MACHINE LEARNING ALGORITHM COMPARISONS IN THE FIELD OF LOW-ENERGY BUILDING DESIGN AND OPERATION

Blair Birdsell
BFA, University of Calgary, 2005

A Thesis Submitted in Partial Fulfilment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Civil Engineering

We acknowledge with respect the Lekwungen peoples on whose traditional territory the university stands and the Songhees, Esquimalt and WSÁNEC peoples whose historical relationships with the land continue to this day.

Machine learning Algorithm Comparisons in the field of Low-Energy Building Design and Operation

by

Blair Birdsell
BFA, University of Calgary, 2005

Supervisory Committee

Dr. Ralph Evins, Supervisor
(Department of Civil Engineering)

Dr. Thomas Froese, Departmental Member
(Department of Civil Engineering)

Abstract

This work spans multiple disciplines to focus on innovation in data-driven decision-making in building performance, utilizing machine learning algorithms. It provides performance benchmarks and analysis that can guide industry best-practice in building design and operations. Specific outcomes from this work have broad application such as the optimization of building energy efficiency, bettering occupant comfort, potential energy saving retrofits, and improving overall building performance. Data-driven decisions are made by evaluating and comparing data from various sources, including building simulations, historical records, and sensor measurements. Through the application of machine learning tools, this data is transformed into a foundation for effective decision-making by building designers and operators or incorporated into intelligent building management systems.

Gradient Boosted Trees when applied to building energy performance demonstrated robust prediction characteristics across different data types and problem configurations as well as being highly accurate and computationally efficient. In predicting building performance time series, the research documents neural network architectures containing convolutional layers as having the ability to forecast short-term high-frequency variations in airflow and predict monthly minimal and maximum airflow values with high accuracy. These results support their inclusion and application in building management systems for high-performance smart buildings.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

ANN	Artificial Neural Network
CFM	Cubic Feet per Minute
CNN	Convolutional Neural Network
FDC	Fitness Distance Correlation
FLA	Fitness Landscape Analysis
GBT	Gradient Boosted Tree
GP	Gaussian Process
MAE	Mean Absolute Error
MAPE	Mean Absolute Percent Error
MLP	Multilayer Perceptron
MPC	Model Predictive Control
NN	Neural Network
PEF	Primary Energy Factor
R2	R-squared
ResNet	Residual Network
RNN	Recurrent Neural Network
RTU	Roof Top Unit
SVM	Support Vector Machine
SM	Surrogate Model

Author Contributions

All chapters were written by Blair Birdsell under the supervision of Dr. Ralph Evins, University of Victoria. BB implemented the models, develop the methodology, preformed analysis, and wrote the text. RE contributed to developing the methodology, edited the manuscript, and supervised the work.

Two final two papers are included as an appendix. They represent peer reviewed conference papers, both of which have been accepted for presentation at the 18th IBPSA International Conference and Exhibition Building Simulation 2023. Full citations details for each work are provided below:

Chapter two: Birdsell, B. & Evins, R. (2023) *Surrogate modeling performance across a standard set of building design problems*. Submitted to Journal of Building Performance Simulation.

Chapter three: Birdsell, B. & Evins, R. (2023) *A comparison of machine learning functions for time-series prediction in buildings*. Submitted to Energy & Buildings journal.

Appendix B: Birdsell, B. & Evins, R. (2023) *Surrogate Modeling performance for building design problems*. 18th IBPSA International Conference and Exhibition. Shanghai, China.

Appendix C: Birdsell, B. & Evins, R. (2023) *A comparison of machine learning functions for time-series prediction in buildings*. 18th IBPSA International Conference and Exhibition. Shanghai, China.

1 Chapter 1

Introduction

Data-driven decision-making in building performance utilizes data analysis and predictions to inform decisions related to building design and operation. It involves leveraging data from various sources, such as sensor measurements, historical records, simulations, and experiments. All of it aims to allow building designers and operators to optimize building performance by making informed choices based on data.

The following five points highlight common areas where applications of machine learning algorithms affect decision making for low-energy building performance.

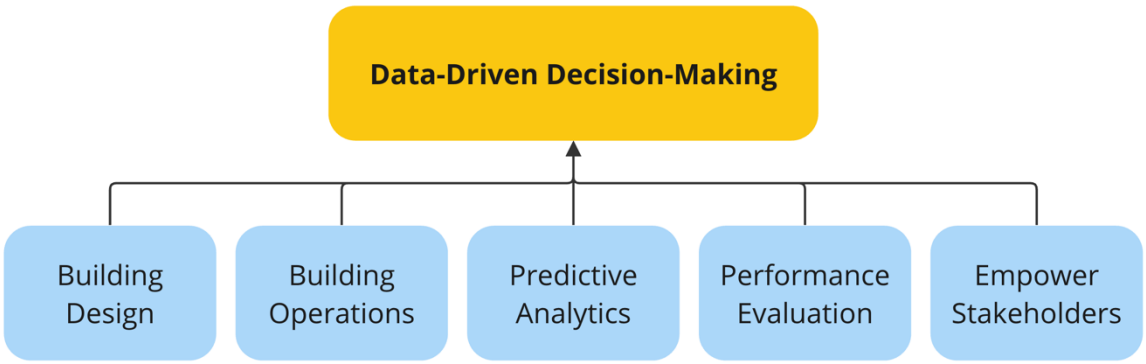

Figure 1.1 Schematic diagram of data-driven decision-making in building performance.

Building design: During the design phase, data plays an important role in assessing and comparing the performance of various design alternatives. This process involves gathering and analysing data from simulations, which aids in the creation of surrogate models that represent the performance of the building. These surrogate models provide valuable insights and inform decisions regarding the optimal design solution.

Building operation: The operational phase of a building presents an opportunity to leverage real-time and historical data to enhance the performance of its systems. It involves analyzing data patterns, correlations, and trends to identify areas for improvement, anomalies, and potential sensor or mechanical failures. Through data-driven decision-making, stakeholders can optimize system control strategies, schedule maintenance activities, and allocate resources effectively.

Predictive analytics: The application of predictive analytics refers to using machine learning techniques and algorithms to predict future performance by analyzing historical data, sensor measurements, and simulations. By analysing these datasets, stakeholders can make accurate estimates about the proposed energy usage patterns and optimize energy management strategies accordingly. This approach enables proactive decision-making and facilitates the optimization of energy efficiency in buildings.

Performance evaluation and benchmarking: The evaluation of performance and benchmarking involves analyzing the effectiveness of various algorithms and performance metrics to assess different approaches or systems utilized by designers or operators. Key indicators such as energy consumption and the status of building control systems are considered in this analysis. By comparing the results, stakeholders can use data to determine the most suitable strategies or technologies to achieve optimal building energy performance.

Empower stakeholders: Data-driven decision-making in building performance enable stakeholders to harness the value of data as a valuable asset. It facilitates informed choices in various aspects, including building design, operation, and maintenance. By leveraging data analysis and predictions, stakeholders can optimize energy efficiency, enhance occupant comfort, manage operational costs, and improve overall building performance. Ultimately, this approach supports the development of sustainable and efficient building that meet the evolving needs of occupants and contribute to a greener future.

Chapter two explores the use of surrogate modeling as a method to reduce the computational load of simulating potential building designs for energy performance. The study examines ten different building design problems using four types of surrogate models and was conducted using the Building and Energy Simulation, Optimization and Surrogate-modelling (BESOS) platform. The findings go on to show that gradient boosted decision trees perform exceptionally well in accurately predicting energy use and capturing the original fitness landscape of the simulated data. The project stands as an example of the type of data modeling and data mining the BESOS platform can perform which can have a positive effect on research and industry.

Chapter three explores methods aimed at enhancing intelligent building management systems. It provides a detailed comparison of different neural network architectures for predicting building performance in both short-term and long-term scenarios. Specifically, four types of neural nets are evaluated: convolutional-based residual neural nets, recurrent neural networks, WaveNet, and a hybrid Residual-RNN model. The findings highlight the strengths of convolutional nets in forecasting sequences and demonstrate superior accuracy in predicting monthly minimum and maximum values when estimating annual performance. By leveraging machine learning techniques, building owners and operators can effectively harness the growing amount of data generated by monitored buildings. The results of this study have significant implications for decision-making in building operation, enhancing the overall efficiency and effectiveness of new and existing buildings.

Chapter four restates the major conclusions of the work. Furthermore, the appendix contains two conference papers of the subject matter contained in chapters two and three entitled, “Surrogate modeling performance across a standard set of building design problems” and “A comparison of machine learning functions for time-series prediction in buildings” respectively.

This opening chapter provides context for the subsequent chapters, having highlighted the overarching theme of data-driven building performance which unifies the entire work. Rather than provide a separate literature review, each chapter contains a detailed review of the relevant prior work. Industry can expect to benefit from research that describes specific aspects and applications of machine learning algorithms in relation to high-performance building design and intelligent building management systems. The following chapters offer multiple avenues for enhancing the decision-making processes for building designers, owners, operators, and occupants by documenting best-practices when applying machine learning techniques.

2 Chapter 2

Surrogate modeling performance across a standard set of building design problems

Birdsell, B. & Evins, R. (2023)
Submitted to Journal of Building Performance Simulation.

2.1 Abstract

Simulating building energy performance is a key part of the modern building design process. To address the computational burden associated with simulations, surrogate modeling has emerged as an efficient approach. This method uses machine learning functions to approximate complex functions, such as physical simulations or optimization algorithms, thereby reducing the computational load during the evaluation of potential building designs. This characteristic of surrogate modeling enables efficient and intuitive exploration of building design problem spaces. To better understand the influence of problem characteristics on surrogate model performance, ten different building design problems are considered using four types of surrogate models. This study utilizes fitness landscape analysis to explore the relationship between the objective function and the input parameters of the building design problem. This analysis provides valuable insights into the difficulty of the problem. The results of the study highlight the exceptional performance of gradient boosted decision trees in accurately predicting energy use and capturing the original fitness landscape of the simulated data. These findings establish a significant connection between problem characteristics and surrogate model performance, serving as a basis for future research on surrogate modeling for building energy performance.

2.2 Introduction

Building energy simulations predict the energy and thermal performance of a building, making them essential to evaluate the building's performance during the design phase. These simulations help improve building design by identifying energy efficiency measures, reducing energy consumption, lowering carbon emissions, and improving occupant comfort (Gan et al. 2020). They provide valuable information for building owners and operators, who can use this data to make informed decisions about energy management, identify retrofit opportunities, and improve building energy performance (Westermann et al. 2019). However, physics-based building energy simulations face challenges due to their computational demands, limiting their widespread use. Surrogate modeling, an application of machine learning, offers a solution by reducing computational load while predicting building performance. This research investigates the accuracy and impact on the fitness landscape associated with surrogate modeling for each design problem. The objective was to see how each machine learning algorithm performed regarding accuracy and reproducing the original fitness landscape using a standard set of building design problems and varying the training data size and features. The results will enable decision-makers to balance performance and accuracy in their surrogate models effectively.

2.3 Related Work

The following section describes the use of surrogate modeling for predicting building energy use and expands on the algorithms chosen for comparison as surrogate models that were implemented in Python's scikit-learn module (Pedregosa et al. 2011).

2.3.1 Surrogate Modeling

With the appropriate training data, surrogate models can approximate complex building energy simulations. The benefits are two-fold: first, the trained model is a fast and accurate way to represent computationally intensive energy simulations. Second, surrogate models allow designers a method to rapidly explore the problem design space of buildings rather than optimize for a pre-

specified objective (Westermann 2021). Surrogate models possess additional characteristics that enhance their functionality compared to physical simulations. These models facilitate sensitivity analysis of input parameters and provide a simpler means of quantifying output uncertainty (Westermann et al. 2019).

Applications of surrogate models such as Prada et. al. (2018) show that surrogate models have greatly reduced computational time when optimizing buildings for energy performance compared to a brute force approach. Comparative analysis has also been a fruitful exploration with surrogate modeling. Hygh et. al. (2012) describes how a surrogate model was able to accurately predict with an R-squared value of better than 0.9 the performance of the same office building model across different climate zones. Retrofits are also a direction of investigation with surrogate models. Ascione et al. (2017) describes a surrogate model's ability to accurately predict within 2% ~ 11% the energy performance of possible retrofits to six existing buildings compared to direct EnergyPlus simulation.

2.3.2 Gaussian Processes

A common point to start understanding Gaussian processes (GP) is linear regression. Linear regression models the relationship between a dependent variable and one or more independent variables. Gaussian processes, in contrast, assume the error is a Gaussian distribution with a linear combination of parameters, allowing for greater flexibility in modeling complex relationships between variables (Kochenderfer 2019). To address the design problems encountered in building simulations, ridge regression with an alpha parameter was an appropriate choice for the Gaussian process model, as each variable has a measurable effect on the output. The alpha parameter is a penalty term used to affect the amount of regularization applied to prevent overfitting, with one being a high value to reduce overfitting and zero leading to no regularization. The use of Gaussian processes has been elaborated on in multi-sensor time series prediction, relevant to the types of

sensor systems found in modern buildings by Osborne et al. (2010). Melo et al. (2016) further extended the application of GP to model cooling demand for several example buildings.

2.3.3 Support Vector Machines

Support vector machines (SVM) are widely recognized for their superior classification performance, as they can create high-dimensional planes that separate different categories of values. However, SVMs are also applicable to regression problems such as time series prediction, which is often overlooked. In his 2009 survey, Sapankevych et al. describes the use of SVMs in predicting financial data time series and environmental parameters such as air quality. Instead of the SVM finding a plane in high-dimensional space that makes distinguishing between points easier, SMV, when used for regression, defines a line in high-dimensional space (the timeseries) from which the algorithm finds coefficients which minimize the error. The performance of SVMs in forecasting energy loads has been extensively studied, as documented by Mohandes (2002) and Chen et al. (2004). Radial basis functions have the ability to capture non-linear behaviour, and when used as a kernel in SVM, can produce excellent results. The radial basis function use as a kernel in a SVM works to project the training data to a higher-dimension where relationships between points are more distinct, therefore allowing the SVM to better operate on them for the purpose of regression. A case study of SMV predicting the cooling demand of buildings resulted in a mean absolute percent error of 4.0% (Melo et al. 2016). During training, the radial basis function's gamma parameter can be adjusted. This parameter represents how distance to neighbouring points influences the approximation.

2.3.4 Gradient Boosted Trees

Decision trees are a popular choice for prediction models due to their human-like reasoning and interpretability. Random Forest and Gradient Boosted Trees are two prominent types of decision trees. In the Random Forest model, multiple agents build a decision tree, and the results are averaged at the end. Gradient Boosted Trees (GBT), however, combine results from the start,

leading to better computational efficiency and accuracy (Friedman 2001). Gradient Boosted Trees were chosen due to the absence of random noise in the training and testing data. This made it unnecessary to employ a Random Forest model, which typically involves averaging to handle such noise. The depth of the tree and the number of estimators were the primary parameters examined in this study, as they significantly influence the model's performance. The tree depth determines the model's ability to learn complex data relationships, while the number of estimators determines the number of boosting rounds used to improve model accuracy. Studies by Papadopoulos et al. (2018) and Touzani et al. (2018) provide strong evidence of GBT's ability to estimate building energy performance.

2.3.5 Multilayer Perceptron

A multilayer perceptron (MLP) is an artificial neural network composed of multiple layers of interconnected neurons that mimics the structure of the human brain (Werbos 1974). MLP models are well-suited for solving problems with non-linear characteristics and their performance improves with an increasing amount of data due to their adaptability. In contrast, GP models have a limited set of static parameters and their performance does not improve significantly with increased training data. The performance of an MLP model depends on the various configurations of layers, including the number of hidden layers and the width of each layer, which can affect the model's ability to learn relationships within the data. Goel et al. (2017) and Khan et al. (2020) have each looked at neural nets' efficacy at predicting energy consumption. Deb et al. (2021) used nerual nets to predict cost-optimized retrofit solutions. Here the results show the NN model was able to provide retrofit solutions with greater computational efficiency than if all potential designs were simulated.

2.4 Building design problems

The following building design problems were collected from the literature by Waibel et al. (2019b) to form a representative sample of building optimization problems in order to compare optimization algorithms. The set of problems was reused in this study for the related purpose of comparing surrogate modeling algorithms.

Building	Building Type	N	Variables	Variable Types	Source
1	Office	4	continuous	Building orientation, window width, shading transmittance	Wetter et al. (2004)
2	Office	4	continuous	"	"
3	Office	4	continuous	"	"
4	Office	13	continuous	Window width, window overhang, shading setpoint, seasonal cooling setpoint, cooling supply air temperature	"
5	Office	13	continuous	"	"
6	Office	13	continuous	"	"
7	Office	13	integer	"	"
8	Office	11	continuous	"	"
9	Office	13	continuous	Window height, cooling air temperature, heating and cooling night and day setpoints.	Kämpf et al. (2010)
10	School	10	continuous	Wall insulation thickness, zone radiator size	Djuric et al. (2007)

Table 2.1. Summary of Building design problems.

2.4.1 Building design problems 1 to 3

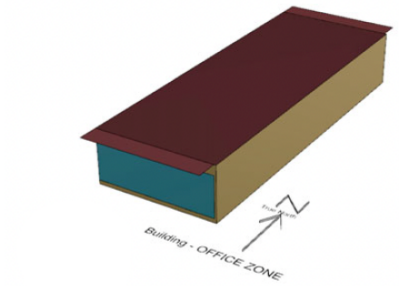

Figure 2.1 Representation of simple office building (Waibel et al. 2019b).

