3 Effect of impact factors on SMOReg.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
SMOReg	**Energy**	10.27	10.36	10.36	7.24
	Illuminance	35.62	37.67	37.59	19.19
	Value Function	24.52	24.94	24.92	14.10

Table 9:6 Effect of impact factors on SMOReg.

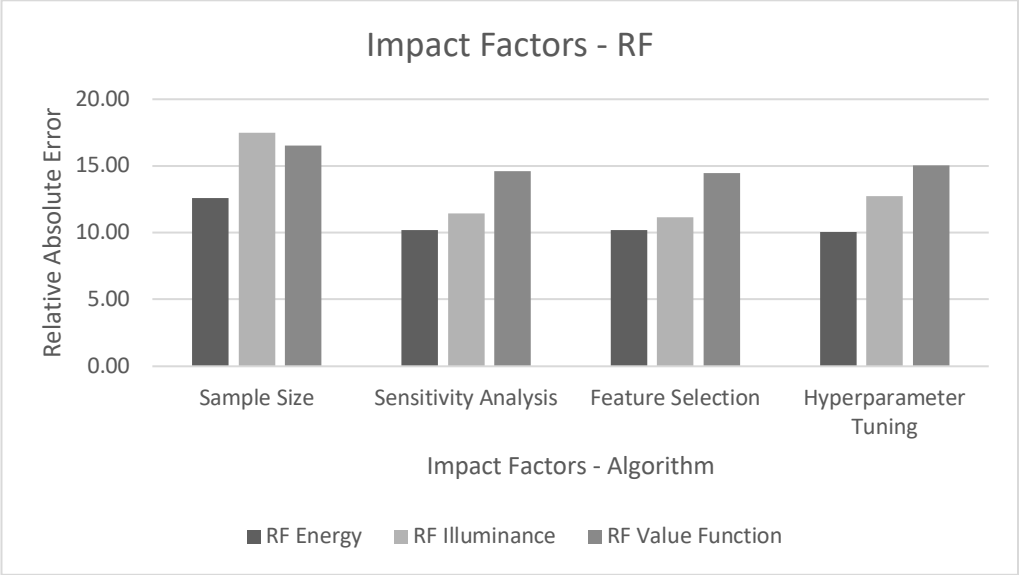


Figure 9:4 Effect of impact factors on RF.

		Sample Size	Sensitivity Analysis	Feature Selection	Hyperparameter Tuning
RF	**Energy**	12.62	10.18	10.22	10.05
	Illuminance	17.51	11.46	11.16	12.73
	Value Function	16.51	14.61	14.47	15.06

Table 9:7 Effect of impact factors on SMOReg.

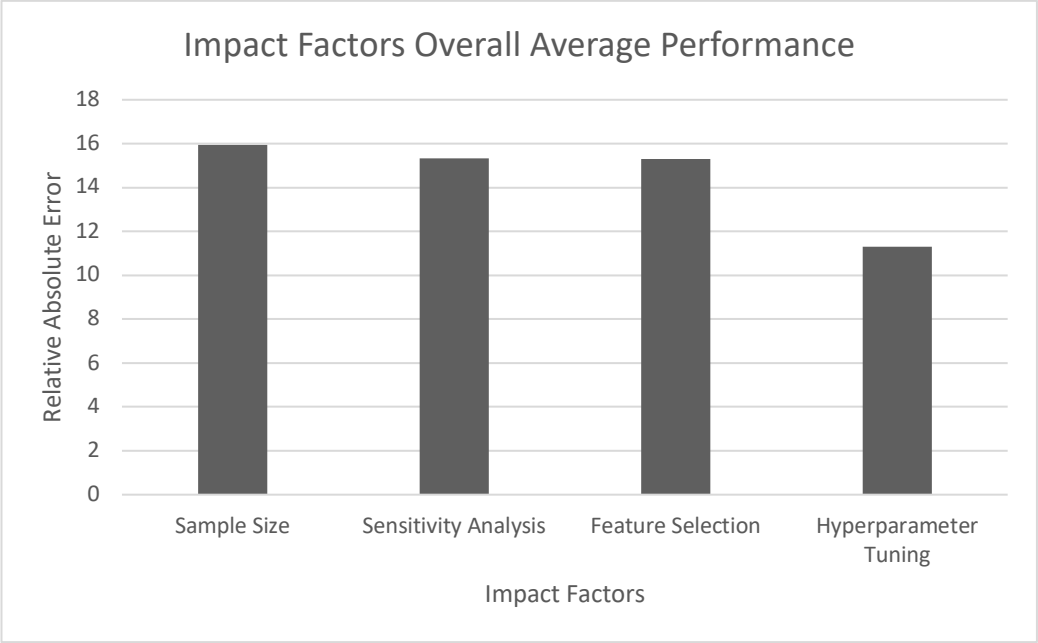


Figure 9:5 Overall average performance of impact factors.

9.1.4 Design Space Complexity

From Table 9:1 it is evident that machine learning algorithms can be effective on the reference projects with complex design space conditions (MRT and SR). In order to assess whether design space complexity has a significant impact on algorithm performance we perform one-way ANOVA on the consolidated data set introduced in Section 9.1.2. This will detect any significant difference between the accuracies of the reference projects. In addition, we will use Tukey HSD post processing to establish if any of the reference projects has a significantly better performing RAE score than any other.

If we do not detect a significant difference between the reference projects, we can infer that design space complexity, in these cases, does not impact the performance of the machine learning algorithms. If, however, we can establish a significant difference between the baseline reference project (RT) and both the other reference projects then it is possible that design space complexity is impacting algorithm performance.

Table 9:8 shows that at 90% confidence levels there are no significant differences between the reference projects (*p=0.165*). In addition, Tukey HSD post processing (Table 9:9) shows that there are no significant differences between Residential Tower, the baseline, and any of the other reference projects. It appears that in this case design space complexity has no significant impact on algorithm performance. This means that machine learning algorithms can be robust under complex design space conditions. It should be noted that this is not the same as saying that machine learning algorithms are *always* robust under complex design space conditions.

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	556.043	2	278.021	1.813	0.165
Within Groups	40942.748	267	153.344		
Total	41498.790	269			

Table 9:8 Design complexity one-way ANOVA.

(I) CID		Mean Difference (I-J)	Std. Error	Sig.	90% Confidence Interval	
					Lower Bound	Upper Bound
MT	RT	-1.91500	1.84598	0.554	-5.7199	1.8899
	SR	-3.51033	1.84598	0.140	-7.3152	0.2945
RT	MT	1.91500	1.84598	0.554	-1.8899	5.7199
	SR	-1.59533	1.84598	0.663	-5.4002	2.2095
SR	MT	3.51033	1.84598	0.140	-0.2945	7.3152
	RT	1.59533	1.84598	0.663	-2.2095	5.4002

Table 9:9 Design complexity Tukey HSD post processing.

One way to understand this result, which may seem surprising, is to examine the Energy objective scatter plot graphs for the two reference projects MRT and SR (Figure 9:6). For the SR project the *Option ID* parameter creates a strong correlation with the Energy output. This correlation improves the performance of the machine learning algorithms. In the case of the MRT project, the correlation is not as strong, but we note that the project has a very low sampling ratio (1:10) which improves algorithm performance.