Problems 1 to 3 involve a simple office building with 4 decision variables, as described by Weibel et al. (2019b). The variables determine the building orientation, window widths for the West and East façade, and the transmittance value of the shading overhang. These problems differ only in the weather file used: problem 1 utilizes a typical meteorological year for Seattle, Houston for problem 2, and Chicago for problem 3.

In the cost function for this building, Q_h represents the annual heating energy consumption and Q_c represents the annual cooling energy consumption, both calculated from zone ideal loads in kWh/m^2. E denotes the annual electricity consumption for zone lighting in kWh/m^2. Primary Energy Factor (PEF) converts a building's site energy consumption into the total primary energy required at the source, accounting for generation and distribution losses. The PEF in this set of problems for electricity is set to 3.0, while the plant efficiencies for cooling and heating are represented by $\eta_c = 0.77$ and $\eta_h = 0.44$, respectively. To achieve the objective of these problems, the annual energy consumption is divided by the floor area A, resulting in the minimization of primary annual energy consumption in kWh/m^2.

$$f(x) = \frac{\frac{Q_h(x)}{\eta_h} + \frac{Q_c(x)}{\eta_c} + PEF * E(x)}{A}$$

2.4.2 Building design problems 4 to 8

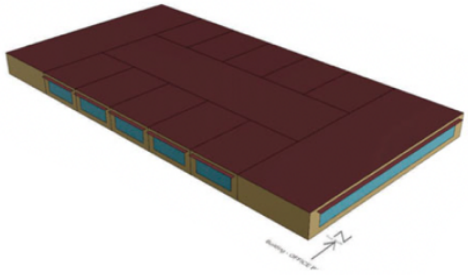

Figure 2.2 Representation of detailed office building (Waibel et al. 2019b).

Building design problem 4 to 6 only differ in respect to the weather file; Seattle, Houston, and Chicago. The building is comprised of two large zones in the East and West direction, along with five smaller zones oriented towards the North and South. The floors and ceilings are considered adiabatic, and the 13 decision variables include window widths, overhang depths, external shading setpoints, setpoints for the zone air temperature, and the cooling design supply air temperature for the HVAC system.

The aim of these tasks is to minimize the annual primary energy consumption, expressed in kWh/m^2, for heating and cooling coils, fans, and zone lighting. The objective function is defined as

$$f(x) = \frac{PEF_{el}\big(E_{el}(x) + E_c(x)\big) + PEF_{gas}\, E_h(x)}{A}$$

where E_{el} represents energy consumption for fans and zone lighting, E_h denotes heating coil energy consumption, and E_c represents cooling coil energy consumption. PEF_{el}=3 and PEF_{gas}=1 signify the primary energy factors for electricity and gas, respectively. All of which is divided by the building's area (Waibel et al. 2019b).

Problem 7 also uses the Chicago weather file but forces the problem from being a continuous design space to a discreet integer design space. Building 8 is also located in Chicago but modifies the problem slightly to hold the setpoint steady at a constant 25°C for the nightly cooling zone air temperature in summer, and the cooling design supply air temperature for the HVAC system is set to 12°C and therefore only has 11 decision variables.

2.4.3 Building design problem 9

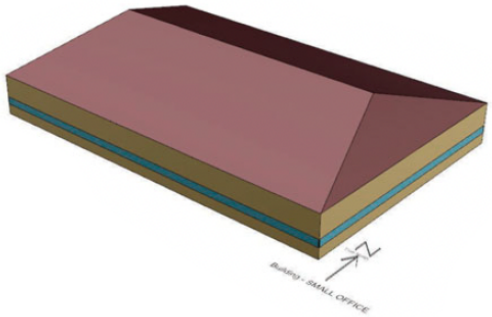

Figure 2.3 Representation of Building 9 (Waibel et al. 2019b).

Problem 9 involves an office building situated in San Francisco, specifically a one-storey office building with two thermal zones, perimeter and core. The decision variables are real-valued and include window positions (controlling window height), cooling supply air temperature, and

temperature setpoints for night setback heating and cooling on different days (weekdays, holidays, and Sundays).

Similar to the previous building design problems 1 to 8, the goal of this problem is to minimize annual primary energy consumption in MJ/m^2 for heating, cooling, and zone lighting. The objective function below incorporates the values of 1.092 for PEF_{gas} and 3.095 for PEF_{el} (Waibel et al. 2019b). Here, E_{el} encompasses electricity usage for zone lighting and cooling systems.

$$f(x) = \frac{PEF_{gas}E_h(x) + PEF_{el}E_{el}(x)}{A}$$

2.4.4 Building design problem 10

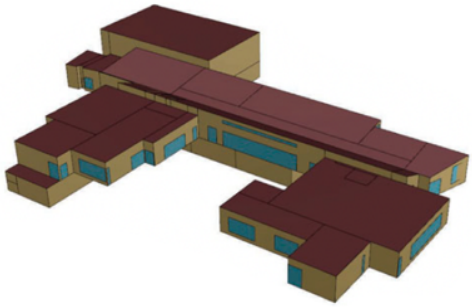

Figure 2.4 Representation of school (Waibel et al. 2019b).

Building design problem 10 focuses on cost optimization of energy, insulation, and radiator sizing for a school building in Belgrade, Serbia. The building model includes nine thermal zones, and each zone has a radiator that can be optimized. In addition, the insulation thickness of external walls is a decision variable. The objective function aims to minimize the total cost in € for heating energy (CE), insulation (C_{in}), and radiator (C_{rad}), and is represented as follows:

$$f(x) = CE(x) + C_{in}(x) + C_{rad}(x)$$

2.5 Methodology

The following section provides a detailed description of the process for training and evaluating the models, including the justification and selection of metrics.

2.5.1 The BESOS platform

The Building and Energy Simulation, Optimization and Surrogate-modelling (BESOS) platform is a comprehensive tool that offers a suite of integrated modules for building energy simulation,

optimization, and surrogate modeling. It enables architects, engineers, and other building professionals to model building energy performance with a high degree of accuracy, while reducing computational load. The platform is flexible, modular, and offers a range of different optimization algorithms, allowing users to find the optimal design solution for their specific building project. The BESOS platform is a powerful tool for building energy simulation and optimization, with applications in building design, construction, and operation, because it allows programmatic interaction via the Python programming language with EnergyPlus and EnergyHub, facilitating the building optimization and the generation of training and testing data used in machine learning applications. A detailed description of the platform's development and capabilities can be found in Westermann et al. (2021).

Complimenting the results of this study, the work undertaken extends the functionality of BESOS by incorporating a standardized set of building design problems. The inclusion of a standardized set of building design problems enables the exploration of innovative machine learning and optimization methods, overcoming potential challenges posed by using physics-based simulations. The entire workflow of generating building simulation data, training surrogate models, and analysing results was conducted on the BESOS platform utilizing Jupyter notebooks. Upon completion of the project the notebooks and associated code will be made available on the platform so that others can use the standard set of building design problems to explore the performance of surrogate models. Final analysis and graphics were produced in Wolfram Mathematica 13.

2.5.2 Training Process

The process of surrogate modeling involves constructing an approximate model using training data. Given the well-established impact of sample size on model performance (Hastie et al. 2009), two sizes of training data were considered. A total of 130 and 1300 input samples were selected for each design problem and simulated using EnergyPlus 9.1.2. The effect of standardization was

also considered. Standardization involves shifting and rescaling the values to achieve a zero mean and unit sample variance. The sampling of values were carried out as defined building problem parameters using a Latin Hypercube methodology to ensure an even dispersion of samples across the problem space. As such, each model, with varying parameters, was trained on four distinct categories of data to acquire knowledge of the cost function. These categories included a large and small data set, with both standardized and non-standardized inputs. The models were trained on 104 and 1040 samples, respectively, and tested against 26 and 260 samples. During the model fitting process, each algorithm for each building problem was trained on the four categories of training data. Each training run completed by the model was then compared to the held back test data to calculate the surrogate model's predictive accuracy on input data it had never seen before.

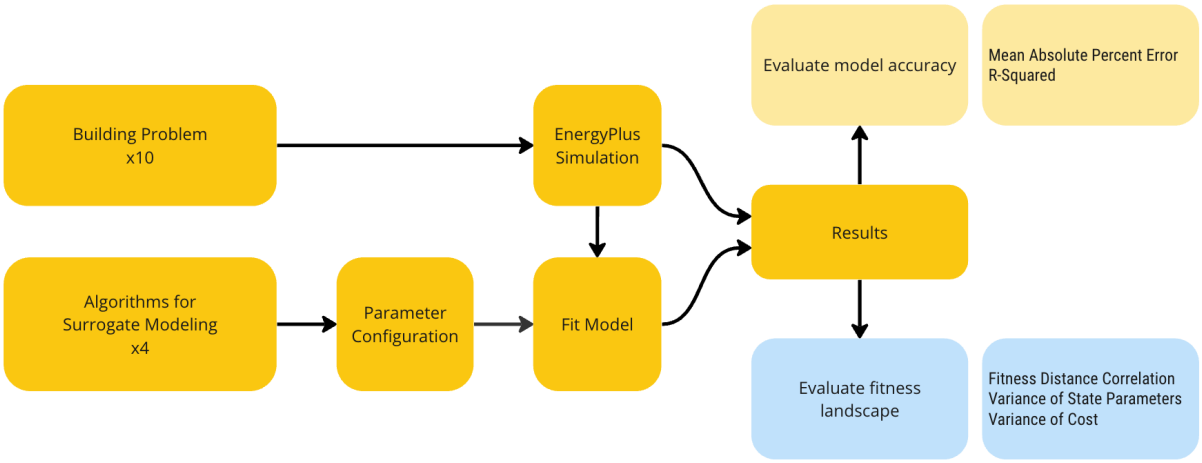

Figure 2.5 Overview of surrogate modeling methodology.

The chosen accuracy metrics return normalized values making comparison between building design problems possible where the absolute cost errors might be orders of magnitude different between problems. The mean absolute percent error subtracts the predicted cost value from the real EnergyPlus cost value and then divides the results by the real cost value to get the error as a percent. The R-squared value divides the variance of the predicted values by the variance of the real values, resulting in a ratio of how much of the variance in the results is explained by the model.

Following the methodology of Waibel et al. (2019a), this study assessed three measures of fitness landscape for each problem across all 49 models: Fitness distance correlation, variance of state parameters, and variance of cost. Based on sampling the model, fitness landscape analysis through evaluation gives insight into the nature of the problem surface. Each is briefly described in the following section.

2.5.3 Fitness Landscape Analysis

Fitness landscape analysis (FLA) aims to quantify the surface characteristics of design problems in terms of their input-output relation, specifically the patterns that arise from the decision variables x in relation to the cost value f(x). "Fitness" refers to the cost value and "landscape" refers to the multi-dimensional surface which is defined by the unknown cost-response f(x) to the input decision variables x (Waibel et al. 2019a). Problem landscapes deemed difficult can be more challenging to optimize.

Fitness Distance Correlation (FDC) is a well-known measure of problem difficulty. The evaluation starts by defining the input values of the minimal cost and then calculating the Euclidean distance of various samples to that minimum. More challenging surfaces show less correlation between the inputs, whereas highly correlated surfaces will show similar inputs have similar costs and approach an FDC value of one.

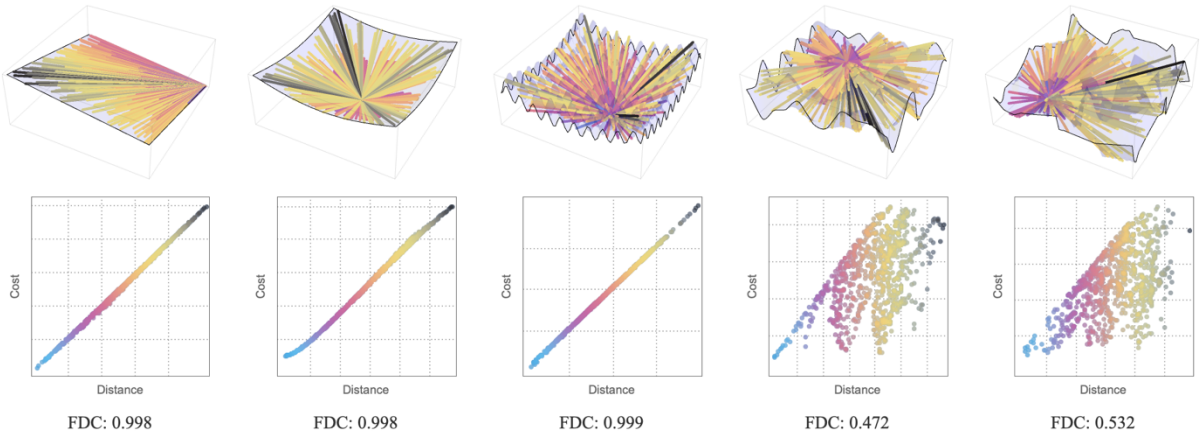

Figure 2.6 Example of FDC for 2 input parameters.
Coloured line represents distance from sampled point to minimum.

The relative variance of the input parameters (the state) in relation to similar costs is another meaningful fitness landscape metric. The evaluation of the variance of state parameters value begins by categorizing or binning the cost values and then analyzing the variance of the inputs within each bin. This is achieved by calculating the distance of each point within the bin and average point of the bin. These distances are subsequently normalized within the bin and variance measured. The average variance of all the bins evaluated in such a way is another approach to quantifying the relationship between input parameters and costs. Smoother, less challenging surfaces will have closely related input values if they are in the same bin, whereas when high variance is detected within a bin, it means that similar cost values can have widely varying inputs in the state space.

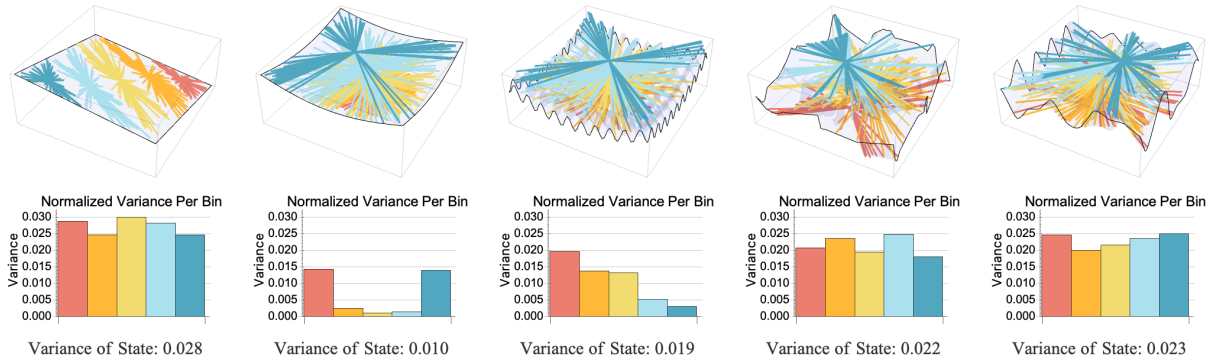

Figure 2.7 Example of variance of state for 2 input parameters. Colours represent the 5 different cost bins in this example.

Finally, a conventional evaluation of cost value variance is provided, normalized before calculation so that problems of different magnitude can be similarly compared. This evaluation can be represented as a probability density function indicating the likelihood of finding low-cost solutions if selected at random. The scarcity or prevalence of low-cost solutions within the design space can impact how challenging a design problem is to optimize.

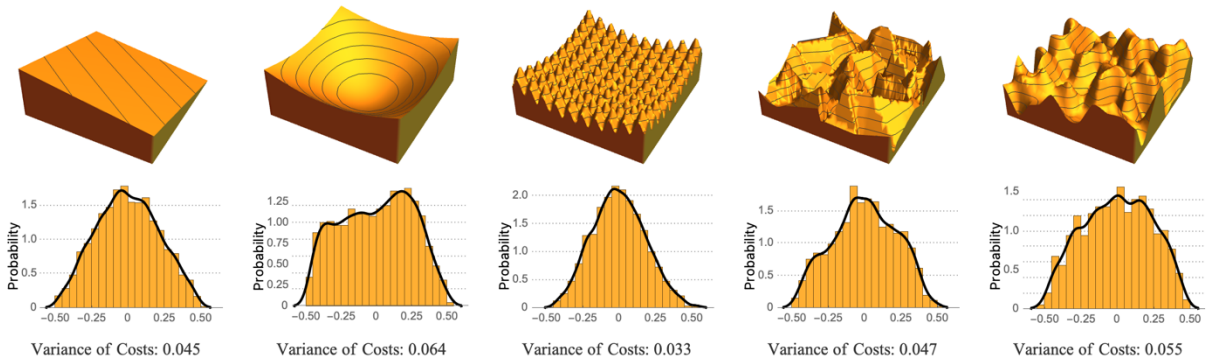

Figure 2.8 Histogram with probability density function overlaid for various 2 input parameters examples.

2.6 Results & Interpretation

This section presents a summary of the results, followed by an in-depth analysis of the error and fitness landscape of each building design problem.

2.6.1 Summary of Results

When analyzing the results, tables 2.2 and 2.3 demonstrate the impact of the number of training samples on model performance. The performance of GP, SVM, and MLP models declines when trained on fewer samples. Building design problem 8 is especially revealing of this trend. Despite standardization, the models lack sufficient training data to effectively learn the design problem space. However, SMV achieves relatively high R-squared values and low MAPE for this specific design problem. Increasing the number of training samples, certain model parameter configurations still exhibit performance that is on par or worse than random. However, the performance of SVM and MLP models with appropriate parameter configurations increases significantly with more training data. When training data is standardized and given in a suitable amount to the model for training, SVM and MLP's performance on Buildings 8, 9 and 10 is marginally better than GBT.

When the analysis is extended to include 1040 non-standardized training samples (table 2.4), GBT as a standout performer emerges. It consistently maintained similar levels of accuracy for standardized and non-standardized input data and surpasses all models trained on 104 samples. SVM and MPL performance deteriorated significantly on non-standardized data except for SVM performance on Building 9. This observation suggests the internal mechanics of SVM work

effectively with the multiple temperature setpoint parameters of the building problem in both standardized and non-standardized formats.

Another perspective on the data presented in figure 2.9 provides a clearer depiction of the most optimal parameter configuration for each model type and illustrates the relationship between model parameters and model performance as the model parameters vary. It is worth noting that, in general, the best configuration of model parameters performs consistently well across all building problems. Adjusting the GP alpha parameter is observed to have minimal impact on model performance. In the case of the SVM, the gamma parameter is typically selected as a smaller value approaching zero, striking a balance that allows the RBF kernel to make accurate inferences from neighbouring data points. GBT consistently exhibits exceptional performance for both 104 and 1040 training samples, with easily identifiable optimal parameter settings. The number of estimators is recognized to have a large impact on GBT accuracy. Last, the plots for MLP models illustrate the positive effect of increasing network depth and widening each layer on model performance, indicating the importance of architecture design choices in achieving better results.

	Standardized Error Values	01		02		03		04		05		06		07		08		09		10	
		R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE
GP	0.	0.7102	3.8676	0.7719	3.8061	−0.1000	4.0264	0.4544	6.4846	0.7369	8.3891	0.5266	7.0634	0.6163	6.2129	−0.1000	0.9332	0.5162	1.8546	0.8895	1.8871
GP	0.0714	0.7101	3.8646	0.7716	3.8046	−0.1000	4.0288	0.4538	6.4806	0.7367	8.3847	0.5264	7.0581	0.6160	6.2103	−0.1000	0.9328	0.5152	1.8547	0.8894	1.8858
GP	0.1429	0.7100	3.8615	0.7714	3.8031	−0.1000	4.0311	0.4531	6.4766	0.7366	8.3802	0.5262	7.0528	0.6158	6.2080	−0.1000	0.9324	0.5142	1.8547	0.8894	1.8844
GP	0.2143	0.7099	3.8585	0.7711	3.8016	−0.1000	4.0335	0.4525	6.4726	0.7364	8.3765	0.5260	7.0475	0.6155	6.2058	−0.1000	0.9319	0.5132	1.8547	0.8894	1.8831
GP	0.2857	0.7098	3.8555	0.7708	3.8001	−0.1000	4.0359	0.4519	6.4686	0.7363	8.3732	0.5257	7.0423	0.6152	6.2037	−0.1000	0.9315	0.5122	1.8547	0.8894	1.8818
GP	0.3571	0.7097	3.8524	0.7706	3.7987	−0.1000	4.0382	0.4512	6.4646	0.7361	8.3698	0.5255	7.0370	0.6150	6.2016	−0.1000	0.9311	0.5112	1.8547	0.8894	1.8804
GP	0.4286	0.7096	3.8494	0.7703	3.7972	−0.1000	4.0406	0.4506	6.4607	0.7359	8.3665	0.5253	7.0318	0.6147	6.1995	−0.1000	0.9306	0.5102	1.8547	0.8893	1.8791
GP	0.5	0.7095	3.8464	0.7701	3.7957	−0.1000	4.0429	0.4500	6.4567	0.7358	8.3632	0.5250	7.0265	0.6144	6.1974	−0.1000	0.9302	0.5092	1.8548	0.8893	1.8778
GP	0.5714	0.7094	3.8434	0.7698	3.7943	−0.1000	4.0453	0.4493	6.4527	0.7356	8.3599	0.5248	7.0213	0.6141	6.1953	−0.1000	0.9298	0.5082	1.8548	0.8893	1.8764
GP	0.6429	0.7093	3.8404	0.7695	3.7928	−0.1000	4.0476	0.4487	6.4488	0.7355	8.3565	0.5245	7.0161	0.6138	6.1932	−0.1000	0.9294	0.5072	1.8548	0.8892	1.8751
GP	0.7143	0.7092	3.8374	0.7692	3.7913	−0.1000	4.0499	0.4480	6.4449	0.7353	8.3532	0.5243	7.0109	0.6136	6.1911	−0.1000	0.9289	0.5062	1.8548	0.8892	1.8738
GP	0.7857	0.7090	3.8344	0.7690	3.7899	−0.1000	4.0523	0.4474	6.4409	0.7351	8.3500	0.5241	7.0057	0.6133	6.1890	−0.1000	0.9285	0.5052	1.8548	0.8892	1.8725
GP	0.8571	0.7089	3.8314	0.7687	3.7884	−0.1000	4.0546	0.4468	6.4370	0.7350	8.3467	0.5238	7.0006	0.6130	6.1869	−0.1000	0.9281	0.5041	1.8548	0.8892	1.8711
GP	0.9286	0.7088	3.8284	0.7684	3.7870	−0.1000	4.0569	0.4461	6.4331	0.7348	8.3434	0.5236	6.9954	0.6127	6.1848	−0.1000	0.9277	0.5031	1.8549	0.8891	1.8698
GP	1.	0.7087	3.8254	0.7682	3.7855	−0.1000	4.0592	0.4455	6.4292	0.7346	8.3402	0.5233	6.9903	0.6124	6.1827	−0.1000	0.9273	0.5021	1.8549	0.8891	1.8685
SVM	gamma: 5e−05	−0.1000	4.6997	−0.1000	4.4929	−0.1000	5.2084	0.5421	5.7514	0.5515	9.2266	0.1997	4.9038	−0.1000	8.5437	−0.1000	0.9862	−0.1000	1.9192	0.2066	8.0118
SVM	gamma: 0.0001	−0.1000	3.8725	0.4872	2.1606	−0.1000	2.8123	0.4633	8.4195	0.2761	8.8967	0.0440	5.3875	−0.1000	6.3848	0.2100	0.6494	0.2768	1.6299	0.4525	7.9674
SVM	gamma: 0.0005	0.3924	3.2757	0.0575	3.8483	0.7427	2.1321	0.7722	4.5761	0.4788	7.5882	0.3486	4.0032	−0.1000	6.3516	0.8658	0.3632	0.7053	1.5840	0.2936	9.3937
SVM	gamma: 0.001	0.4471	2.9092	0.3744	3.2253	−0.1000	3.7112	0.5700	4.5510	0.7772	6.2656	0.7916	3.0023	0.7146	4.6900	0.8575	0.4133	0.6812	1.2515	0.9825	0.8581
SVM	gamma: 0.005	0.4651	2.6323	0.8769	1.4601	0.7551	2.0288	0.7945	4.6335	0.7690	6.2777	0.6723	5.2602	0.3459	6.7377	0.8537	0.3428	0.8542	1.2677	0.9846	0.8585
SVM	gamma: 0.01	0.5759	2.5317	0.9452	1.3560	0.9314	1.4048	0.6656	5.4403	0.9053	4.1731	0.7125	4.1912	0.6370	7.1291	0.8013	0.5425	0.9220	0.8067	0.9782	1.0419
SVM	gamma: 0.05	0.9323	1.9348	0.9708	1.2500	0.7270	2.0886	0.6475	5.2186	0.7626	7.1018	0.3595	5.0165	0.0847	6.6496	0.8555	0.3890	0.5620	1.3954	0.9637	1.2274
SVM	gamma: 0.1	0.9147	1.7431	0.9820	0.9407	0.9544	1.2790	0.0626	5.3347	0.2645	6.9209	−0.1000	5.3994	−0.1000	7.5981	0.0539	0.6104	−0.0575	1.5353	0.8703	2.0093
SVM	gamma: 0.5	0.8599	2.1434	0.8554	1.7681	0.8726	1.6086	−0.1000	10.0000	−0.1000	10.0000	−0.1000	8.9761	−0.1000	10.0000	−0.1000	0.9407	−0.1000	2.2796	−0.1000	8.0048
SVM	gamma: 1.0	0.9073	1.6543	0.3440	3.1312	0.7767	1.5906	−0.1000	10.0000	−0.1000	10.0000	−0.1000	7.5998	−0.1000	10.0000	−0.1000	1.2032	−0.1000	4.2227	−0.1000	6.8363
GBT	depth: 5 est: 25	−0.1000	3.4204	0.7611	2.4194	0.7584	2.6780	0.7560	3.6352	0.5487	5.3309	0.2462	3.2302	0.8466	3.1151	−0.1000	0.7418	0.4186	1.1666	0.5684	3.3343
GBT	depth: 5 est: 50	0.5524	2.4059	0.8287	2.0378	0.9311	1.9595	0.9487	1.8025	0.7958	4.0060	0.6799	2.0413	0.9848	1.5333	0.3839	0.6754	0.7766	0.8609	0.8242	2.4759
GBT	depth: 5 est: 60	0.5958	2.3201	0.8427	1.9992	0.9301	1.9501	0.9600	1.5771	0.8237	3.7825	0.7174	1.8953	0.9859	1.4835	0.4063	0.6788	0.8015	0.8507	0.8391	2.4414
GBT	depth: 5 est: 80	0.6753	2.2153	0.8430	2.0394	0.9424	1.8234	0.9693	1.3942	0.8322	3.8357	0.7749	1.7410	0.9928	1.1670	0.4417	0.6638	0.8291	0.8503	0.8609	2.3622
GBT	depth: 5 est: 100	0.6990	2.1318	0.8495	1.9616	0.9472	1.7645	0.9720	1.3313	0.8347	3.7972	0.7781	1.6877	0.9942	1.1168	0.4502	0.6689	0.8405	0.8541	0.8642	2.3912
GBT	depth: 5 est: 100	0.6851	2.2599	0.8214	2.0422	0.9521	1.7435	0.9774	1.2707	0.8493	3.6271	0.7830	1.6419	0.9926	1.1790	0.4482	0.6641	0.8306	0.8561	0.8564	2.4125
GBT	depth: 5 est: 125	0.7077	2.1032	0.8561	2.0101	0.9462	1.7811	0.9714	1.3067	0.8396	3.8069	0.7661	1.7145	0.9927	1.2091	0.4776	0.6585	0.8409	0.8568	0.8661	2.3751
GBT	depth: 5 est: 150	0.7112	2.0952	0.8549	1.9829	0.9522	1.7430	0.9732	1.3005	0.8439	3.7473	0.8056	1.6633	0.9935	1.1230	0.4502	0.6658	0.8497	0.8088	0.8662	2.3697
GBT	depth: 3 est: 100	0.7763	1.8291	0.8622	1.7623	0.9431	1.5354	0.9599	1.4428	0.9804	1.6614	0.9197	1.2708	0.9807	1.4747	0.5086	0.5589	0.8098	0.9271	0.8475	2.4650
GBT	depth: 10 est: 25	−0.1000	3.6322	0.6141	2.8824	0.7454	2.7692	0.6915	3.9641	0.5878	5.1847	0.4968	3.0652	0.8579	3.0512	−0.1000	0.8140	0.3955	1.2812	0.5144	3.7305
GBT	depth: 10 est: 50	0.5047	2.9875	0.8069	2.5413	0.9299	2.0738	0.9273	2.2672	0.7718	4.4370	0.8036	2.1891	0.9852	1.5353	0.2551	0.8010	0.7465	1.1082	0.7253	3.3722
GBT	depth: 15 est: 25	−0.1000	3.6046	0.6921	2.7305	0.7451	2.7853	0.6942	3.9655	0.5769	5.3241	0.4952	3.0266	0.8577	3.0264	−0.1000	0.8307	0.3635	1.3248	0.5137	3.6771
GBT	depth: 30 est: 25	−0.1000	3.6715	0.6830	2.7484	0.7442	2.7701	0.6957	3.9709	0.5734	5.3284	0.5053	3.0651	0.8556	3.0243	−0.1000	0.8454	0.3781	1.2893	0.4899	3.7252
MLP	(64,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000
MLP	(100,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000
MLP	(150,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.0591	9.0831
MLP	(64, 64)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.2444	8.8810
MLP	(100, 100)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.4806	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.2490	8.0202
MLP	(25, 25, 25)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.5870	9.9034	0.6874	10.0000	−0.1000	10.0000	0.5372	10.0000	0.0455	10.0000	−0.1000	10.0000	0.3882	6.9069
MLP	(64, 64, 64)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.7500	7.8288	0.7893	10.0000	0.3498	7.6617	0.4190	9.6430	0.0824	10.0000	0.0950	10.0000	0.2514	7.1118
MLP	(96, 96, 96)	0.5658	8.9187	0.6303	8.2519	0.3234	10.0000	0.8325	6.1041	0.8042	9.4559	0.3411	9.8280	0.5970	8.8996	0.0506	9.3280	0.0356	10.0000	0.2498	8.0025
MLP	(25, 25, 25, 25)	0.5819	8.5742	0.6278	7.4732	0.4152	9.8948	0.6843	8.8646	0.7884	10.0000	0.3548	10.0000	0.3420	9.0654	0.0610	10.0000	−0.0404	10.0000	0.4140	6.2780
MLP	(64, 64, 64, 64)	0.7166	4.8090	0.6596	5.5576	0.5253	6.0591	0.7535	7.7328	0.8680	7.2374	0.4739	10.0000	0.2469	10.0000	−0.0020	9.3899	−0.0036	10.0000	−0.1000	10.0000

Table 2.2 Table of 104 standardized training samples across set of building problems. Red is more accurate. Blue means lower accuracy values and grey means did not meet threshold performance. Colours are scaled over all problems for each error metric.