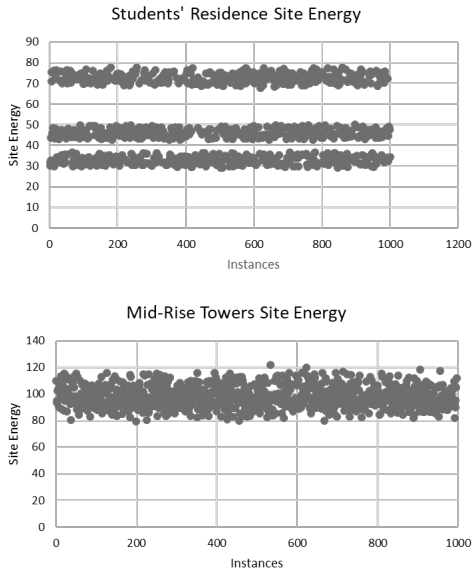


Figure 9:6 Energy scatterplots for the SR and MRT reference projects.

9.2 Verification of Findings

The findings of this thesis are based on real projects performed under specific project conditions. These conditions include project type, formal type, location, climate, design assumptions and compliance requirements. The project type is residential, both commercial residential and institutional residential projects; the formal types are low to mid-rise towers and a bar building type; the location is the lower mainland of British Columbia (BC); the climate is Pacific Northwest maritime; the design assumptions include the use of Vancouver 2016 CWEC weather files (Government of Canada, 2019), and the compliance requirements are based on BC building performance codes and standards. The verification is therefore done by performing machine learning on a comparable parametric analysis exercise and comparing the results with our findings.



Figure 9:7 (Perkins&Will, 2019) Commercial Residential Towers project.

For the purpose of this verification we will introduce the *Commercial Residential Towers* (CRT) (Figure 9:7). The project is a three-tower commercial-residential project located in Vancouver BC at the intersection of two major arterial routes. It is adjacent to one of the busiest transit hubs in BC. The project seeks to introduce a transit-oriented typology with a focus on family housing.

The project comprises of three residential towers above a commercial podium which covers the ground plane and encompasses a public realm plaza. It seeks to address the community needs for housing, childcare, a public community space and an engagement with the busy arterial routes and transit hub to which it is adjacent. As such, the programmatic mix comprises family-oriented units, market condo and rental units and new commercial and retail spaces.

In engaging the parametric analysis process the design team wished to understand the balance between energy performance and adequate daylight at the different levels and occupancies of the project. They also wished to understand the contextual effect on each tower of the neighboring towers. Additional analyses that were performed included a compactness analysis, solar access at the public plaza level and solar radiation on the different facades. We focus on the energy and daylighting analyses.

The design team elected to perform an initial analysis on the tallest tower to gain an understanding of the input parameters and parameter ranges that would be most effective for the total project analysis. The project clearly exhibits design space complexity. Given the different mix of occupancies and the geometric variations of the design it was decided to divide the analysis into seven slices (Figure 9:8). Two of the slices had identical footprints. The total building energy performance would be an area weighted aggregate of the performance of the seven slices. Daylight would be assessed on a level by level basis. The energy metric was the EUI of each slice while the daylight metric was the daylight factor.

The data description of the project is shown in Table 9:10. The design space description is shown in Table 9:11. 64 samples were selected from the design space for each of the slices giving a total of 448 simulation runs.

Input	**Data Type**	**Units**	**Description**
WWR_N	Numerical - Discrete	%	Window to wall ratio on the North façade
WWR_W	Numerical - Discrete	%	Window to wall ratio on the West façade
WWR_S	Numerical - Discrete	%	Window to wall ratio on the South façade
WWR_E	Numerical - Discrete	%	Window to wall ratio on the East façade
Wall U Value	Numerical - Discrete	$W/m^2 K$	Wall thermal transmittance value
Roof U Value	Numerical - Discrete	$W/m^2 K$	Roof thermal transmittance value
Exposed Floor U Value	Numerical - Discrete	$W/m^2 K$	Exposed Floor thermal transmittance value
Window U Value	Numerical - Discrete	$W/m^2 K$	Window thermal transmittance value
SHGC	Numerical - Discrete	-	Solar Heat Gain Coefficient
Slice ID			Identification of the floor to be analyzed
Output	**Data Type**	**Units**	**Description**
Floor Cooling	Numerical - Continuous	kWh/m^2	Floor Cooling load
Floor Heating	Numerical - Continuous	kWh/m^2	Floor Heating load
Floor Site Energy	Numerical - Continuous	kWh/m^2	Floor Total Site Energy
Floor Daylight Factor	Numerical - Continuous	%	Floor daylight factor

Table 9:10 Commercial Residence Towers data description.

Parameter	Description	Range	Size
WWR North	Window to wall ratio	45,50,55,60	4
WWR West	Window to wall ratio	45,50,55,60	4
WWR South	Window to wall ratio	45,50,55,60	4
WWR East	Window to wall ratio	45,50,55,60	4
Wall U Value	Wall U Value	1.76,2.64,3.52	3
Roof U Value	Roof U Value	2.64,3.52,5.28	3
Exposed Floor U Value	Floor U value	1.76,2.64,3.52	3
Window U Value	Shading Depth	1.7,1.94,2.56	3
SHGC	Wall U Value	0.2,0.3,0.4	3
Slice ID	Floor Identifier	1 - 7	7
			435,456

Table 9:11 Commercial Residence Towers design space description.

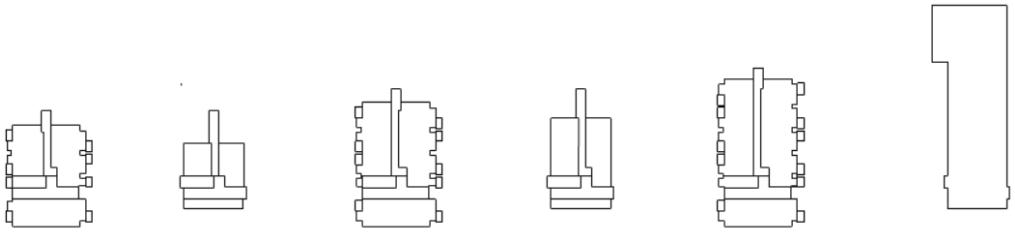


Figure 9:8 (Perkins&Will, 2019) CRT project, analysis slices.

The findings were that the best energy performance occurred at the lower levels 1 and 2 (Figure 9:9), the best daylight performance happened at the upper levels (Figure 9:10) and the best-balanced performance happened at level 6 near the top (Figure 9:11). This information was useful for deciding on the types of glazing and envelope insulation to use at different levels of the building.