	Standardized Error Values	01		02		03		04		05		06		07		08		09		10	
		R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE
GP	0.	0.5611	4.0378	0.7891	2.9057	0.6041	3.7102	0.6644	5.9223	0.5669	8.5920	0.4344	5.4391	0.3769	8.0812	0.1087	0.7727	0.6697	1.4618	0.9223	1.9771
GP	0.0714	0.5611	4.0376	0.7891	2.9056	0.6040	3.7101	0.6643	5.9221	0.5668	8.5918	0.4343	5.4389	0.3768	8.0810	0.1086	0.7727	0.6696	1.4618	0.9223	1.9770
GP	0.1429	0.5610	4.0375	0.7891	2.9055	0.6039	3.7101	0.6643	5.9219	0.5668	8.5916	0.4342	5.4388	0.3767	8.0808	0.1085	0.7727	0.6696	1.4617	0.9223	1.9769
GP	0.2143	0.5609	4.0373	0.7890	2.9054	0.6039	3.7100	0.6643	5.9217	0.5667	8.5913	0.4342	5.4386	0.3766	8.0806	0.1084	0.7727	0.6695	1.4617	0.9222	1.9768
GP	0.2857	0.5609	4.0372	0.7890	2.9053	0.6038	3.7100	0.6642	5.9215	0.5666	8.5911	0.4341	5.4384	0.3765	8.0804	0.1083	0.7726	0.6695	1.4616	0.9222	1.9766
GP	0.3571	0.5608	4.0371	0.7890	2.9052	0.6037	3.7099	0.6642	5.9213	0.5666	8.5909	0.4340	5.4382	0.3764	8.0802	0.1081	0.7726	0.6695	1.4616	0.9222	1.9765
GP	0.4286	0.5608	4.0369	0.7890	2.9050	0.6037	3.7099	0.6641	5.9211	0.5665	8.5907	0.4339	5.4381	0.3763	8.0800	0.1080	0.7726	0.6694	1.4616	0.9222	1.9764
GP	0.5	0.5607	4.0368	0.7889	2.9049	0.6036	3.7098	0.6641	5.9209	0.5664	8.5905	0.4339	5.4379	0.3763	8.0798	0.1079	0.7726	0.6694	1.4615	0.9222	1.9763
GP	0.5714	0.5606	4.0366	0.7889	2.9048	0.6035	3.7098	0.6641	5.9206	0.5664	8.5903	0.4338	5.4377	0.3762	8.0796	0.1078	0.7726	0.6693	1.4615	0.9222	1.9761
GP	0.6429	0.5606	4.0365	0.7889	2.9047	0.6035	3.7097	0.6640	5.9204	0.5663	8.5901	0.4337	5.4376	0.3761	8.0794	0.1077	0.7726	0.6693	1.4614	0.9222	1.9760
GP	0.7143	0.5605	4.0363	0.7889	2.9046	0.6034	3.7097	0.6640	5.9202	0.5662	8.5899	0.4336	5.4374	0.3760	8.0792	0.1076	0.7726	0.6692	1.4614	0.9222	1.9759
GP	0.7857	0.5605	4.0362	0.7888	2.9045	0.6033	3.7096	0.6639	5.9200	0.5662	8.5897	0.4335	5.4372	0.3759	8.0790	0.1075	0.7725	0.6692	1.4614	0.9222	1.9758
GP	0.8571	0.5604	4.0360	0.7888	2.9044	0.6033	3.7095	0.6639	5.9198	0.5661	8.5895	0.4335	5.4371	0.3758	8.0788	0.1074	0.7725	0.6691	1.4613	0.9222	1.9756
GP	0.9286	0.5603	4.0359	0.7888	2.9043	0.6032	3.7095	0.6639	5.9196	0.5660	8.5893	0.4334	5.4369	0.3757	8.0786	0.1073	0.7725	0.6691	1.4613	0.9221	1.9755
GP	1.	0.5603	4.0358	0.7887	2.9042	0.6031	3.7094	0.6638	5.9194	0.5660	8.5891	0.4333	5.4367	0.3756	8.0784	0.1072	0.7725	0.6691	1.4612	0.9221	1.9754
SVM	gamma: 5e−05	−0.1000	4.9663	0.4495	2.5719	−0.1000	3.6139	0.4193	5.8984	0.5351	7.3957	−0.1000	4.7752	−0.1000	6.9000	0.7766	0.4212	0.5124	1.2140	0.4381	8.4216
SVM	gamma: 0.0001	−0.1000	3.5026	0.4755	2.5649	−0.0812	3.5133	0.7080	4.4054	0.6447	6.3839	0.2314	3.6207	−0.1000	6.3285	0.9160	0.2874	0.8040	0.8021	0.2982	10.0000
SVM	gamma: 0.0005	0.0275	3.3174	0.6387	2.7018	−0.0214	3.7775	0.9631	1.4403	0.9632	2.0014	0.9116	1.7234	0.6895	3.1561	0.9280	0.2840	0.9667	0.4841	0.0863	10.0000
SVM	gamma: 0.001	0.2134	3.2635	0.5814	2.6138	0.1733	3.2876	0.9541	1.5435	0.9709	1.9191	0.8983	1.7010	0.9088	2.1667	0.9484	0.2533	0.9683	0.4845	0.0154	10.0000
SVM	gamma: 0.005	0.8157	2.0252	0.9262	1.3938	0.9303	1.5847	0.9715	1.4784	0.9762	2.2163	0.9584	1.3108	0.8873	1.9973	0.9235	0.2887	0.9978	0.1289	0.9748	1.1001
SVM	gamma: 0.01	0.9041	1.7241	0.9531	1.2193	0.9376	1.4149	0.9643	1.7401	0.9783	2.3233	0.9564	1.5983	0.8989	2.7959	0.8877	0.3823	0.9953	0.1631	0.9967	0.3799
SVM	gamma: 0.05	0.9795	1.2243	0.9914	0.6285	0.9827	0.9018	0.9511	1.8874	0.9646	2.5054	0.9171	1.9274	0.8864	3.1485	0.8956	0.3117	0.9514	0.4489	0.9935	0.4408
SVM	gamma: 0.1	0.9893	0.7957	0.9949	0.4551	0.9912	0.6167	0.9418	2.0385	0.8954	3.6046	0.7795	2.5553	0.8359	3.6731	0.9011	0.3256	0.9178	0.6757	0.9820	0.7009
SVM	gamma: 0.5	0.9956	0.4141	0.9983	0.2145	0.9974	0.3089	−0.1000	8.8230	−0.1000	10.0000	−0.1000	7.5983	−0.1000	10.0000	−0.1000	0.7482	−0.1000	2.3367	0.1074	3.4780
SVM	gamma: 1.0	0.9854	0.5546	0.9971	0.3211	0.9967	0.3157	−0.1000	10.0000	−0.1000	10.0000	−0.1000	8.5003	−0.1000	10.0000	−0.1000	0.9265	−0.1000	2.5996	−0.1000	6.3306
GBT	depth: 5 est: 25	0.7168	2.5755	0.8027	2.3004	0.7636	2.2561	0.8083	3.1012	0.8049	4.1162	0.8252	2.4834	0.8477	3.3772	0.2486	0.5542	0.7002	0.9817	0.5752	3.0592
GBT	depth: 5 est: 50	0.9477	1.4112	0.9748	1.0248	0.9606	1.1763	0.9792	1.2824	0.9741	1.6286	0.9818	1.0175	0.9900	1.1490	0.7827	0.3782	0.9256	0.6303	0.9013	1.9262
GBT	depth: 5 est: 60	0.9645	1.1832	0.9830	0.8537	0.9724	0.9941	0.9869	1.0772	0.9819	1.3463	0.9886	0.8521	0.9949	0.8704	0.8297	0.3467	0.9397	0.5756	0.9258	1.7290
GBT	depth: 5 est: 80	0.9779	0.9209	0.9894	0.6623	0.9813	0.8060	0.9917	0.8930	0.9869	1.1095	0.9927	0.7046	0.9978	0.6220	0.8798	0.3010	0.9535	0.5113	0.9490	1.4875
GBT	depth: 5 est: 100	0.9831	0.7640	0.9916	0.5674	0.9844	0.7158	0.9933	0.8107	0.9883	1.0052	0.9937	0.6444	0.9985	0.5179	0.9032	0.2744	0.9595	0.4744	0.9590	1.3575
GBT	depth: 5 est: 100	0.9866	0.6570	0.9931	0.4911	0.9878	0.6218	0.9945	0.7255	0.9900	0.8798	0.9942	0.5884	0.9990	0.4191	0.9291	0.2377	0.9682	0.4208	0.9690	1.1842
GBT	depth: 5 est: 125	0.9856	0.6893	0.9926	0.5093	0.9860	0.6611	0.9939	0.7668	0.9892	0.9362	0.9944	0.5984	0.9988	0.4500	0.9192	0.2523	0.9646	0.4436	0.9655	1.2582
GBT	depth: 5 est: 150	0.9866	0.6578	0.9930	0.4900	0.9867	0.6378	0.9943	0.7406	0.9896	0.8975	0.9941	0.5868	0.9990	0.4162	0.9264	0.2412	0.9666	0.4267	0.9688	1.1978
GBT	depth: 3 est: 100	0.9739	0.9736	0.9841	0.8314	0.9764	0.9739	0.9921	0.9016	0.9871	1.2339	0.9899	0.8003	0.9967	0.7618	0.8580	0.3223	0.9467	0.5476	0.9354	1.6317
GBT	depth: 10 est: 25	0.7523	2.4476	0.8218	2.1710	0.7969	2.1408	0.8209	3.0616	0.7974	4.2331	0.7843	2.5430	0.8549	3.3205	0.4009	0.5309	0.6901	1.0519	0.6004	3.0999
GBT	depth: 10 est: 50	0.9378	1.4118	0.9687	1.1018	0.9450	1.3292	0.9743	1.3811	0.9630	1.8570	0.9510	1.3051	0.9907	1.0433	0.7560	0.4137	0.8691	0.7944	0.8490	2.4250
GBT	depth: 15 est: 25	0.7476	2.4740	0.8252	2.1707	0.7938	2.1731	0.8171	3.0784	0.7972	4.2769	0.7749	2.6012	0.8552	3.3185	0.3461	0.5713	0.6409	1.1344	0.5647	3.2779
GBT	depth: 30 est: 25	0.7464	2.4802	0.8223	2.1761	0.7939	2.1736	0.8159	3.0941	0.7944	4.2741	0.7757	2.5857	0.8544	3.3250	0.3660	0.5614	0.6402	1.1411	0.5607	3.2746
MLP	(64,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.9666	1.2405
MLP	(100,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.9867	0.7459
MLP	(150,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.9949	0.4483
MLP	(64, 64)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.8595	3.6875	0.8980	4.3832	0.8652	3.3528	0.8120	5.0835	0.3634	2.0921	0.4488	3.6079	0.9918	0.5981
MLP	(100, 100)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	0.9372	2.5621	0.9386	3.5244	0.9046	2.8603	0.8802	3.4052	0.5557	1.0908	0.6128	2.5227	0.9917	0.5718
MLP	(25, 25, 25)	0.8315	3.8837	0.8908	2.8694	0.8776	3.0395	0.9707	1.7303	0.9618	2.6961	0.9448	1.8976	0.8968	3.3029	0.7216	0.6028	0.8689	1.0470	0.9986	0.2311
MLP	(64, 64, 64)	0.9374	1.9437	0.9769	1.1891	0.9599	1.4559	0.9658	1.8863	0.9620	2.7915	0.9370	2.0763	0.9154	2.6943	0.8139	0.4835	0.9383	0.6241	0.9915	0.5745
MLP	(96, 96, 96)	0.9532	1.6851	0.9781	1.1392	0.9697	1.2760	0.9686	1.8453	0.9649	2.6227	0.9448	1.9651	0.9095	2.8492	0.8190	0.4522	0.9424	0.6210	0.9937	0.5062
MLP	(25, 25, 25, 25)	0.9570	1.6010	0.9765	1.2022	0.9632	1.3920	0.9568	2.1576	0.9647	2.6063	0.9366	2.1868	0.9097	2.9686	0.7357	0.5213	0.9165	0.7842	0.9929	0.5686
MLP	(64, 64, 64, 64)	0.9609	1.5754	0.9846	0.9741	0.9744	1.1573	0.9746	1.6095	0.9604	2.7991	0.9392	2.1191	0.9086	2.8887	0.8076	0.4808	0.9461	0.6148	0.9918	0.5696

Table 2.3 Table of 1040 standardized training samples across set of building problems. Red is more accurate. Blue means lower accuracy values and grey means did not meet threshold performance. Colours are scaled over all problems for each error metric.

	Non–Standardized Error Values	01		02		03		04		05		06		07		08		09		10	
		R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE	R2	MAPE
GP	0.	0.6055	4.2265	0.7665	3.1494	0.6028	3.8760	0.5910	5.8244	0.4712	8.9506	0.3222	5.4832	0.3623	7.9336	0.0985	0.7525	0.6580	1.5115	0.9325	1.8973
GP	0.0714	0.6055	4.2264	0.7665	3.1493	0.6028	3.8759	0.5910	5.8243	0.4711	8.9506	0.3221	5.4831	0.3623	7.9335	0.0985	0.7525	0.6577	1.5115	0.9310	1.8851
GP	0.1429	0.6055	4.2264	0.7665	3.1492	0.6028	3.8759	0.5910	5.8243	0.4711	8.9505	0.3221	5.4831	0.3622	7.9335	0.0985	0.7525	0.6574	1.5115	0.9292	1.8754
GP	0.2143	0.6055	4.2263	0.7665	3.1492	0.6028	3.8758	0.5909	5.8242	0.4710	8.9505	0.3220	5.4830	0.3622	7.9334	0.0984	0.7525	0.6571	1.5115	0.9273	1.8671
GP	0.2857	0.6054	4.2263	0.7665	3.1491	0.6027	3.8758	0.5909	5.8242	0.4710	8.9504	0.3220	5.4829	0.3622	7.9333	0.0984	0.7525	0.6567	1.5116	0.9251	1.8619
GP	0.3571	0.6054	4.2262	0.7665	3.1490	0.6027	3.8757	0.5909	5.8241	0.4709	8.9504	0.3220	5.4828	0.3622	7.9332	0.0984	0.7525	0.6564	1.5116	0.9228	1.8579
GP	0.4286	0.6054	4.2262	0.7665	3.1490	0.6027	3.8757	0.5909	5.8241	0.4709	8.9503	0.3219	5.4828	0.3621	7.9332	0.0984	0.7525	0.6561	1.5116	0.9203	1.8561
GP	0.5	0.6054	4.2261	0.7665	3.1489	0.6027	3.8756	0.5908	5.8240	0.4708	8.9503	0.3219	5.4827	0.3621	7.9331	0.0983	0.7525	0.6558	1.5116	0.9175	1.8550
GP	0.5714	0.6054	4.2261	0.7665	3.1488	0.6027	3.8755	0.5908	5.8240	0.4708	8.9502	0.3218	5.4826	0.3621	7.9330	0.0983	0.7525	0.6555	1.5116	0.9146	1.8554
GP	0.6429	0.6053	4.2260	0.7665	3.1488	0.6026	3.8755	0.5908	5.8239	0.4708	8.9502	0.3218	5.4826	0.3621	7.9329	0.0983	0.7525	0.6552	1.5117	0.9115	1.8570
GP	0.7143	0.6053	4.2260	0.7665	3.1487	0.6026	3.8754	0.5907	5.8239	0.4707	8.9501	0.3217	5.4825	0.3620	7.9329	0.0983	0.7525	0.6549	1.5117	0.9083	1.8609
GP	0.7857	0.6053	4.2259	0.7664	3.1486	0.6026	3.8754	0.5907	5.8239	0.4707	8.9501	0.3217	5.4824	0.3620	7.9328	0.0983	0.7525	0.6546	1.5118	0.9048	1.8679
GP	0.8571	0.6053	4.2259	0.7664	3.1485	0.6026	3.8753	0.5907	5.8238	0.4706	8.9500	0.3216	5.4824	0.3620	7.9327	0.0982	0.7525	0.6543	1.5119	0.9012	1.8760
GP	0.9286	0.6053	4.2258	0.7664	3.1485	0.6025	3.8753	0.5906	5.8238	0.4706	8.9500	0.3216	5.4823	0.3619	7.9327	0.0982	0.7525	0.6540	1.5119	0.8975	1.8856
GP	1.	0.6053	4.2258	0.7664	3.1484	0.6025	3.8752	0.5906	5.8237	0.4705	8.9499	0.3215	5.4822	0.3619	7.9326	0.0982	0.7525	0.6537	1.5120	0.8936	1.8953
SVM	gamma: 5e–05	−0.1000	3.4620	0.2180	3.0659	0.1455	3.2387	0.1859	10.0000	0.1265	10.0000	0.0986	10.0000	0.1166	10.0000	0.4449	0.7981	0.9491	0.4934	−0.1000	10.0000
SVM	gamma: 0.0001	−0.1000	3.4833	0.4652	2.6798	0.4501	2.7810	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	1.3014	0.9673	0.4877	−0.1000	9.0582
SVM	gamma: 0.0005	0.3966	2.6606	0.5129	2.7666	0.3787	3.1565	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	1.2096	0.9964	0.1538	−0.1000	6.4513
SVM	gamma: 0.001	−0.1000	3.4372	0.7389	2.2992	0.6279	2.3472	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.7397	−0.1000	10.0000	−0.1000	1.1523	0.9990	0.0876	−0.1000	7.2899
SVM	gamma: 0.005	0.7767	1.9003	0.9022	1.3550	0.8293	1.5112	−0.1000	10.0000	−0.1000	10.0000	−0.1000	8.1453	−0.1000	10.0000	−0.1000	1.0255	0.9994	0.0659	−0.1000	6.8702
SVM	gamma: 0.01	0.8755	1.6823	0.9497	1.0402	0.8893	1.5474	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.2358	−0.1000	10.0000	−0.1000	1.0493	0.9988	0.0796	−0.1000	7.0410
SVM	gamma: 0.05	0.9532	1.4335	0.9553	1.0174	0.8743	1.7078	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.0204	−0.1000	10.0000	−0.1000	1.0914	0.9805	0.3121	−0.1000	7.1547
SVM	gamma: 0.1	0.7526	2.4070	0.9043	1.5225	0.8019	2.2974	−0.1000	10.0000	−0.1000	10.0000	−0.1000	8.9888	−0.1000	10.0000	−0.1000	1.1126	0.9084	0.6725	−0.1000	7.0495
SVM	gamma: 0.5	−0.1000	4.5491	−0.1000	4.0503	−0.1000	4.2884	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.0686	0	10.0000	0	1.0653	−0.1000	1.8040	0	7.2071
SVM	gamma: 1.0	−0.1000	6.3013	−0.1000	5.7853	−0.1000	5.9182	0	10.0000	−0.1000	10.0000	−0.1000	8.6627	−0.1000	10.0000	−0.1000	1.0884	−0.1000	2.5865	−0.1000	7.4178
GBT	depth: 5 est: 25	0.7718	2.4908	0.7458	2.1711	0.8164	2.3961	0.8483	2.9419	0.8256	4.5062	0.7868	2.4555	0.8461	3.2196	0.4393	0.5124	0.6747	1.1582	0.5870	2.9690
GBT	depth: 5 est: 50	0.9598	1.3399	0.9621	1.0210	0.9761	1.1439	0.9849	1.1549	0.9846	1.6750	0.9670	1.0639	0.9893	1.1589	0.8363	0.3455	0.9193	0.7300	0.9046	1.8342
GBT	depth: 5 est: 60	0.9717	1.1342	0.9745	0.8669	0.9848	0.9338	0.9897	0.9829	0.9912	1.2984	0.9763	0.9113	0.9945	0.9019	0.8718	0.3127	0.9367	0.6591	0.9289	1.6422
GBT	depth: 5 est: 80	0.9816	0.8841	0.9843	0.6775	0.9906	0.7152	0.9929	0.8358	0.9952	0.9752	0.9828	0.7655	0.9974	0.6785	0.9083	0.2712	0.9527	0.5748	0.9496	1.4294
GBT	depth: 5 est: 100	0.9852	0.7515	0.9878	0.5820	0.9924	0.6208	0.9938	0.7875	0.9962	0.8571	0.9845	0.7005	0.9981	0.5732	0.9253	0.2462	0.9599	0.5292	0.9595	1.2997
GBT	depth: 5 est: 100	0.9883	0.6389	0.9905	0.4981	0.9938	0.5365	0.9946	0.7044	0.9968	0.7278	0.9854	0.6371	0.9986	0.4876	0.9426	0.2241	0.9671	0.4736	0.9688	1.1770
GBT	depth: 5 est: 125	0.9872	0.6747	0.9893	0.5341	0.9933	0.5647	0.9944	0.7446	0.9966	0.7816	0.9860	0.6565	0.9985	0.5084	0.9350	0.2296	0.9652	0.4937	0.9656	1.2113
GBT	depth: 5 est: 150	0.9883	0.6424	0.9900	0.5103	0.9936	0.5405	0.9947	0.7214	0.9968	0.7466	0.9862	0.6308	0.9987	0.4765	0.9413	0.2189	0.9674	0.4766	0.9688	1.1607
GBT	depth: 3 est: 100	0.9814	0.9421	0.9814	0.7915	0.9856	0.8909	0.9923	0.8727	0.9943	1.0915	0.9888	0.8431	0.9967	0.7886	0.8816	0.3125	0.9438	0.6359	0.9361	1.6121
GBT	depth: 10 est: 25	0.7485	2.4994	0.7418	2.1745	0.8183	2.3767	0.8598	2.8997	0.8233	4.4525	0.7951	2.4344	0.8494	3.2613	0.4709	0.5059	0.6393	1.1913	0.6683	2.8159
GBT	depth: 10 est: 50	0.9353	1.4829	0.9435	1.1942	0.9599	1.3243	0.9812	1.2772	0.9760	1.8963	0.9522	1.2475	0.9898	1.1251	0.7786	0.3903	0.8630	0.8958	0.8770	2.1483
GBT	depth: 15 est: 25	0.7426	2.5573	0.7317	2.2114	0.8136	2.4684	0.8556	2.9480	0.8144	4.5616	0.7846	2.4163	0.8472	3.3191	0.4706	0.5242	0.6293	1.2373	0.6375	2.9826
GBT	depth: 30 est: 25	0.7372	2.5889	0.7343	2.2089	0.8134	2.4781	0.8566	2.9261	0.8168	4.5555	0.7926	2.3919	0.8469	3.3121	0.4504	0.5313	0.6157	1.2492	0.6294	3.0081
MLP	(64,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.0903	10.0000	−0.1000	6.9694	−0.1000	6.7339
MLP	(100,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	6.6328	−0.0015	4.0058	−0.1000	6.7414
MLP	(150,)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	5.1296	0.0415	3.8662	−0.1000	6.8058
MLP	(64, 64)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.2536	−0.1000	10.0000	0.1077	1.2495	0.5740	2.4402	−0.1000	6.7261
MLP	(100, 100)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.6038	−0.1000	10.0000	0.1028	1.5822	0.5839	2.3976	−0.1000	6.8667
MLP	(25, 25, 25)	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.6144	−0.1000	10.0000	0.0802	1.7346	0.5936	2.3489	−0.1000	6.7916
MLP	(64, 64, 64)	0.4186	6.7268	0.4307	6.9171	0.4025	7.1863	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.1990	−0.1000	10.0000	0.1613	1.1123	0.5984	2.2869	−0.1000	6.7838
MLP	(96, 96, 96)	0.7655	3.7341	0.6694	4.3689	0.6933	4.2373	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.7487	−0.1000	10.0000	0.0722	1.7105	0.5930	2.3320	−0.1000	6.8095
MLP	(25, 25, 25, 25)	0.8690	2.7414	0.7145	4.0314	0.5138	5.7291	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.2468	−0.1000	10.0000	0.1000	1.5120	0.5967	2.3272	−0.1000	6.8016
MLP	(64, 64, 64, 64)	0.9616	1.4008	0.9754	0.9973	0.9723	1.1557	−0.1000	10.0000	−0.1000	10.0000	−0.1000	9.5254	−0.1000	10.0000	0.0986	1.5435	0.5802	2.5013	−0.1000	6.8156

Table 2.4 Table of 1040 non-standardized training samples across set of building problems. Red is more accurate. Blue means lower accuracy values and grey means did not meet threshold performance. Colours are scaled over all problems for each error metric.

Figure 2.9 Accuracy measure as model parameter varies for standardized N=104 (top) and N=1040 (bottom). Each building represents a line. Line colour is the average accuracy for the building as parameters vary. Red signifies higher accuracy, blue lower.

2.6.2 Best Performers

Taking the top performers of each model by R-squared value reveals important information applicable to surrogate modelling (table 2.5). The non-linear nature of building design problems 4, 5, and 6 introduced by the geometric problem parameters meant models with more layers or more estimators performed well. The results document how the performance of SVM and MLP models improve when inputs are standardized. Building 8 shows the most dispersion in the pair plots which matches the relatively lower accuracy for all models and data types recorded below. The strong performance of SVM on Building 9 in both standardized and non-standardized formats is again seen. With the more linear nature of building design problem 10, SVM and MLP predict values with extremely high accuracy, whereas the GP is limited in accuracy because of the fundamental assumption Gaussian noise is present, with standardization being essential for SVM and MLP models to perform well. In the context of overall performance evaluation, it is crucial to acknowledge the consistently high performance of the GBT model, which exhibits comparable results for both the standardized and non-standardized training sets. This observation highlights the robustness and reliability of the GBT model across different data configurations and reinforces its position as a top-performing model.

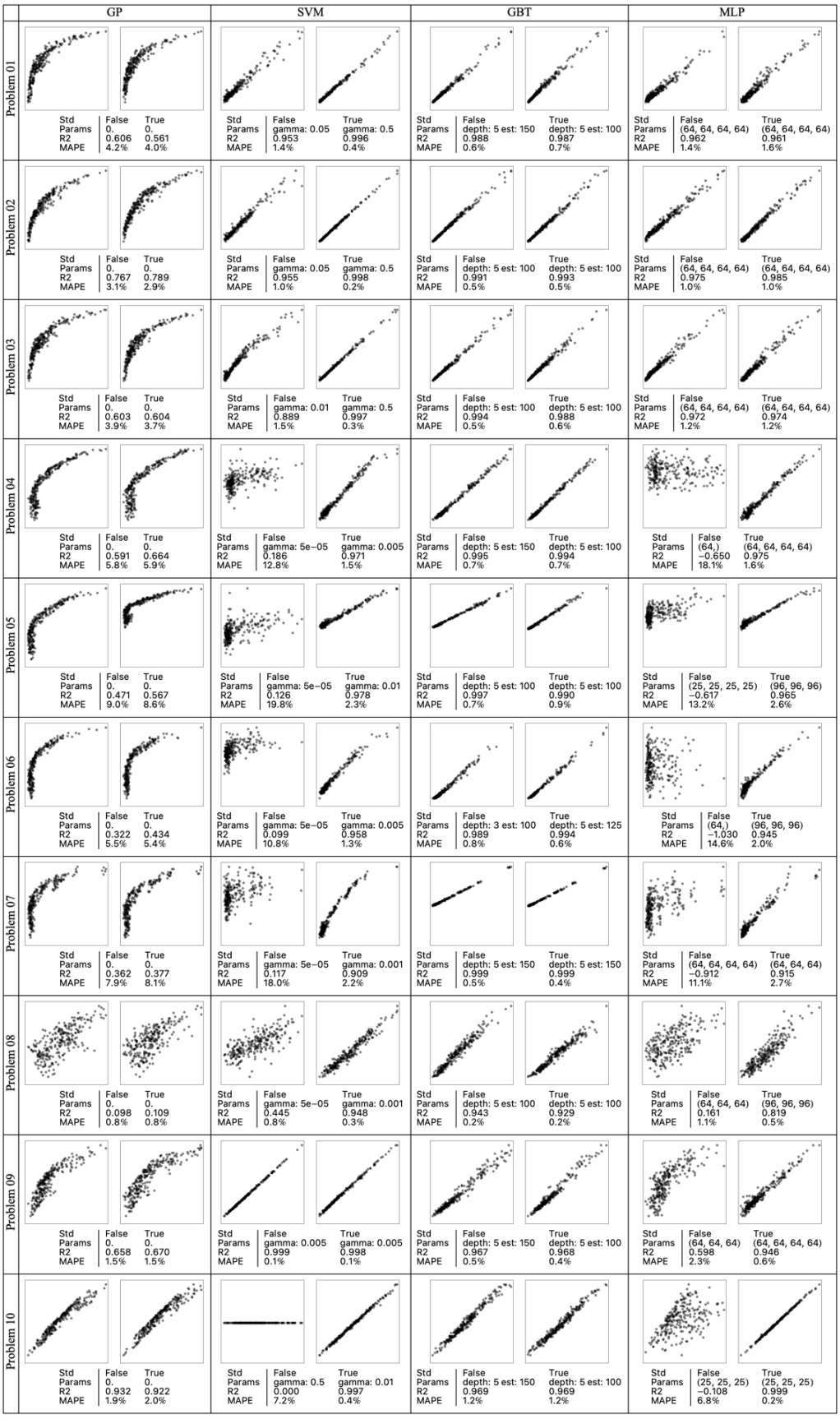

Table 2.5 Best performing models for each building problem using 1040 training samples.

2.6.3 Fitness landscape results

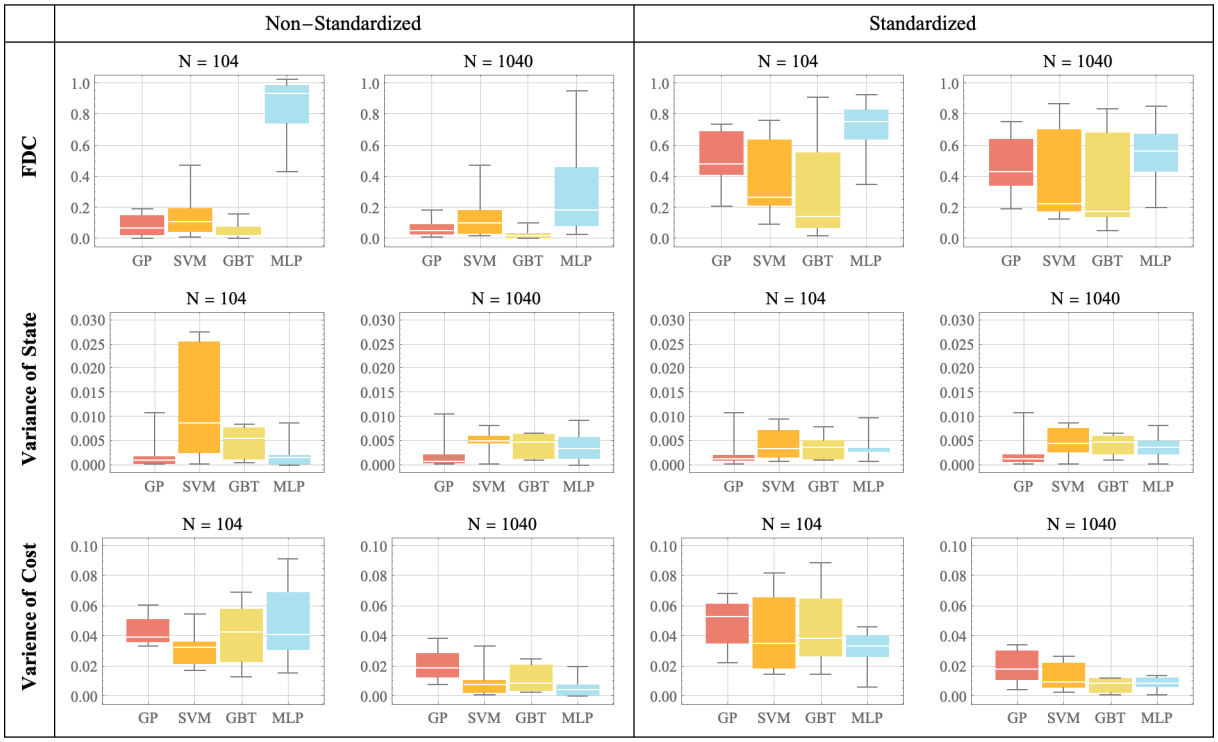

Figure 2.10 Overview of measured absolute error between the fitness landscape metric for the EnergyPlus problem and surrogate model. While line represents median value.

Figure 2.10 provides an analysis of the fitness landscape metrics, presenting an overview by model type of the best performing models' absolute error values for all 10 building design problems. This error was evaluated by taking the difference between the model FLA metric and EnergyPlus FLA error metric. In general, all best performing models improved the reproduction of the original EnergyPlus fitness landscape when more training samples were provided. Taken as a whole, there seems to be some evidence for the ability of all models except MLP to reproduce the fitness landscape better when inputs are non-standardized. The FDC error metrics show an example of this behaviour. FDC represents the relationship of the input parameters to costs through evaluating the distance to the minimal cost. Here, GP and GBT performance is notably better when inputs are non-standardized, accurately reproducing the correlation of input parameters and costs in a way other models can't. MLP lags behind in its ability to correlate input parameters and cost values through the FDC metric in both standardized and non-standardized evaluations. Variance

of state is another method utilized to quantify the relationship between input parameters and costs. The analysis of GP error metrics reveals a wide range of values, although predominantly exhibiting a cluster of low error values signalling an accurate representation of the relationship has been captured. Notably, GBT demonstrates consistent variance of state error performance across the different categories of data. In so far as variance of cost metrics reflect the ability of the models to accurately reproduce the distribution of cost values, SVM and MLP models did better when a high number of training samples were provided.

The analysis of error values between the FLA metric of the model and the EnergyPlus data on a per-problem basis expands on the above findings (figure 2.11). For 104 training samples, non-standardized training data consistently demonstrates superior FDC performance. GP, SVM, and GBT models exhibit significantly lower FDC error values compared to the EnergyPlus problems, whereas the MLP model's effectiveness is limited by the smaller training dataset. The results suggest that standardized inputs lead to a smoother problem landscape. SVM stands out in terms of the variance of state and variance of cost FLA metrics, showcasing a tendency to overcompensate and produce landscapes with higher variance of state and reduced cost variance when inputs are non-standardized. Continuing the pattern identify above, GBT's performance in the variance of cost error metric deteriorates slightly when inputs are standardized. MLP also exhibits a decrease in variance of state error with non-standardized inputs but is better able to reproduce the distribution of cost values (when measured by variance) with standardized inputs. Interestingly, the non-linear characteristics of building design problems 4, 5, and 6 are evident in the variance of state error value, even though the only distinction among these buildings is their location. However, the specific difference between building 4 and buildings 5 and 6 remains unexplained. The per-problem FLA error metrics do not reveal any apparent reasons for the

challenges faced by the best performing models in accurately predicting cost values for building 8 (table 2.5).

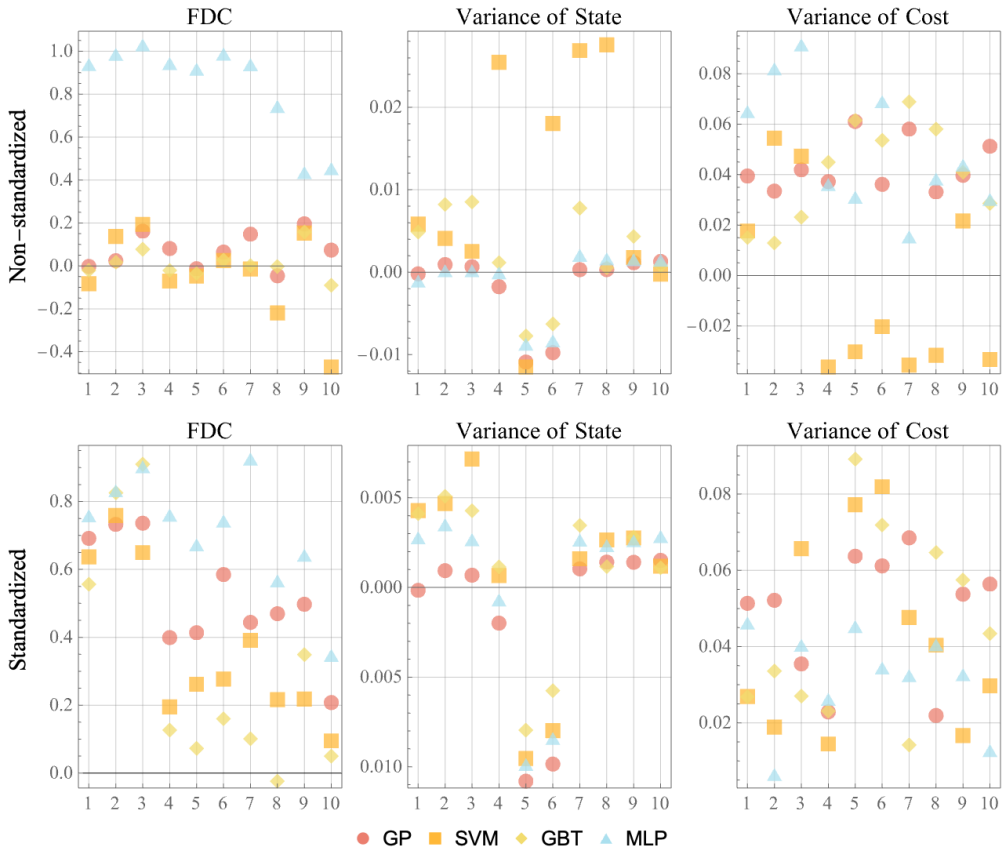

Figure 2.11 Error of best model's FLA value versus the EnergyPlus FLA metric per problem when N=104.

2.7 Conclusions

Surrogate modeling offers an efficient approach to simulating building energy performance and reduce computational burden. This allows designers to more efficiently explore the design problem space, whether it involves optimizing a new building or predicting the value of retrofit options for existing buildings. By analysing various building design problems and utilizing different categories of training data, this study's findings contribute to understanding surrogate modeling for building energy performance and inform model development. The results clearly demonstrate the value of standardization when using SVM or MLP. The use of multiple metrics for accuracy and fitness landscape analysis provides a more comprehensive and balanced view of model performance. Relying solely on one metric might be insufficient to fully capture all aspects of a model's

performance. In contrast, using multiple metrics gives a robust assessment which considers various aspects of a model's performance, thereby highlighting its strengths and weaknesses. This leads to a holistic evaluation of a model's ability to predict building energy performance.

The comparative analysis undertaken demonstrates that the GBT model consistently performs exceptionally well across all building design problems and training data categories. The model's superior performance stands out due to its high computational efficiency and ability to maintain accuracy across varying input conditions. During testing, the GBT model exhibited MAPE values ranging from 0.7 to 1.2% and R2 values ranging from 0.969 to 0.995. This level of performance is evident in the EnergyPlus fitness landscape metrics, where the GBT models achieved some of the lowest mean error values. GBT's documented accuracy when taking into account its computational performance (figure 2.12) show it to be a strong choice for surrogate modeling regardless of problem type or dataset.