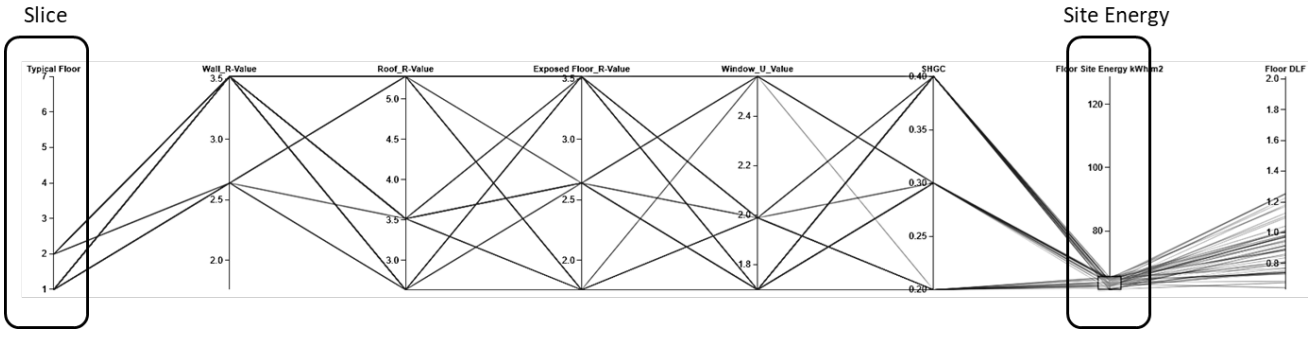


Figure 9:9 (Perkins&Will, 2019) CRT project, best energy performing levels.

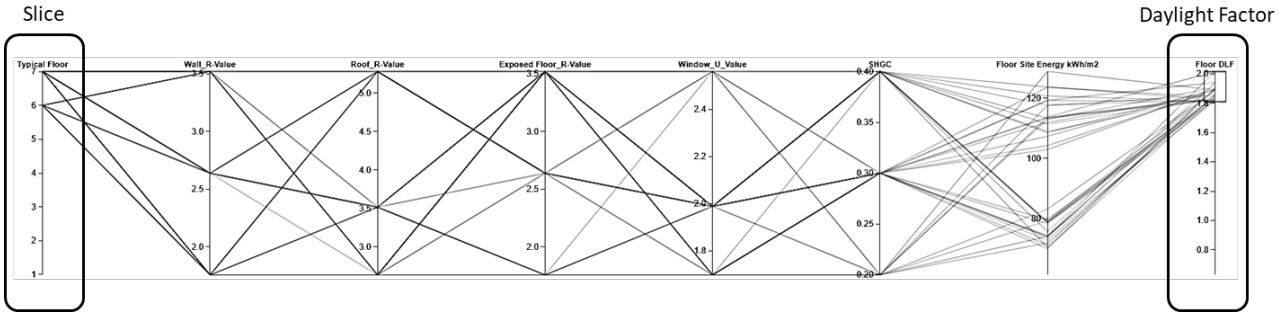


Figure 9:10 (Perkins&Will, 2019) CRT project, best daylight factor performing levels.

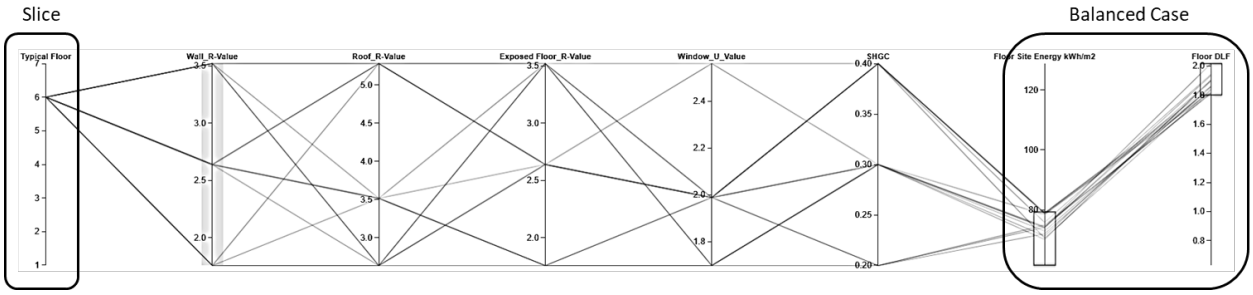


Figure 9:11 (Perkins&Will, 2019) CRT project, best balanced case.

The question we seek to verify is whether machine learning algorithms can be effective at predicting this project's design space. In particular, can an algorithm be effective given the design space complexity of the project? We note that CRT only performed 438 simulations. However, since 438 is less than our 1000 instance benchmark, if an algorithm is successful then the question is answered in the affirmative. In addition, we note that the project chose to use the daylight factor metric which is simpler to compute than Illuminance based metrics and should be easier for an algorithm to learn.

From the 438-instance dataset we randomly extracted a 400-instance training set. Using the process of Section 5.5.3 we calculated the value function. We then ran all three algorithms on all three objectives using their default settings and ten-fold cross-validation. The results are seen in Table 9:12.

ALGORITHM	ENERGY	DAYLIGHT	VALUE FUNCTION
MLP	7.4	12.99	10.69
SMOReg	32.1	15.88	43.84
RF	8.86	21.91	20.4

Table 9:12 CRT project results of verification runs.

MLP and RF are effective for the Energy objective. MLP is close to effective for the Daylight objective while SMOReg is somewhat effective. MLP is close to effective for the Value Function objective. Machine learning algorithms, and in particular MLP, are effective for Energy and close to effective for Daylight and the Value Function. This suggests that with more data (up to 1000 instances) and with the use of impact factors the algorithms can be effective for all objectives.

9.3 Impact of Machine Learning on Design Space Reduction

We have described how parametric analysis improves on conventional analysis by providing capabilities like the ability to perform optimization, sensitivity analysis and uncertainty analysis. However, we have also acknowledged the data analytical challenges of parametric analysis associated with questions of design space size and design space complexity. We have noted that approaches to resolving these

challenges can be hardware-based, algorithmic or design space reduction. In this study we have focused on design space reduction.

Compared to algorithmic methods like genetic algorithms, machine learning algorithms model uncertainty. Given a large design space a genetic algorithm can perform a search and identify optimum values. However, we do not know the error between the algorithmic optimum value and the actual optimum value. This undermines our confidence in making use of the result. Machine learning algorithms on the other hand model uncertainty and can report on a range of uncertainty metrics. The regression-based machine learning algorithms in Weka, for example, report the correlation coefficient, mean absolute error, root mean squared error, relative absolute error and the root relative squared error.

Design space reduction methods, by definition, involve sampling the design space. While sampling methods can report accuracy, and can achieve high accuracies in their own right, unlike predictive methods they do not supply a full design space. This can impact the accuracy of additional analyses, like sensitivity analysis, as well as impeding thorough design exploration. For example, the DOE heuristic method we described in Sections 2.3.4 and 6.2.5 had a high-performance bias. Therefore, the samples produced include preferred alternatives and avoid poorly performing alternatives. However, it can be quite useful to understand what input parameter values contribute to poor performance. Designers can avoid such values or else, by understanding their underlying causes, better understand the design.

Another advantage of machine learning over algorithmic or sampling methods is the ability to interpolate between input parameter values. If the input parameters are numeric variables, then the machine learning algorithm can make predictions based on interpolated inputs. For example, on the Residential Tower the design space specified a balcony depth of 0', 5' and 10'. However, one of the designers became interested in the effect of using a 7.5' balcony depth. This can enable more nuanced optimization beyond the initial assumptions. With machine learning it is possible to use the data model to make this prediction by simply entering this intermediate value in the prediction model. This saves potentially time-consuming iterations of simulation.