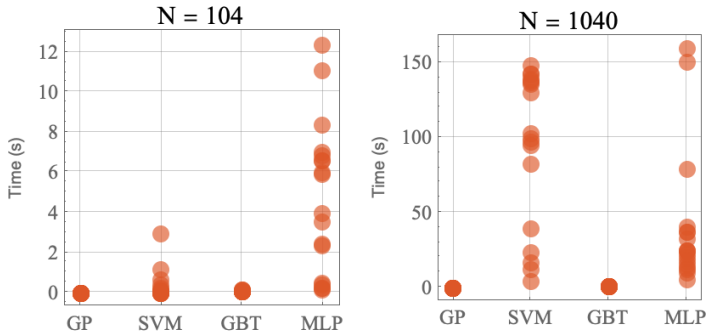

Figure 2.12 Timing in seconds to train model with shortest and longest training period removed.

One consideration for selecting GBT for surrogate modeling is that GBT performance will start to deteriorate as the number of input parameters increases. This threshold can vary depending on the dataset and therefore should be observed empirically for each application. If performance is seen to deteriorate when the number of input parameters is high and access to a large number of training samples is possible, shifting to a MLP network maybe beneficial.

2.8 Future work

Further investigation is warranted to determine the underlying cause of the observed lower FLA error metrics for non-standardized data. The notion that non-standardized data may lead to a more accurate reproduction of the original fitness landscape challenges the prevailing belief that standardized data yields more precise results. Future experiments will be conducted to explore this relationship and shed light on its implications.

Using the same methodology as outlined above, more building design problems should be developed for the BESOS platform. This expanded dataset has the potential to reveal further novel characteristics of model selection on the building design problems' fitness landscape. Better understanding of the relationship between a design problem's fitness landscape and performance can lead to better surrogate modeling.

Effort is also being put toward formatting the set of building design problems as a Python module for use on the BESOS platform and be made publicly available.

3 Chapter 3

A comparison of machine learning functions for time-series prediction in buildings

Birdsell, B. & Evins, R. (2023)
Submitted to Energy & Buildings journal.

3.1 Abstract

This paper presents a comparison of several different neural net architectures for predicting building performance in both short-term (day-ahead) and long-term (annual) scenarios. The study evaluates four types of neural networks: convolutional-based residual neural nets, recurrent neural networks, WaveNet, and hybrid residual-RNN. The paper provides detailed descriptions of the tested neural networks, and offers valuable insights for optimizing building performance, which can have significant benefits for building operation.

The findings highlight the strengths of convolutional layers in forecasting sequences, particularly for short-term predictions of building systems. Additionally, when estimating annual performance, convolutional-based networks demonstrate superior accuracy in predicting monthly minimum and maximum values. These results have important implications for decision-making in building operation, enhancing the overall efficiency and effectiveness of buildings.

3.2 Introduction

The goal of this project is to advance the understanding and implementation of machine learning techniques for predicting time series related to building performance. Like numerous studies in the field, it is fundamentally motivated by the fact that in North America buildings are responsible for

consuming 30-40% of all energy produced (U.S. Department of Energy 2012, Statistics Canada 2016). Therefore, improving building operations holds great potential to significantly reduce energy consumption and greenhouse gas emissions.

Characteristics inherent in building data pose challenges for time series prediction. It becomes crucial to understand these characteristics as they are closely linked to the structure of the data the model aims to learn and predict from. Notably, these data characteristics are similar to the characteristics with which traditional statistical methods struggle.

Long-term building time series data contain patterns of a diverse range of durations, including seasonal, weekly, and daily variations. The presence of high-frequency noise when data is recorded in high-resolution, such as when intervals are 15 minutes or shorter, presents significant challenges for modeling. Moreover, trends in the data often display non-linear relations, signifying that the associations between variables cannot be simply described as linearly proportional. This complexity is further heightened by the knowledge that systems can evolve in ways that surpass linearly proportional effects, implying the presence of dynamic behaviour.

Building owners and operators face challenges in effectively utilizing the growing amount of data generated by monitored buildings and the rapid evolution of machine learning algorithms. In order to address this, instead of developing new and intricate neural network architectures, this research examines previously established and widely recognized neural network architectures. The main objective is to compare the accuracy and computational performance of four neural network architectures under two prediction scenarios: day-ahead forecasting and long-term prediction. The insights gained from this study have the potential to offer valuable guidance for decision-making in applications of optimizing building performance.

3.3 Prediction & Forecasting

In this research, a robust comparison of the algorithms is sought, encompassing both day-ahead and annual predictions. A subtle distinction exists between the terms “forecast” and “prediction” within the context of this study. Nate Silver in his 2012 book “The Signal and the Noise” states that many fields use the terms “forecast” and “prediction” interchangeably. In this paper they are not used interchangeably. Here, forecasting will be taken to mean a short-term estimate, while prediction will refer to a longer-term aggregated estimate. To illustrate this distinction, consider the example of energy consumption. A forecast would look at the past week of data and estimate the next day’s usage in high-resolution. Prediction, meanwhile, would estimate the total energy expected to be consumed over the course of a year.

Machine learning algorithms for time series prediction in buildings find their primary applications in smart buildings and model predictive control. With the decreasing costs of computation and sensor deployment, modern building management systems are increasingly adopting predictive control as a valuable feature. Serale et al. (2018) documents several areas of improvement when smart buildings have better prediction capabilities. First, energy efficiency can be enhanced because of precise foreknowledge of energy consumption patterns. This allows intelligent building systems to optimize energy generation, storage, and distribution, resulting in less wasted energy. Second, occupant comfort and health are improved when intelligent building systems anticipate and regulate environmental conditions such as thermal, air quality, and visual lighting conditions accordingly. Third, the ability to accurately predict allows smart buildings to effectively schedule maintenance, reduce downtime, and optimize resource utilization. This leads to cost savings and lower operational costs. Last, Chen (2018) and Zhan (2022) highlight the correlation between model predictive control and the growing adoption of renewable energy sources in buildings. Given the substantial variation of renewable energy sources, it becomes

imperative for intelligent building systems to possess predictive adaptability, enabling them to proactively respond to changing conditions. Thus, the effective utilization of renewable resources hinges on the ability of these systems to predictively adapt.

Model Predictive Control (MPC) addresses several criteria that traditional control methods based on simulation cannot. MPC has the ability to handle parameter uncertainties and interdependencies, effectively tracking slow-moving processes that might be time delayed within the evolving system and capture nonlinear dynamics. MPC, moreover, enables simultaneous prediction of multiple objectives. The significant advantage of MPC lies in its ability to employ anticipatory control instead of corrective control, as emphasized by Afram (2014). According to their findings, MPC generally demonstrates superior performance in terms of lower energy consumption, improved transient response, resilience to disturbances, and consistent performance across varying conditions.

Turning to the benefits of better prediction algorithms in the long-term, the existing building stock offers a rich target in terms of potential retrofits. Building owners and operators have started to generate large volumes of data on the performance and behaviour of their assets over time (Gonzalez-Vidal et al. 2017). Prior work has employed recurrent neural networks and multivariate time series analysis to classify buildings (Baasch 2019). In more recent work, Deb et al. (2021) investigated the application of machine-learning techniques for achieving cost-optimal building retrofits.

3.4 Literature Review

This section presents an overview of time series prediction techniques. It begins with a description of statistical techniques that were employed prior to the widespread availability of high-performance computers. Then, the focus shifts to the development of the four specific neural networks that have been selected for examination in this study.

3.4.1 Approximating Functions

Classical prediction techniques for time series data, such as regression, differences, and the moving average of residual errors (ARIMA) as described by Box and Jenkins (1970) assume linear dependence over time and neglect the possibility of non-uniform causal relationships among variables. These methods often involve challenging model parameter selection and ad hoc adjustments. In contrast, Artificial Neural Networks (ANN) are black-box models composed of linear layers that imitate the structure of the human brain (Werbos 1974). ANNs do not require physical simulations of the building, instead relying solely on data to predict the target variable. Numerous studies have demonstrated the superior accuracy of ANN techniques compared to traditional ARIMA methods (Divina et al. 2019, Cai et al. 2019).

3.4.2 Recurrent Neural Networks

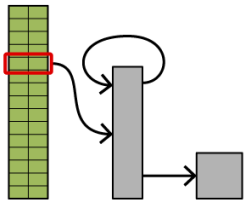

Figure 3.1 Schematic of RNN.

Recurrent neural networks (RNNs) have emerged as a popular choice for time series modeling, demonstrating state-of-the-art performance across diverse domains such as natural language processing, speech recognition, text generation, and language translation (Graves et al. 2013, Graves 2013, Sutskever 2014). Their design is tailored to effectively capture dependencies among consecutive input features, making them particularly suitable for analyzing time series data. The implementation of state memory within the RNN layer, which can be configured differently depending on the specific type of RNN layer used, enables RNNs to capture temporal dependencies with precision. During the training process, the sequence is incrementally fed into the network, allowing for the iterative updating of layer weights towards the desired target value. RNNs have been successfully employed in the domain of building performance and energy consumption by

Gao et al. (2019) and Sehovac et al. (2019), underscoring their efficacy in handling temporal building data.

3.4.3 Convolutional Based Residual Networks

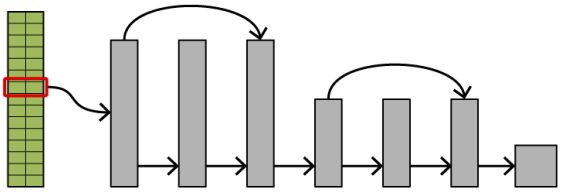

Figure 3.2 Schematic of CNN-based ResNet.

Convolution plays a crucial role in identifying latent patterns and structure within data by evaluating the product of a kernel applied to data. Convolutional layers are most commonly associated with spatial data like images, but, under certain assumptions, time series can exhibit spatial characteristics too. The increased use of convolutional layers has primarily been driven by advancements in computational power. The availability of increased computational power facilitates the repeated application of kernels, allowing for the iterative refinement of arrays of weights. This iterative process enables the approximation of intricate non-linear behaviours, contributing to the widespread adoption of this method.

In this project, convolutional layers are packaged into blocks allowing for skipped connections between layers that have been shown to improve accuracy on time series prediction and classification (He et al. 2016, Wang et al. 2019). In contrast to a linear network, which endeavours to learn the output function $H(x)$, the layers in a residual block are designed to learn the residual: $H(x) = F(x) - x$ (Choi et al. 2018, Benhaddi 2021). This design enables the neural network to effectively extract a greater amount of informative features from the given data.

3.4.4 Temporal Dilated Causal Convolutional Networks

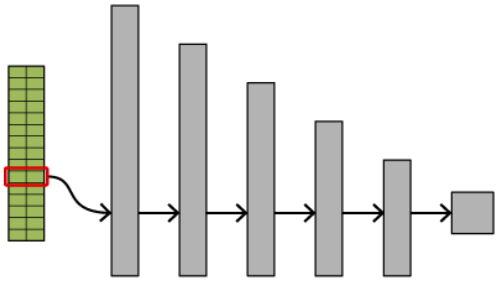

Figure 3.3 Schematic of WaveNet.

Similar to the success of convolutional neural networks in image processing, Temporal Dilated Causal Convolutional Networks (also called WaveNet) were developed to tackle challenges in audio processing. In the field of speech generation, Oord et al. (2016) introduced WaveNet which exhibited promising results. WaveNet effectively tackles the challenges posed by high-noise environments and extracts meaningful structures from data across different temporal scales. The communication of language offers an example of the challenge. It relies on the instantaneous movement of waveforms at small scales to understand legibly, while the expression of complex ideas necessitates the extraction of meaning from larger structures within the data such as sentences and paragraphs.

The WaveNet architecture employs convolutional layers with an expanding dilation parameter scheme, enabling an increased receptive field. Voss et al. (2018) successfully applied a WaveNet model to residential short-term load forecasting, outperforming linear regression methods and simple artificial neural networks. Building on this, Rueda et al. (2021) further extended the use of WaveNet models for electrical load forecasting in France. The authors emphasize that the expanding receptive window in WaveNets play a crucial role in handling challenges such as data sparsity, long-term sequence dependencies, and multi-resolution analysis.

3.4.5 Hybrid Neural Net

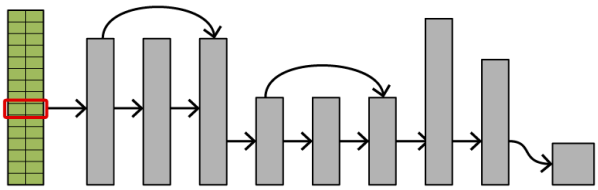

Figure 3.4 Schematic of Hybrid Convolutional-RNN network.

A hybrid model combining both Residual and RNN architectures is also considered. Previous studies by Goel et al. (2017) and Khan et al. (2020) have investigated the application of hybrid networks for time series analysis and electricity forecasting, with a specific focus on short-term predictions. In another project by Deb at el. (2021), a hybrid model was used to predict retrofit solutions using energy simulation data from different building retrofit solutions. The hybrid ResNet-RNN model demonstrated improved computational efficiency when predicting cost-optimized retrofit solutions versus simulating all potential designs.

3.5 Methodology

This section addresses how the experiments were conducted. It provides context for the dataset and describes the development of the methodology used to carry out the study and quantify the neural networks' performance.

3.5.1 Data Source

The data for this report was sourced from Building 59, a state-of-the-art smart building situated at Lawrence Berkeley National Laboratory (LBNL) in Berkeley, California. Completed in 2015, this building encompasses an area of 112,000 square feet and comprises two office floors, one mechanical equipment floor, and a dedicated floor for a data and computing centre. Only the datasets for the office floors were made available for analysis, thereby forming the basis for both annual and day-ahead predictions.

Data obtained directly from sensors or mechanical units presents challenges that can significantly impact the performance of machine learning applications if not appropriately

addressed. In the context of neural networks investigated in this study, one crucial requirement is the presence of uniform times steps in the input data. However, data collection gaps of different durations can occur due to various factors, including sensor or mechanical unit failures, maintenance activities, or other unforeseen circumstances, resulting in missing data spanning from hours to months. Moreover, building data often exhibits unpredictable transient events caused by fluctuations in user activity or the activation of large mechanical units. This can lead to the introduction of artifacts in the data which can affect analysis. To address these challenges, pre-processing techniques can be employed to mediate these artifacts to some extent, taking into account the specific requirements of the project at hand.

Building data was collected from January 2018 to December 2021. It was analyzed and prepared by LBNL, which involved data file aggregation and categorization, diagnostic gap filling, and outlier modification. Detailed information about the post-processing of the raw data can be found in the dataset's accompanying documentation provided by Lawrence Berkeley National Laboratory (2021).

3.5.2 Data Processing

In addition to the previously described data processing, the training and testing data underwent two further steps. After selecting which features would be used and targeted, the data was down sampled to 30-minute time steps. Next, the input features were standardized by subtracting the mean and dividing by the standard deviation of each feature. The prediction targets remained unchanged. To reduce the computational burden during the training phase, the annual dataset was further down sampled to a 1-hour time resolution.

3.5.3 Use Case

This study focuses on utilizing building data to predict the flow rate of four rooftop units that play a critical role in providing treated fresh air to occupants and represent an energy intensive

load. The prediction model employed in this research does not rely on any prior knowledge of physical building systems, but instead utilizes the roof top unit's previous state to estimate the predicted flow rate in the future. Neural networks are employed to create an approximating function that maps the input parameters to the flow rate at the subsequent timestep ($t + 1$). This predictive capability aligns with the goals of building management systems, which aim to forecast and control the RTU's future flow rate based on historical data. To assess the robustness of the day-ahead neural net architectures, four specific days were chosen, with each day being separated by a three-month interval. This approach allows for the evaluation of the day-ahead architecture under diverse environmental conditions and varying mechanical demands leading to inspection of how well the model configuration generalizes across different periods.

3.5.4 Neural Net Training

In this study, the treatment of the four rooftop units (RTUs) as individual features allowed for a multivariable input at each time step. In the annual version a fifth feature was introduced representing the month and fraction of the month. This addition to the annual training data allowed the neural network to more effectively match preceding years' data during training and enhance the accuracy of annual predictions. For day-ahead forecasting, a 72-hour window of 8 weeks of training data was utilized, with 30-minute timesteps. The output of each window corresponded to the subsequent 30 minutes of RTU flow rate values. In terms of the annual training data, it was segmented into 3-hour periods of hourly timesteps, encompassing the preceding 72 hours of data. This resulted in a target array consisting of 12 values for each of the 4 RTUs. Detailed representations of the day-ahead and annual model architectures used can be found in the appendix A of this work.

The training process was carried out on an Apple M1 Studio equipped with 32GB of RAM and operating macOS Ventura. The coding and construction of the neural network were conducted using Wolfram Mathematica 13.

Input features	Outputs
Roof top unit 1 standardized flow rate	Roof top unit 1 (CFM)
Roof top unit 2 standardized flow rate	Roof top unit 2 (CFM)
Roof top unit 3 standardized flow rate	Roof top unit 3 (CFM)
Roof top unit 4 standardized flow rate	Roof top unit 4 (CFM)
Month + Fraction of Month	

Table 3.1 List of model parameters.

3.5.5 Performance Metrics

This study utilizes Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as performance metrics for day-ahead and annual predictions. MAPE, commonly used in building performance studies, has been reported to range between 5-10% for CNNs and RNNs (Gonzalez-Vidal 2017, Koutlis 2020). Recent studies have achieved <4% absolute error and R-squared values ranging from 0.92 and 0.94 for short-term forecasting of heating loads (Eguizabal 2022). Performance values are obtained by averaging each MAE and MAPE across the four RTUs and seasons. R-squared, included for comparison purposes, is better suited for explanatory models and linear regression, whereas this study is aiming to empirically quantify the error to make its interpretation in engineering applications more straightforward. ASHRAE Guideline 14 allows for a maximum 20% variation in annual predictions, but its practical significance is questionable (A. S. Committee et al. 2002, Garrett 2014, Gonzalez-Vidal 2017). Annual predictions are evaluated by comparing estimated and historical monthly minimum and maximum values. Additionally, a naïve approach using a moving average based on previous historic values provides additional context. Short-term forecasting utilizes a 6-hour moving average, while the average of the preceding year serves as a naïve annual estimate for the testing year.