Another benefit of a full design space is the ability to perform robust design space exploration. For example, on the Residential Tower project the designers wished to use WWR 50% while the building envelope consultant had recommended a value closer to WWR 40%. While it is possible to perform a single-run analysis and establish that WWR 50% is indeed feasible this will not be robust. It tells us nothing about how a WWR 50% will perform if other parameters fluctuate as they invariably will in reality. With a full design space, we can assess the robustness of WWR 50% by, for example, doing a one sample t-test at WWR 50% with a specified target mean. If there is no significant difference between the target mean and the mean at WWR 50% then we can confidently infer that the latter will be robust in practice.

Analysis of the impact of sample size on algorithm performance reveals that in many cases machine learning algorithms can perform better than the prescribed targets (sample size = 1000; accuracy = 5 - 10% RAE; simulation speed = 33.33 hours). For example, for the Residential Tower Energy objective, it was shown that 200 simulations resulted in 7.62% RAE which Is effective. This means that in practice the simulation speed can be reduced from 33.33 hours to 6.77 hours. The faster parametric analyses can be performed the better for the design process.

Sampling methods also run a risk of reporting localized optima. If the design space is represented by complex high dimensional hypersurfaces, there is always the risk that working with anything other than the full design space will report localized maximum and minimum values, missing the true optimum. Predictive models alleviate this concern by ensuring we are working with the full design space.

Ultimately, since parametric analysis is a data-driven method it presents the opportunity for long term data mining. If complete parametric analysis results are stored in a suitable data repository this can become a valuable database for data mining. For example, it is possible to build a recommender system that suggests

parameter value ranges for different project types when generating new design spaces. This can cut down on research effort and improve the quality and accuracy of parametric analyses. It can also become possible to make simulation predictions for different project types directly from a predictive model derived from the database. This eliminates the need for time consuming sample simulations when developing predictive models.

9.4 Conclusion

This study began as a quest to understand if machine learning algorithms can be effective in solving the challenge of high cardinality design spaces in architectural practice. Based on observations of comparable large architectural firms few, if any, have reported successful implementations of multi-discipline parametric analysis in their design process. Success in implementing parametric analysis has been reported from the world of building performance engineering: Integral Group (Integral Group, 2019) have been using a proprietary tool called Integral Drive™ and Morrison Hershfield have developed PathFinder™ (Morrison Hershfield, 2019). However, both are primarily energy analysis tools and are not multi-discipline in the sense we have discussed.

Current attempts to implement parametric analysis in architectural practice, such as by using the DSC methodology, make use of compromises and approximations. As an example, designers have proposed using static, formula-based, calculators like the Energy Performance Calculator or EPC (Georgia Tech, 2019) and the Passive House Planning Package or PHPP (Passive House Insitute, 2019) as opposed to dynamic simulators like EnergyPlus. As another example designers have proposed using simpler metrics for parametric analysis like the daylight factor instead of more time-consuming Illuminance based metrics. Both of these proposals, while useful within context, are essentially a compromise of simulation accuracy for simulation speed.

Another approximation often encountered is the use of sampling methods to reduce the design space to be simulated. Recommendations are then made from estimates based on the sampled results. However, many of these sampling methods do not quantify the associated uncertainty. For example, in the three reference projects we have examined the parametric analysis process did not report the error associated with the results. Instead designers relied on the judgement of a building performance expert to validate estimations based on sampling. This is a qualitative assessment that recalls the criticism of conventional analysis methods made in Section 2.1. Failure to quantify the uncertainty of data analytical methods undermines confidence levels in their output. Machine learning, as a design space reduction method, reports the accuracy associated with predictions. By proposing a method for assessing the effectiveness of machine learning algorithms this study seeks to contribute to the improved analytical rigor of parametric analysis.

As we have mentioned one possible approach to solving the high cardinality design space problem is to invest in a hardware solution. Welle et al. (2019), at Perkins&Will, have in fact taken this approach developing an interactive, web-based, multiple performance assessment tool called Speed™. Speed performs multi-discipline parametric analysis leveraging the power of Amazon Web Services to process large design spaces at comparatively high speeds. Welle has identified the challenges of hardware-based solutions for parametric analysis. These include costs, both of development and in use, integration challenges, functionality challenges and deployment and testing challenges. In this study we have similarly identified the challenges of solutions based on design space reduction. These include speed of simulation, accuracy of predictions, sampling ratio considerations and complex design space considerations.

The methodology used in this study can be seen as a contribution towards a more systematic process of assessing the effectiveness of machine learning for parametric analysis in practice. This can help designers assess the applicability of machine learning algorithms for their particular project types. This use of systematic process ties in with the research tradition of design space exploration. Thus Clevenger & Haymaker (2013) proposed a systematic process for evaluating the quality of design exploration. Haymaker et al. (2018) proposed a systematic process for integrating multi-discipline parametric analysis in

architectural practice. Likewise, the experimental methodology of this study can be developed into a systematic process for assessing if machine learning algorithms can be integrated into parametric analysis.

9.5 Future Work

One question that undermines the use of machine learning for building performance analysis is the fact that a new machine learning model is required for every new project. Even when projects are of similar type or have identical input parameters machine learning models cannot easily be transferred from one project to the next. Thus, if a machine learning model is trained on data from a K-12 school project this model cannot be used to make predictions on a residential project. Worse, even if a model is trained on a residential project it cannot easily be used to make accurate predictions on a different residential project.

We demonstrate this data reuse problem by performing a machine learning experiment on two mixed-use commercial-residential projects located in Vancouver, BC. The projects are the Residential Tower and the Mid-Rise Tower project. We will train machine learning models using simulated data from the Residential Tower and then use it to make predictions on both the Residential Tower and the Mid-Rise Tower dataset. We will then compare the errors from the two predictions and show that it is much greater for the heterogeneous case (Mid-Rise Tower prediction) than the homogenous case (Residential Tower prediction).

The steps of the experiment are:

1. Create 200 random samples from the Residential Tower design space and perform energy analysis obtaining the Site Energy as a metric. Create 100 random samples from the Mid-Rise Towers project for comparison testing.

2. Use 100 instances of the Residential Tower dataset to create machine learning models using ANN, SVM and RF algorithms.

3. Use the Residential Tower machine learning models to make predictions on the Residential Tower holdout data and the Mid-Rise Towers sample data. Calculate the prediction error of using the Residential Tower model on Residential Tower data compared to using the Residential Tower model on Mid-Rise Towers data.

	ANN	SVM	RF
Residential Tower	0.54%	0.94%	1.52%
Mid-Rise Towers	19.72%	14.01%	22.51%

Table 9:13 RAE for the RT and MRT using the RT training set.

The results are shown in Table 9:13. The RAE is quite low when the test set and the training set are from the same project. However, even with identical input parameters, the error is significantly higher when the test set is from a different project. The error from the heterogeneous dataset is in fact 22 times larger than the error from the homogenous dataset.