3.6 Results & Analysis

In the following section, the results are presented which detail the comparison of the four models under varying prediction conditions. The analysis aims to provide context and understanding of the models' performance, highlighting their respective strengths and weaknesses.

3.6.1 Day-ahead Forecast

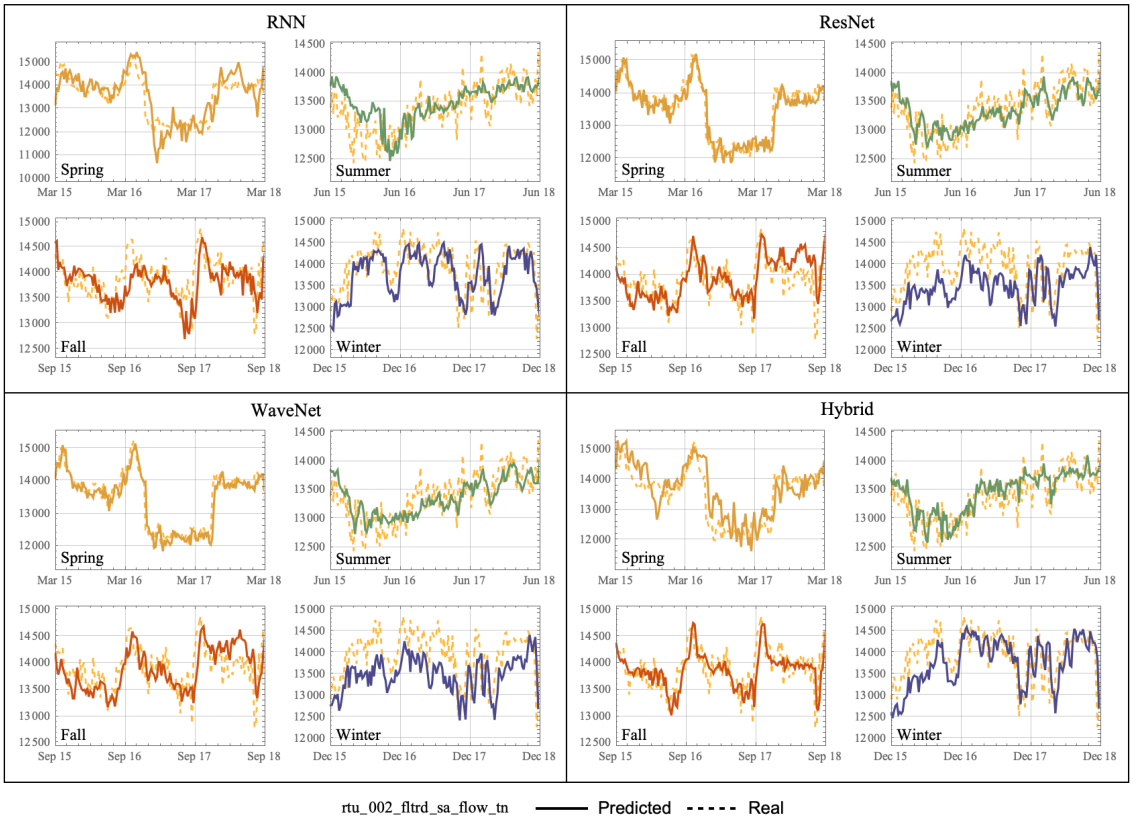

Figure 3.5 Day ahead results of RTU 2's air flow rate fluctuations and predictions for four target dates. All units in cubic feet per minute.

When the mean absolute percentage error (MAPE) and mean absolute error (MAE) values are averaged across the four RTUs and target dates, all four architectures demonstrate promising results, surpassing the performance of the naïve model. Upon predicting the next 24 hours' values, average MAPE ranges between 2.2% and 2.7%. During the test period, the ResNet and WaveNet models demonstrate an average deviation of less than 300 cubic feet per minute for any given hour. Considering that the air handling units of the case study building typically operate in the range of 12,000 to 16,000 cubic feet per minute, this marginal difference is negligible in terms of estimation

and design. When expanding to include a 72-hour outlook, except for the ResNet, the results show accuracy decreases as the forecast window extends past the 24-hour estimate and the model begins to encounter more data it's never seen before. This is realistic of real-world building applications where time series are continually added to over time and only ever see small periods of new data as the forecast window moves and model updated. The CNN-ResNet model consistently achieves an R-squared value of 0.95 for both the 24-hour and 72-hour windows. Both the ResNet and WaveNet architectures exhibited satisfactory performance in visually approximating the noise present in the actual building data. The winter target date posed the greatest challenge in terms of accurately capturing the noise pattern. Closer inspection of the raw data reveals that RTU 4 posed the most difficult target for training, consistently exhibiting higher error values across all tests.

	24-hour			**72-hour**		
	MAPE (%)	**MAE (CFM)**	**R2**	**MAPE** (%)	**MAE (CFM)**	**R2**
RNN	2.7	338	0.94	2.9	362	0.93
ResNet	2.4	301	0.95	2.4	302	0.95
WaveNet	2.2	281	0.96	2.3	295	0.95
Hybrid	2.4	309	0.95	2.5	312	0.95
Naive	4.9	603	0.83	5.4	676	0.75

Table 3.2 Day ahead forecast results summary.

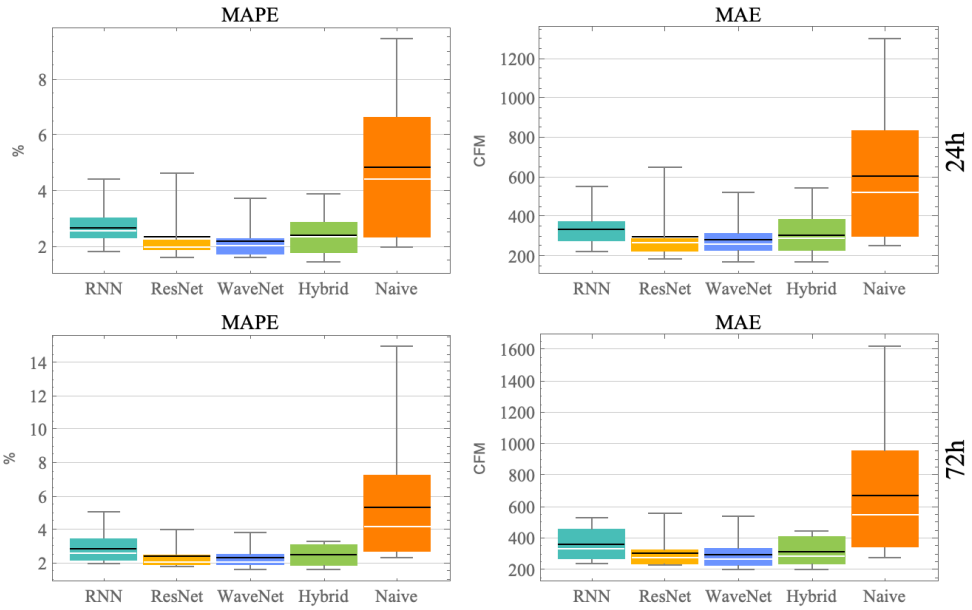

Figure 3.6 Aggregated 24-hour (top) and 72-hour (bottom) accuracy results.

The Box-Whisker diagrams presented in figure 3.6 provides an overview of the four RTU error values obtained for each of the four target dates, resulting in a total of 16 measurements for each model. A naïve model, represented by a 6-hour moving average, is included for comparison. These diagrams allow a direct comparison between models based on the aggregated MAPE and MAE values, as well as their variance. Notably, both ResNet and WaveNet models exhibit tight and consistent clusters of error values, with 75% of values falling within the coloured box. Conversely, the RNN and Hybrid models exhibit slightly greater variance for these error metrics across different RTUs and target dates. Furthermore, the naive approach demonstrated inferior performance relative to other models.

Figure 3.7 offers a clear and informative visualization of the R-squared value, with each model's 72-hour predicted values aggregated to evaluate this metric. The pair plots reveal distinctive clustering of the RTUs, with ResNet and WaveNet models exhibiting the most tightly compacted clusters along the diagonal. Clusters of each RTU are distinguished because of their different flow rate mean values and variance reflecting the differing capacity of the RTUs and requirements of airflow to specific areas of the building, the specifics of which were not a focus of this study.

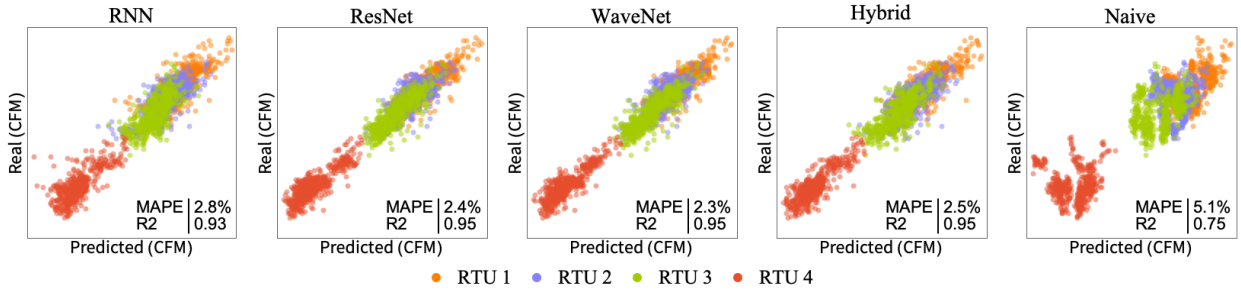

Figure 3.7 72-hour day-ahead pair plots of predicted vs real cubic feet per minute values.

3.6.2 Annual Prediction

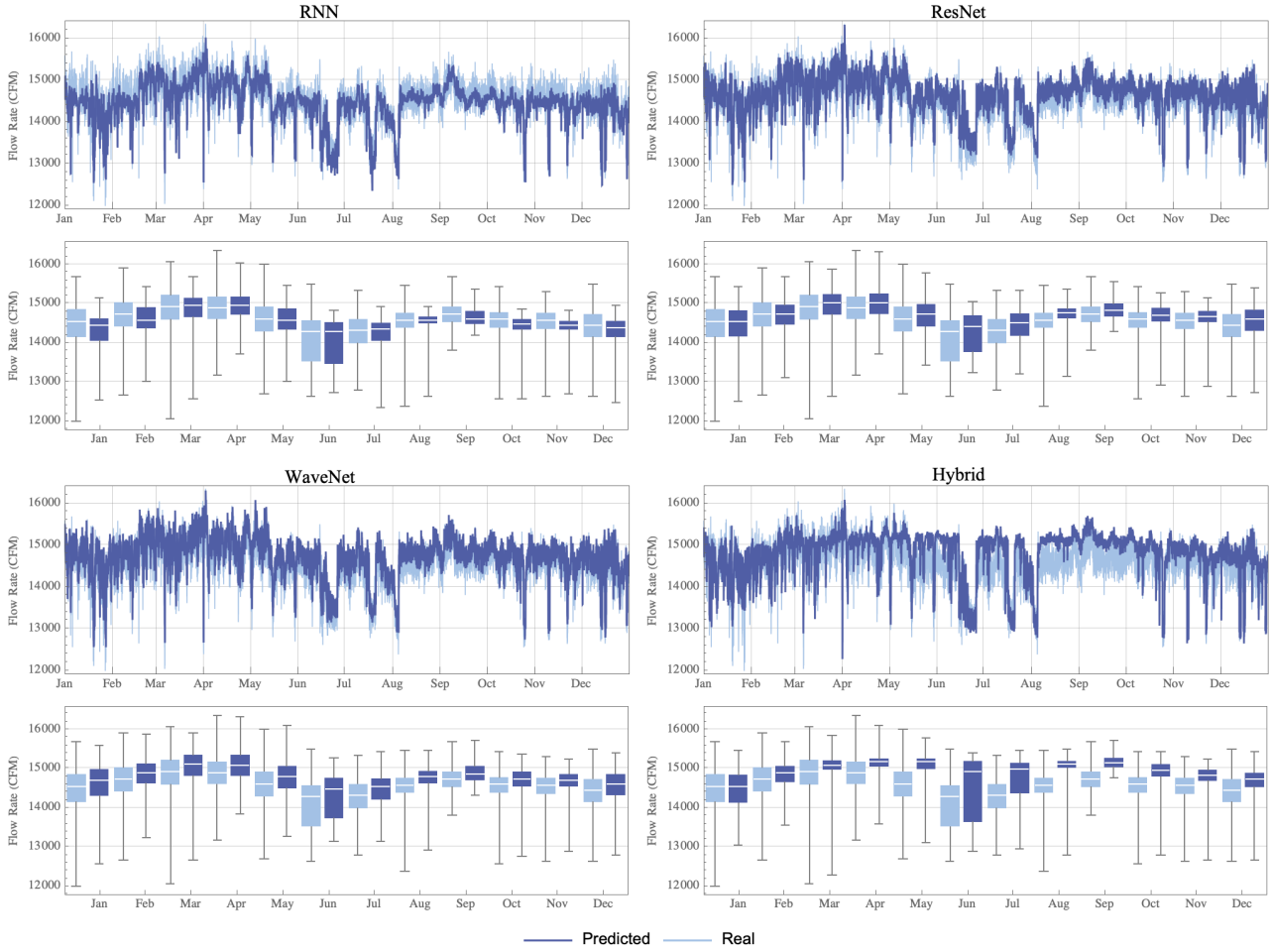

Figure 3.8 Annual historic versus predicted values for RTU 2 (top). Distribution of monthly values (bottom).

Figure 3.8 provides an overview of the annual results for RTU 2. By analyzing three years of hourly training data, it becomes evident that both the ResNet and WaveNet models demonstrate exceptional generalization capabilities when applied to the fourth-year testing data of which the models had never encountered before. These models successfully captured the high-frequency noise present in the original data. Furthermore, both nets did an admirable job predicting the distribution of monthly values and the monthly minimum and maximum values. Although the RNN model showed signs of convergence during training, the implementation of the early stopping method used in this study, aimed at preventing overfitting, led to predicted values that were smaller than the recorded historical values in the testing data. It is also noted that the Hybrid net tended to constrain values unrealistically. The naïve model's predicted annual values are not catastrophically

wrong as interpreted by the MAPE and MAE metrics, however, the R-squared value shows that when using the previous year's average, no meaningful part of the real time series historical pattern can be duplicated, and R-squared performance is measured to be worse than random.

	MAPE (%)	**MAE (CFM)**	**R2**
RNN	2.2	331	0.56
ResNet	1.8	264	0.70
WaveNet	1.9	283	0.67
Hybrid	2.7	406	0.36
Naive	4.6	695	-1.00

Table 3.3 Annual prediction results summary.

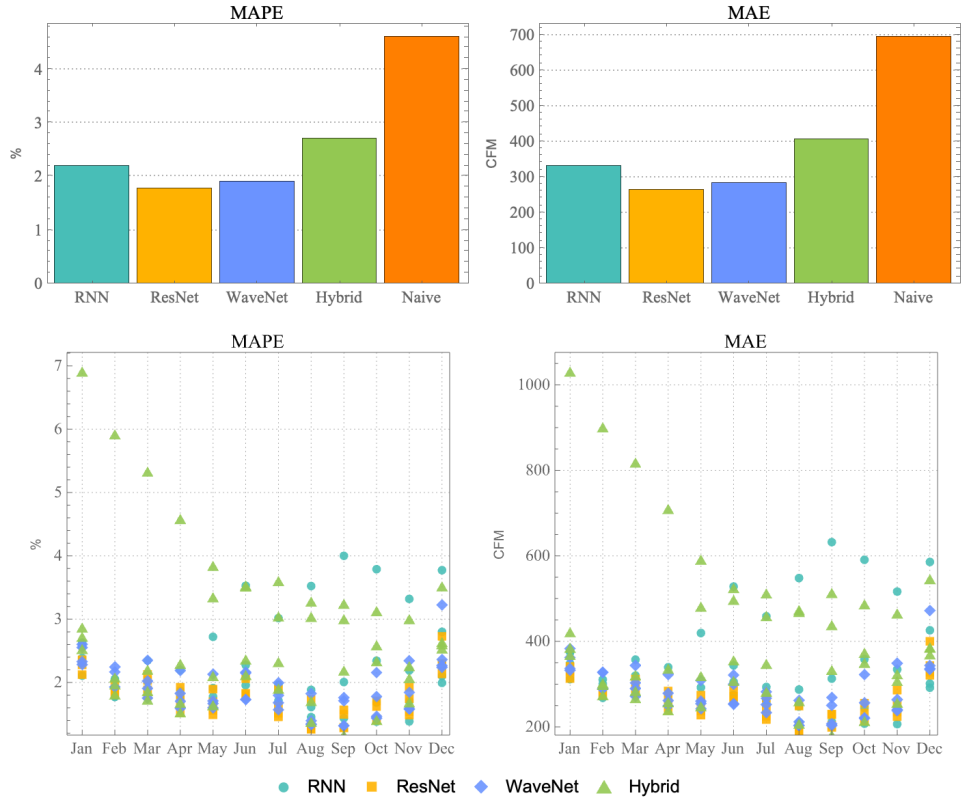

Figure 3.9 Top row shows averaged performance across all RTUs with the lower half representing each metric by month.