To address the challenge of data reuse Geyer & Singaravel (2018) proposed a component based approach to machine learning where the algorithms are trained, not on the whole building but rather on components of the building. Since the components of buildings are reusable then the approach facilitates data reuse on a wide range of design situations where these components are utilized. For example, when training a whole building the input parameters could be *building type*, *wall u-value*, *floor u-value*, *roof u-value* and glazing *solar heat gain coefficient*. If faced with a different building of a different type in a different climate location, a design space reduction model based on these inputs will not be accurate. However, if we train a building component, such as a building zone, the input parameters could be *area*, *air changes per hour*, *elevation of the zone, heat radiation* and *heat transfer*. The whole building performance is an aggregation of its zonal performances. But the zone is easier to train and more transferable to other building designs since it is a small element that is easy to configure for different design requirements.

As a follow up to this study, and after the fashion of Geyer & Singaravel, we propose a transferable zone-based machine learning process for parametric analysis (Figure 9:12). For a given project type this will have the advantage of allowing predictions for different projects using the same generic zone object. Predictions will be made at the zone level and then aggregated for the whole building analysis (Figure 9:13). The process would be as follows:

1. Select a building type from the ASHRAE/IES Standard 90.1 Prototype Building Models (US Department of Energy, 2019).
2. Create a generic zone object for this project type. This generic object will include parametric geometry, parametric inputs and parametric adjacency relationships (Figure 9:13).
3. Create a design space based on the generic zone's geometry, parameters and adjacencies. This design space will likely be massive to encompass a broad range of possible projects scenarios. However, it only needs to be simulated once to create the machine learning model. Moreover, the process probably justifies hardware investment that will make it much faster.
4. Create a machine learning model from the simulated design space.
5. Use the machine learning model to predict zone performance on new projects.
6. Aggregate the predicted zones to obtain whole building performance.

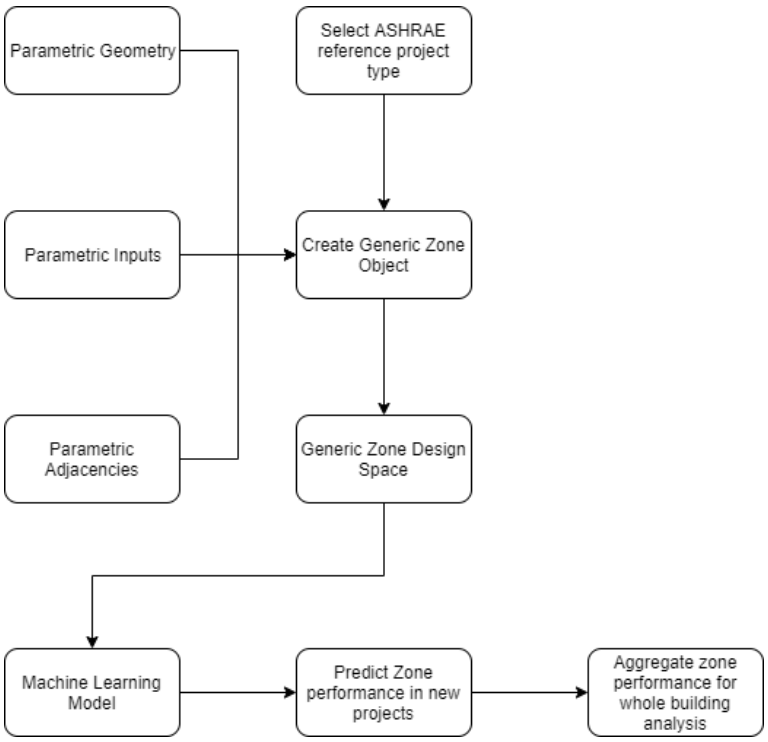


Figure 9:12 Future work. Proposed zone-based machine learning process.

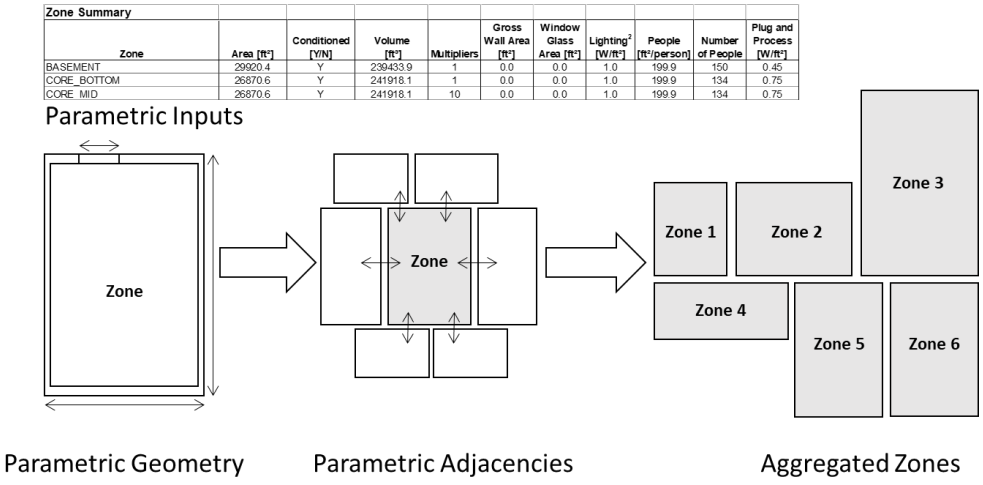


Figure 9:13 Future work. Transferable zone-based machine learning model.

We finish by reviewing the ground we have covered in this study. We investigated the effectiveness of machine learning algorithms for multi-discipline parametric analysis. The motivation for the study was a recognition of the potential benefits of parametric analysis. However, attempting to implement parametric analysis in practice gave rise to challenges some of which were data analytical in nature. The study investigated resolving these data analytical challenges by using machine learning algorithms.

The study was situated in research that investigated the concepts of design spaces and design space exploration. Design spaces are defined as the cross product of parameter value spaces while design space exploration is defined as the mapping of design spaces to design solutions. These concepts give rise to theoretical questions that have, as a matter of course, been addressed by introducing methodological frameworks that rationalize design space exploration.

One such methodological framework is DSC which develops a systematic process for parametric analysis. It includes process definitions for establishing analysis objectives, generating alternatives, evaluating impacts and assessing value. The methodology was developed and implemented at Perkins&Will as part of research in practice, and the author was a member of the implementing team of researchers. This study was based, in large part, on this work.

In formulating the research questions of the study, we first discussed the limitations of conventional building performance analysis. In contrast we found that parametric analysis provides capabilities that conventional analysis does not. These capabilities include the ability to perform optimization analysis, sensitivity analysis and uncertainty analysis. However, implementing parametric analysis gives rise to data analytical challenges associated with the idea of high cardinality design spaces. Design space reduction is one approach for dealing with large design spaces and machine learning is one approach to design space reduction.

It is worth emphasizing that the parametric analysis discussed in this study is multi-discipline building performance analysis. Architecture, unlike other construction disciplines which tend to specialize, is the profession most concerned with whole building analysis. The architect is tasked with coordinating all the different aspects of design and this requires holistic, broad based analytical approaches. Therefore, the architectural parametric analysis process invites the participation of multiple disciplines. This exacerbates the dynamic simulation problems and makes the mitigations of design space reduction more urgent.

The methodology of the study involved three reference projects from which design spaces were developed using the DSC framework. It also involved an integrated, multi-discipline simulation process that simulated both the objectives of energy and daylighting simultaneously. Lastly, it involved machine learning experiments to determine the effectiveness of machine learning algorithms and to identify the most influential impact factors affecting algorithm performance.

The metric of the experiments was the Relative Absolute Error (RAE) of the algorithm on a sample training set ranging in size from 100 to 1000 instances. Effective performance was defined as parametric analysis that was both fast and accurate. This meant machine learning algorithms were required to score less than 5 - 10% RAE from a sample size simulated in less than 33 hours on generally available high specification hardware. This translates to a sample size of about 1000 instances.

Three commonly used algorithms were investigated – ANN, SVM and DT. The machine learning platform of choice was Weka because it is more accessible to architectural use than other platforms like R or Python. Weka implementations of the three algorithms were applied namely MLP, SMOReg and RF. Comparative performance analysis of the algorithms for the different design objectives was undertaken and reported. The design objectives were energy, daylighting and the value function that is used for the multi-discipline (and multi-objective) comparison. In addition, the influence of impact factors (sample size, sensitivity analysis, feature selection and hyperparameter tuning) was investigated in order to identify strategies for improving algorithm performance.

The study found that machine learning algorithms can be effective at design space reduction for multi-discipline parametric analysis under real project constraints. The most effective algorithm was artificial neural networks while the impact factor influencing algorithm performance the most was hyperparameter tuning. The study also found no significant performance difference due to the design space complexity of the reference projects.

References

Ahmad M, Hippolyte , J., Mourshed, M., & Rezgui, Y. (2017). Random Forests and Artificial Neural Network for Predicting Daylight Illuminance and Energy Consumption. *Proceedings of the 15th IBPSA Conference*, (pp. 7-9).

Akin, O. (2006). The whittled design space. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 83-88.

Alpaydin, E. (2004). *Introduction ot Machine Learning.* MIT Press.

Amasyali, K., & El-Gohary, N. (2018). A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews*, 1192-1205.

Amazon. (2019). *Amazon Web Services.* Retrieved from https://aws.amazon.com/

ANSI/ASHRAE. (2017). *ANSI/ASHRAE Standard 140-2014 Space-Cooling Equipment Performance Comparative Tests CE300 to CE545.*

Ascione, F., Bianco, N., De Stasio, C., Mauro, G., & Vanoli, G. (2015). A new methodology for cost-optimal analysis by means of the multi-objective optimization of building energy performance. *Energy and Buildings*, 78-90.

Bergstra, J., & Bengio, Y. (2012). Random Search for hyper-parameter optimization. *Journal of Machine Learning Research*, 281-305.

Breiman, L. (2001). Random Forests. *Machine Learning*, 5-32.

Browntree, J. (2014). *Machine learning mastery.* Retrieved from http://machinelearningmastery. com/discover-feature-engineering-howtoengineer-features-and-how-to-getgood-at-it

Bryson, A., & Ho, Y. (1969). *Applied Optimal Control .* New York: Blaisdell.

Carlucci, S., Cattarin, G., Causone, F., & Pagliano, L. (2015). Multi-objective optimization of a nearly zero-energy building based on thermal and visual discomfort minimization using a non-dominated sorting genetic algorithm (NSGA-II). *Energy and Buildings*, 378-394.

Chachere, J., & Haymaker , J. (2011). Framework for measuring the rationale clarity of AEC design decisions. *Journal of Architectural Engineering,, 17*(3), 86-96.

Chlela, F., Husaunndee, A., Inard, C., & Reiderer, P. (2009). A New Methodology for the Design of Low Energy Buildings. *Energy and Buildings, 41(9)*, 982-990.

Cichosz, P. (2014). *Data Mining Algorithms explained using R.* John Wiley & Sons.

City of Surrey. (2019). *BC Energy Step Code for New Buildings*. Retrieved from BC Energy Step Code for New Buildings: https://www.surrey.ca/city-services/27975.aspx?utm_source=VanityURL&utm_medium=VanityURL&utm_campaign=StepCode

City of Vancouver. Planning, Urban Design Department. (2017). *Green Building Policy for Rezoning - Process and Requirements.* Vancouver: City of Vancouver.

Clevenger, C., & Haymaker, J. (2012). The Value of Design Strategies Applied to Energy Efficiency. *Smart and Sustainable Built Environment, 1*(3), 222-240.

Clevenger, C., & Haymaker, J. (2013). Design exploration assessment methodology: testing the guidance of design processes. *Journal of Engineering Design*, 165-184.

Coakley, D., Raftery, P., & Keane, M. (2014). A review of methods to match building energy simulation models to measured data . *Renewable and Sustainable Energy Reviews*, 123-141.

Corrado, V., & Mechri, H. (2009). Uncertainty and sensitivity analysis for building energy rating. *Journal of building physics, 33*(2), 125-156.

Darwin. (2004). *Genetic Algorithm Based Design Optimization Using ModelCenter and Darwin.* Phoenix Integration Inc.

Datacamp. (2019). *Datacamp.* Retrieved from https://www.datacamp.com/

De Wit, S., & Augenbroe, G. (2002). Analysis of uncertainty in building design evaluations and its implications. *Energy and Buildings*, 951-958.

Delgarm, N., Behrang, S., Khadijeh, A., & Delgarm, S. (2018). Sensitivity analysis of building energy performance: A simulation-based. *Journal of Building Engineering, 15*, 181-193.

Delgarm, N., Sajadi, B., Kowsary, F., & Delgarm, S. (2016). Multi-objective optimization of the building energy performance: A simulation-based approach by means of particle swarm optimization. *Applied Energy*, 293-303.

DesignBuilder. (2017). *ANSI/ASHRAE Standard 140-2014 Building Thermal Envelope and Fabric Load Tests.*

DesignBuilder. (2017). *ANSI/ASHRAE Standard 140-2014 Space Heating Equipment Performance Tests HE100 to HE230.*

DesignBuilder. (2017). *ANSI/ASHRAE Standard 140-2014 Space-Cooling Equipment Performance Analytical Verification Tests CE100 to CE200.*

Dhariwal, J., & Banerjee, R. (2017). An approach for building design optimization using design of experiments. *Building Simulation, 10(3)*, 323-336.

Diakaki, C., Evangelois, G., & Kolokotsa, D. (2008). Towards a multi-objective optimization approach for improving energy efficiency in buildings. *Energy and Buildings*, 1747-1754.

Dunn, K. (2017). *Process Improvement Using Data.* Retrieved from https://learnche.org/pid/

Echenagucia, T., Capozzoli, A., Cascone, Y., & Sassone , M. (2015). The early design stage of a building envelope: Multi-objective search through heating, cooling and lighting energy performance analysis. *Applied Energy*, 577-591.

Eguaras-Martinez, M., Vidaurre-Arbizu, M., & Martin-Gomez, C. (2014). Simulation and evaluation of Building Information Modeling in a real pilot site. *Applied Energy*, 475-484.

Evins, R. (2013). A review of computational optimization methods applied to sustainable building design. *Renewable and Sustainable Energy Reviews*, 230-245.

Fang, Y. (2017). *Optimization of Daylighting and Energy Performance Using Parametric Design, Simulation Modeling, and Genetic Algorithms.* Raleigh: North Carolina State University.

Flager, F., Welle, B., Bansal, P., Grant, S., & Haymaker, J. (2009). Multidisciplinary process integration and design optimization of a classroom building. *Journal of Information Technology in Construction (ITcon), 14*(38), 595-612.

Galanter, P. (2012). Computational Aesthetic Evaluation: Past and Future. In J. McCormack, & M. d'Inverno (Eds.). Springer.

Garg, V., Kshitij, C., Jyotirmay, M., Surekha, T., & Akshey, J. (2011). Development and performance evaluation of a methodology, based on distributed computing, for speeding EnergyPlus simulation. *Journal of Building Performance Simulation*, 257-270.

Georgia Tech. (2019). *EPC Integration.* Retrieved from https://imagine.gatech.edu/node/511

Geyer, P., & Singaravel, S. (2018). Component-based machine learning for performance prediction in building design. *Applied Energy*, 1439-1453.

Glover, F., & Greenberg, H. (1989). New approaches fro heuristic search: A bilateral linkage with artificial intelligence. *European Journal of Operational Research*, 119-130.

Government of Canada. (2019). *Engineering Climate Datasets.* Retrieved from https://climate.weather.gc.ca/prods_servs/engineering_e.html

Grasshopper. (2019). *Grasshopper{Software].* Retrieved from https://www.grasshopper3d.com/

Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. (Eds.). (2008). *Feature extraction: foundations and applications* (Vol. 207). Springer.

Hamdy, M., Nguyen, A.-T., & Hensen, J. (2016). A performance comparison of multi-objective optimization algorithms for solving nearly-zero-energy-building design problems. *Energy and Buildings*, 57-71.

Haymaker, J. (2006). Communicating, integrating and improving multidisciplinary design narratives. *Design Computing and Cognition'06*, 635-653.

Haymaker, J., Bernal, M., Marshall, T., Okhoya, V., Szilasi, A., Rezaee, R., . . . Welle, B. (2018). Design space construction: A framework to support collaborative, parametric decision making. *ITcon, 23*, 157-178.

Heden. (2016). *Predicting Hourly Residential Energy Consumption using Random Forest and Support Vector Regression: An Analysis of the Impact of Household Clustering on the Performance Accuracy.* Masters Thesis, Royal Institute of Technology.

Heiselberg, P., Hesselholt, A., Rasmussen, H., Seinre, E., & Thomas, S. (2009). Application of sensitivity analysis in design of sustainable buildings. *Renewable Energy, 34*(9), 2030-2036.

Hoes, P., Tricka, M., Hensen, J., & Bonnema, B. (2011). Optimizing building designs using a robustness indicator with respect to user behavior. *Building Simulation Proceedings of the 12th Conference of the International Building Performance Simulation Association*, (pp. 1710-1717).

Hygh, J., DeCarolis, J., Hill, D., & Ranjithan, S. (2012). Multivariate regression as an energy assessment tool in early building design. *Building and Environment*, 165-175.

IBM. (2019). *SPSS [Software].* Retrieved from https://www.ibm.com/products/spss-statistics

Integral Group. (2019). *Integral Group.* Retrieved from https://www.integralgroup.com/

James J. Hirsch & Associates. (2019). *DOE2.* Retrieved from http://www.doe2.com/

Jurecka, F. (2007). *Robust design optimization based on metamodeling techniques.* Doctoral dissertation, Technische Universität München.

Kazanasmaz, T., Gunaydin, M., & Binol, S. (2009). Artificial neural networks to predict daylight illuminance in office buildings. *Building and Environment*, 1751-1757.

Keogh, E., & Mueen, A. (2017). Curse of Dimensionality. In *Encyclopedia of Machine Learning and Data Mining* (pp. 314-315). Boston: Springer.

Krishnamurti, R. (2006). Explicit design space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 95-103.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling.* New York: Springer.

Lartigue, B., Lasternas, B., & Loftness, V. (2014). Multi-objective optimization of building envelope for energy consumption and daylight. *Indoor and built environment, 23*(1), 70-80.

Lippman, R. (1987). An Introduction' to Computing with Neural Nets. *IEEE Assp Magazine*, 4-22.

Liu, Y., Colburn, A., & Mehlika, I. (2018). Computing Long-Term Daylighting Simulations from High Dynamic Range Imagery using Deep Neural Networks. *2018 Building Performance Analysis Conference and SimBuild.*

Lorenz, C.-L., & Jabi, W. (2017). Predicting daylight autonomy metrics using machine learning. *Proceedings of the International Conference for Sustainable Design of the Built Environment (SDBE)*, (pp. 991-1002).

M, A., Hippolyte, J., Mourshed, M., & Rezgui, Y. (2017). Random Forests and Artificial Neural Network for Predicting Daylight Illuminance and Energy Consumption. *Proceedings of the 15th IBPSA Conference*, (pp. 7-9).

Machairas, V., Tsangrassoulis, A., & Axarli, K. (2014). Algorithms for optimization of building design: A review. *Renewable and Sustainable Energy Reviews*, 101-112.

Machine Learning, D. L. (2019). *Machine Learning Tutorial.* Retrieved from http://www.machinelearningtutorial.net/

Magnier, L., & Haghighat, F. (2010). Multiobjective optimization of building design using TRNSYS simulations, genetic algorithms and Artificial Neural Network. *Building and Environment, 45(3)*, 739-746.

Magoules, F., & Zhao, H.-X. (2016). *Data Mining and Machine Learning in Building Energy Analysis.* Wiley.

Mazzarella, L., Pasini, M., & Hoonejani, N. (2014). Challenges, limitations, and success of cloud computing for parallel simulation of multiple scenario and co-simulation. *ASHRAE/IBPSA-USA Building Simulation Conference*, (pp. 95-102).

Microsoft. (2019). *Microsoft Azure [Software].* Retrieved from https://azure.microsoft.com/en-us/

Microsoft. (2019). *MS Excel [Software].* Retrieved from https://www.microsoft.com/en-ca

Mitchell, T. (1997). *Machine Learning.* McGraw-Hill.

Morrison Hershfield. (2019). *Building PathFinder.* Retrieved from https://morrisonhershfield.com/bpa_library/building-pathfinder/

Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective.* MIT Press.

Murray, S., Walsh, B., Kelliher, D., & O'Sullivan, D. (2014). Multi-variable optimization of thermal energy efficiency retrofitting of buildings using static modelling and genetic algorithms e A case study. *Building and Environment*, 98-107.

Nguyen, A., Reiter, S., & Rigo, P. (2014). A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy*, 1043-1058.

Nielsen, M., Svenden, S., & Jensen, L. (2011). Quantifying the potential of automated dynamic solar shading in office buildings through integrated simulations of energy and daylight. *Solar Energy*, 757-768.

Ostergard, T., Jensen, R., & Maagaard, S. (2018). A comparison of six metamodeling techniques applied to building performance simulations. *Applied Energy*, 89-103.

Passive House Insitute. (2019). *PHPP.* Retrieved from https://passivehouse.com/04_phpp/04_phpp.htm

Passive House Institute. (2019). *Passive House.* Retrieved from https://passivehouse.com/

Perkin&Will. (2019). Image Source.

Perkins&Will. (2019). *Perkins&Will.* Retrieved from https://perkinswill.com/

Pfeffer, A. (2009). Multi-layer Feed-forward Neural Networks [Lecture Notes]. Harvard University.

Province of British Columbia. (2019, October 8). *Energy Step Code.* Retrieved from Energy Step Code: https://energystepcode.ca/

Qian, X., & Lee, S. (2014). The Design and Analysis of Energy Efficient Building Envelopes for the Commercial Buildings by Mixed-level Factorial Design and Statistical Methods. *Proceeding of American Society of Engineering Education (ASEE).*

Reinhart, C. (2019). *DaySim [Software].* Retrieved from https://daysim.ning.com/

Rezaee, R. (2016). *(Unpublished Thesis). Application of Inverse Modeling to Performance Based Architectural Design in the Early Stage.* Georgia Institute of Technology.

Richman , R., Zirnhelt, H., & Fix, S. (2014). Large-scale building simulation using cloud computing for estimating lifecycle energy consumption. *Canadian Journal of Civil Engineering, 41*(3), 252-262.

Robert McNeel. (2019). *Rhinoceros [Software].* Retrieved from https://www.rhino3d.com

Robinson, C., Dilkina, B., Hubbs, J., Zhang, W., Guhathakurta, S., Brown, M., & Pendyala, R. (2017). Machine learning approaches for estimating commercial building energy consumption. *Applied Energy*, 889-904.

Roudsari, M. (2019). *Honeybee [Software].* Retrieved from https://rhino.github.io/components/honeybee/honeybee.html

Roudsari, M. (2019). *Ladybug [Software].* Retrieved from https://www.grasshopper3d.com/group/ladybug

Russell, S., & Norvig, P. (1995). *Russell, Stuart J., and Peter Norvig. Artificial intelligence: a modern approach.* Pearson Education Limited.

Sadeghifam, A., Zahraee, S., Meynagh, M., & Kiani, I. (2015). Combined use of design of experiments and dynamic building simulation in assessment of energy efficiency in tropical residential buildings. *Energy and Buildings, 86*, 525-533.

SAS. (2019). *JMP [Software].* Retrieved from https://www.jmp.com/en_us/home.html

SAS Institute. (2019). *SAS [Software].* Retrieved from https://www.sas.com/en_ca/home.html

Shevade, S., Keerthi, C., Bhattacharyya, C., & Murthy, K. (1999). Improvements to the SMO Algorithm for SVM Regression. *IEEE Transactions on Neural Networks*.

Shi, X., & Yang, W. (2013). Performance-driven architectural design and optimization technique from a perspective of architects. *Automation in Construction, 32*, 125-135.

Smola, A., & Scholkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing*.

Solemna. (2019). *Diva.* Retrieved from https://www.solemma.com/

Thornton Tomasetti. (2019). *Design Explorer.* Retrieved from http://tt-acm.github.io/DesignExplorer/

Tian, W. (2013). A review of sensitivity analysis methods in building energy analysis. *Renewable and Sustainable Energy Reviews, 20*, 411-419.

Tian, W., Heo, Y., De Wilde, P., Zhanyong, L., Yan, D., Park, C., . . . Augenbroe, G. (2018). A review of uncertainty analysis in building energy assessment. *Renewable and Sustainable Energy Reviews, 93*, 285-301.

Towards Data Science. (2019). *Coding Neural Network — Forward Propagation and Backpropagtion.* Retrieved from https://towardsdatascience.com/coding-neural-network-forward-propagation-and-backpropagtion-ccf8cf369f76

Towards Data Science. (2019). *What the Hell is a Perceptron?* Retrieved from https://towardsdatascience.com/what-the-hell-is-perceptron-626217814f53

TRNSYS. (2019). *TRNSYS.* Retrieved from http://www.trnsys.com/

University of California, Berkley. (2019). *Genopt [Software].* Retrieved from https://simulationresearch.lbl.gov/GO/

University of Waikato. (2019). *Weka [Software].* Retrieved from https://www.cs.waikato.ac.nz/ml/weka/

University of Waikato. (2019, October 19). *Weka javadocs.* Retrieved from Weka javadocs: http://weka.sourceforge.net/doc.stable/

US Department of Energy. (2019). *Building Energy Codes Program.* Retrieved from https://www.energycodes.gov/development/commercial/prototype_models#90.1

US Department of Energy. (2019). *EnergyPlus [Software].* Retrieved from https://energyplus.net/

US Department of Energy. (2019). *Open Studio [Software].* Retrieved from https://www.openstudio.net/

US Green Building Council. (2019). *LEED v4 for Building Design and Construction.* US Green Building Council.

Wang, L., Mathew, P., & Pang, X. (2012). Uncertainties in energy consumption introduced by building operations and weather for a medium-size office building. *Energy and Buildings, 53*, 152-158.

Ward, G. (2019). *Radiance [Software].* Retrieved from https://www.radiance-online.org/

Ware, M. (2019). *Multi-Layer Perceptron [Software].* Retrieved from http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/MultilayerPerceptron.html

Wei, Y., Xingxing, Z., Yong, S., Liang, X., Pan, S., Wu, J., . . . Zhao, X. (2018). A review of data-driven approaches for prediction and classification of building energy consumption. *Renewable and Sustainable Energy Reviews*, 1027-1047.

Welle, B. (2019). *Perkins&Will Speed.* Retrieved from https://speedwiki.io/project/#executive-summary

Welle, B., Hanymaker, J., & Rogers, Z. (2011). ThermalOpt: A methodology for automated BIM-based multidisciplinary thermal simulation for use in optimization environments. *Building Simulation*, 293-313.

Wikipedia. (2019). *Pareto Efficiency.* Retrieved from https://en.wikipedia.org/wiki/Pareto_efficiency

Witten, I. (2011). More Data Mining with Weka. University of Waikato.

Woodbury, R., & Burrow, A. (2006). Whither Design Space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 63-82.

Yan, B., Li, X., Malkawi, A., & Augenbroe, G. (2017). Quantifying uncertainty in outdoor air flow control and its impacts on building performance simulation and fault detection. *Energy and Buildings, 134*, 115-128.

Yang, S., Tian, W., Cubi, E., Meng, Q., Liu, Y., & Wei, L. (2016). Comparison of sensitivity analysis methods in building energy assessment. *Procedia Engineering, 146*, 174-181.

Yu, W., Baizhan, L., Hongyuan, J., Zhang, M., & Wang, D. (2015). Application of multi-objective genetic algorithm to optimize energyefficiency and thermal comfort in building design. *Energy and Buildings*, 135-143.

Zhang, Y., & Korolija, I. (2010). Performing complex parametric simulations with jEPlus. *In SET2010-9th International Conference on Sustainable Energy Technologies*, (pp. 24-27).

Zhou, S., & Dong, L. (2015). Prediction of Daylighting and Energy Performance Using Artificial Neural Network and Support Vector Machine. *American Journal of Civil Engineering and Architecture*, 1-8.