Analysis of annual results indicates that the performance of the RNN model closely trails behind that of the ResNet and WaveNet models. Details of deteriorating performance of the RNN and Hybrid model is seen in the lower portion of figure 3.9. The Hybrid model has poor accuracy throughout the year but especially RTU 4 in the months of January to May. Through the second half of the prediction period, it is the RNN model which struggles for accuracy on RTU 4's 2021

testing data, creating a cluster of high error values in the latter half of the year. For reference, the error calculated from the naive approach representing the mean value of the preceding year is included, resulting in an average MAPE error value of approximately 4.6% across all four RTUs.

Figure 3.10 provides a detailed representation of model performance by graphing the minimum and maximum absolute percent error from the actual minimum and maximum values of the RTU. Notably, the ResNet and WaveNet models, which exhibited the best performance overall, tend to overestimate the minimum value but demonstrate excellent predictive ability for the maximum value with a very low error rate when compared to the real monthly maximums. This latter estimate is particularly useful because accurate predictions of expected maximum demand are critical in calculating the design loads of mechanical systems. Conversely, the minimum and maximum absolute percent error of the RNN model is observed to correspond to the model's performance as depicted in figure 3.8, with the model underestimating values and therefore having a higher error when predicting the monthly maximum.

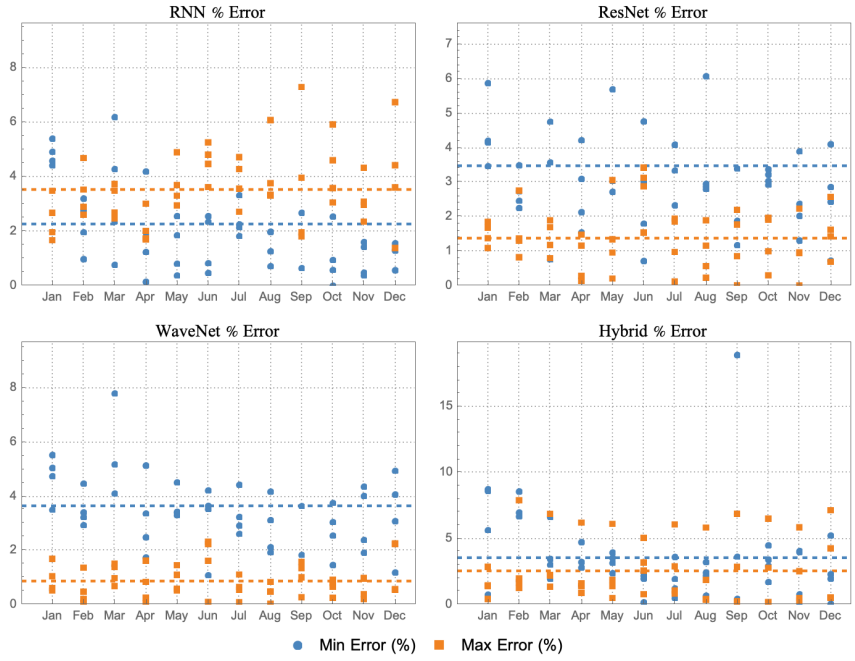

Figure 3.10 Annual min and max air flow rate error (%) summary by RTU and month with the dotted lines being the average of each.

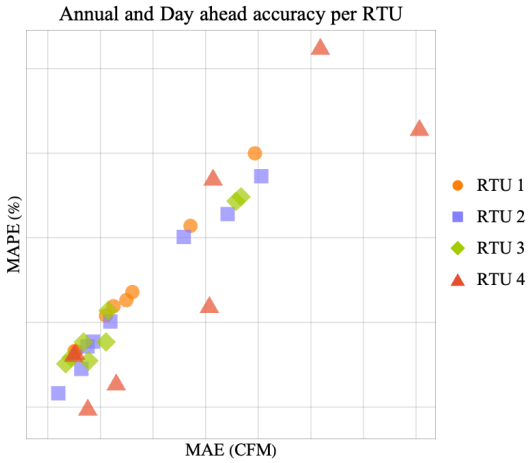

Figure 3.11 Pair plot of annual and day-ahead accuracy metrics showing RTU4 as an outlier.

A more detailed view of the lagging predictive performance of RTU 4 is presented in figure 3.11. When the day-ahead and annual MAE and MAPE values are combined and graphed as a pair plot, the first three RTUs are recognized as being clustered along the diagonal, with RTU 4 scattered off the diagonal with higher error values for each MAPE and MAE. To find a potential cause for this, it is worth revisiting the specifics of the training data from 2018 to 2020 in order to understand the observed behaviour (figure 3.12). The first three RTUs show excellent year-to-year consistency between the training data and 2021's testing data. However, RTU 4 presents a more complex behaviour for any one model to learn because of the unexpected rise in flow rate values in 2020/2021.

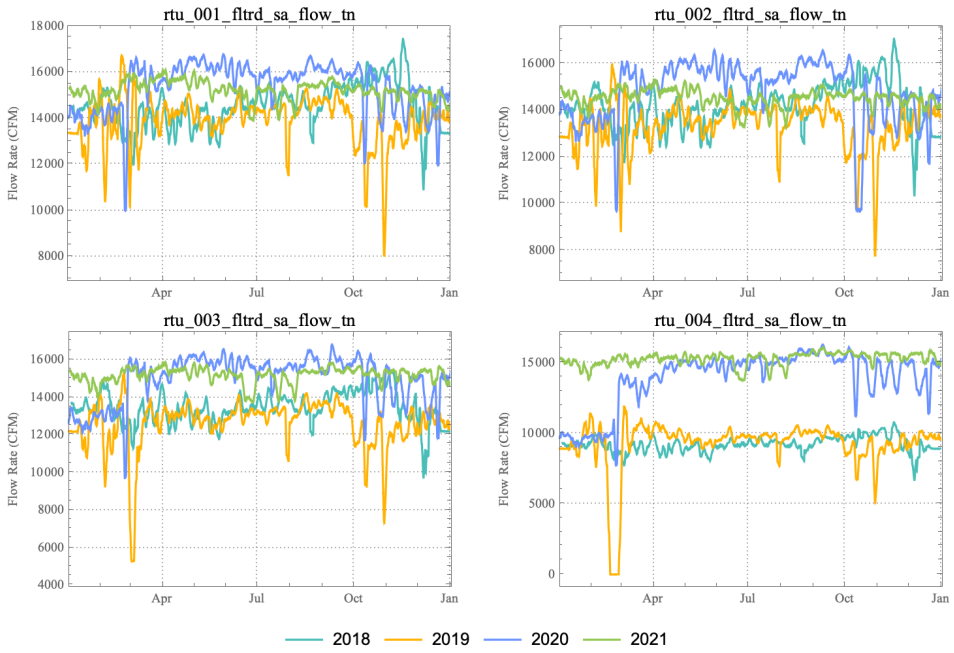

Figure 3.12 7-day moving average overlay of annual training and testing data

Annual predictions play a significant role in informing decision-makers and designers about the minimum, maximum, and average loads that building systems are expected to encounter throughout the year. Therefore, quantifying the similarity between predicted and actual distributions becomes vital for making informed design decisions. Jensen-Shannon Divergence (JSD), an effective yet underutilized tool for comparing distributions, has been previously used to assess the energy performance of building occupancy models (Mahdavi et al. 2016) and time series predictions in building energy simulations (Baasch et al. 2021). JSD is a symmetrical and smoothed version of the Kullback-Leibler Divergence and provides a way to compare the similarity or dissimilarity between two probability distributions. JSD yields lower values as distributions become increasingly similar, resulting in a value of zero if the distributions are identical. The ResNet and WaveNet models exhibit exceptional performance, closely approximating the historic distribution of flow rate values and achieving JSD values of less than 0.0089 when averaged across all RTUs (figure 3.13). This provides supporting evidence that the predicted minimum, maximum,

and mean results of the ResNet and WaveNet models are accurate. In contrast, the RNN and Hybrid models demonstrate unrealistic narrow peaks of values in the distribution histograms, representing a less accurate fit relative to the real data.

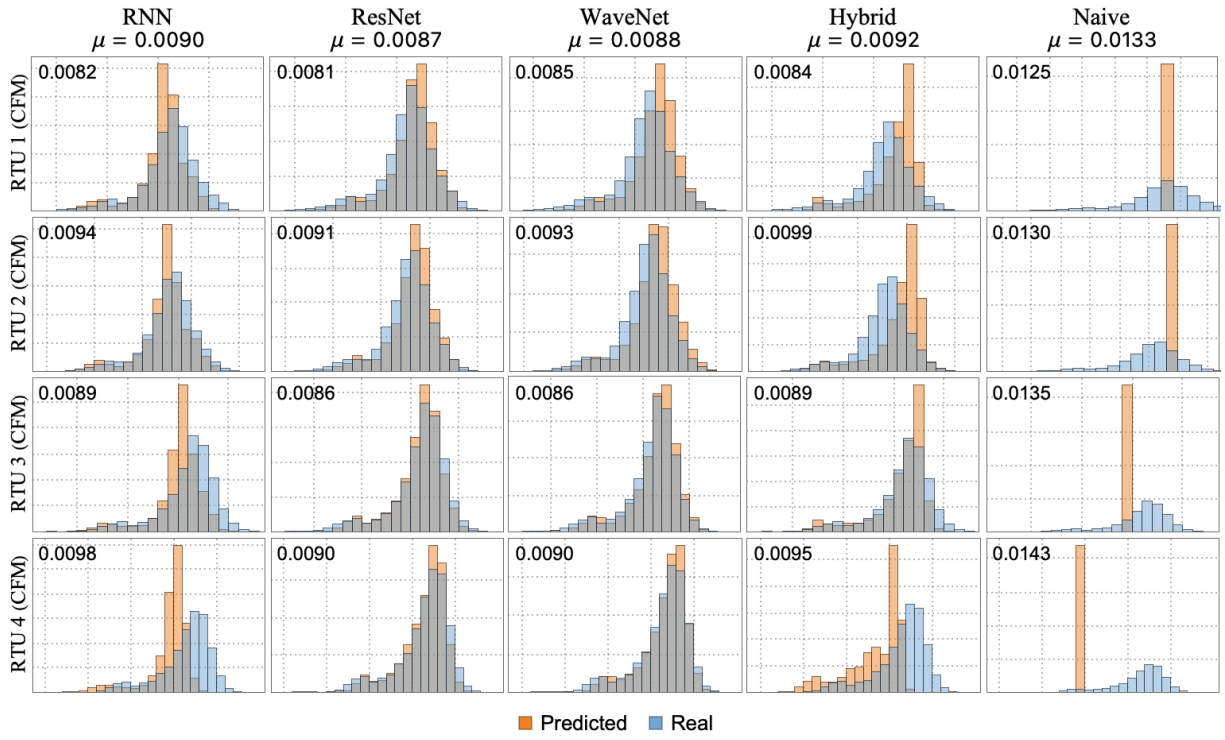

Figure 3.13 Jensen-Shannon Divergence. Lower values represent similar distribution.

3.7 Conclusion

This project evaluated the performance of ResNets, RNNs, WaveNet, and Hybrid architectures for model predictive control in building management systems and annual demand estimation. While ResNets and RNNs are well-established models in the field of time series modeling, this study expands upon these models by including WaveNet and Hybrid architectures. By considering a range of established and novel neural network models, a broader perspective on their applicability and effectiveness can be presented. This research provides a valuable description through analysis of the models' performance. The analysis includes stressing the predictive capabilities of the models by testing day-ahead forecasting throughout the year and examining the ability of models to predict a full year of values based on historical data from previous years. These conclusions

contribute to a better understanding of the suitability of these models for both short-term and long-term predictions, ultimately enhancing their potential application in optimizing building design and operations.

Notably, this study highlights the benefits of incorporating convolutional layers in ResNet and WaveNet models, which significantly improve their ability to forecast short-term sequences and estimate monthly values. Despite potentially slower convergence rates observed in CNN-ResNets, their exceptional capacity to extract latent structures from the data leads to more accurate predictions, offering a valuable trade-off between computational complexity and predictive accuracy.

Moreover, the findings emphasize the computational efficiency of lighter models such as RNN and WaveNet, which strike a balance between computational resources and satisfactory accuracy. Particularly in day-ahead forecasting tasks, these models demonstrate their suitability and effectiveness, as evidenced by the timing results presented in figure 3.14. It is worth noting that the size of the annual training data set has a significant impact on the training time of the Hybrid model, emphasizing the importance of considering computational resources and scalability when applying such models in real-world scenarios.

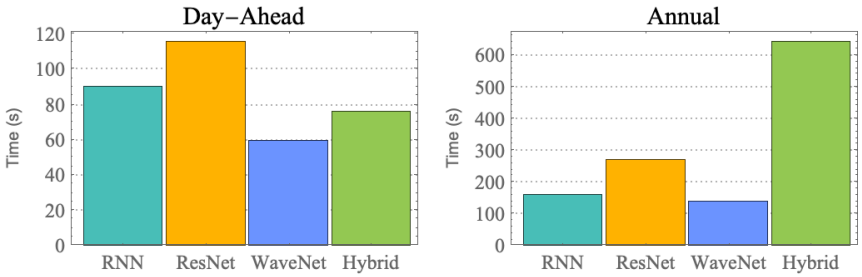

Figure 3.14 Timing in seconds to complete training of the model. Day-ahead results represent the averaged from four target dates.

This study showcases the impressive performance of the neural networks investigated, despite their reliance solely on data. However, it is crucial to acknowledge the importance of domain specific knowledge when applying these models to real-world scenarios. This is to minimize

potential errors in interpretation or application. This recommendation aligns with the research of Miller et al. (2020) who also emphasizes the value of a hybrid approach for the effective modeling of building data, combining data science expertise with engineering knowledge. By leveraging both the power of neural networks and domain-specific insights, the field of building performance prediction can benefit from improved accuracy and practical implementation.

3.8 Future Work

The present study has provided valuable insights into the application of different neural network architectures for predicting building performance, highlighting their ability to capture time series features at various scales. One limitation of this study is that it solely focuses on time series data from RTUs located on one building. In order to enhance the practical uses of these models in building performance applications, future research should concentrate on expanding the types of time series data and number of buildings. While broader than previous studies, expanding the types of data the models are expected to predict holds significant value.

Furthermore, in order to further enhance the comprehensiveness and applicability of this research, there is a compelling opportunity to explore and expand upon the findings to include other model types. Specifically, it would be beneficial to conduct a comparative analysis between neural networks and Gradient Boosted Trees, where the latter has in the author's experience exhibited exceptional performance across a range of data modeling problems while maintaining computational efficiency. By undertaking such a comparison, additional insights can be gained regarding the strengths and weaknesses of these two approaches, enabling a deeper understanding of their respective suitability and effectiveness in various contexts within the field of building performance.

4 Chapter 4

Conclusion

Data-driven decision-making for building performance entails the utilization of data analysis and predictive modeling to inform crucial decisions pertaining to building design and operation. This approach involves harnessing data from diverse sources, including sensor measurements, historical records, simulations, and experiments. The ultimate goal is to empower building designers and operators with the ability to optimize building performance through informed choices derived from data. This section addresses several essential points that relate to data-driven decision-making for building performance. These points inform best practice when using machine learning models in the process of making business decisions.

This work draws important conclusions regarding model selection that are only attainable with a comparative analysis of multiple models. The work supports the argument that neural networks are not the only option when trying to deliver accuracy predictions. For the data used in this study, other machine learning algorithms proved to be very performant and have lower computational costs. In this regard, Gradient Boosted Trees demonstrated excellent performance. It was shown to be robust against different data types and problem configurations, as well as being highly accurate and computationally efficient. For practical industry applications of building simulation or time-series prediction, Gradient Boosted Trees present a promising starting point. The research is also notable for highlighting the need for multiple metrics when evaluating model performance. Whether considering model accuracy or the problem's fitness landscape, no one metric

distinguished itself as the all-around best. Instead, different metrics need to be used in combination to draw out different performance characteristics.

A discussion about a potential future scenario under which all building performance models are expected to perform can now be addressed, and from which certain conclusions can be drawn based on the findings of this study. In the future, Canada can expect to see more extreme weather events. It is reasonable to ask how these models will perform when attempting to make predictions under conditions they have never encountered before such as the expected increased prevalence of heat domes. Some speculation is required to predict how the models will react when a new event is first encountered, however, as the model sees repeated extreme events, the characteristics of neural networks mean they have the ability learn shifting patterns in the data over time from repeated exposure, thereby incorporating these changes into their predictions. In the case of surrogate models, dealing with extreme weather events is more straightforward. It is addressed by updating the training data to include assumptions about the changing climate. Ultimately, predictive models are only one factor among many when discussing the hardening of buildings new and existing against climate change. In order to effectively tackle these challenges, new technological innovations in construction and materials will need to be incorporated into buildings for which the current project did not address.

In conclusion, this research documents the differences between machine learning algorithms, supporting further research into best practices when simulating or predicting building energy performance, and aids informed decision making when implementing machine learning in building design and operation.

5 References

A. S. Committee et al., (2002). *Ashrae guideline 14, measurement of energy and demand savings*. Atlanta.

Afram, A. & Janabi-Sharifi, F. (2014). *Theory and applications of HVAC control systems – A review of model predictive control (MPC)*, Building and environment, vol. 72, pp. 343-355.

Ascione, F., Bianco, N., De Stasio, C., Mauro, G. M., & Vanoli, G. P. (2017). *Artificial neural networks to predict energy performance and retrofit scenarios for any member of a building category: A novel approach*. Energy (Oxford), 118, 999–1017.

Baasch, G., Rousseau, G., & Evins, R. (2021). *A Conditional Generative adversarial Network for energy use in multiple buildings using scarce data*. Energy and AI, 5, 100087.

Baasch, G.M. and Evins, R., (2019). *Targeting Buildings for Energy Retrofit Using Recurrent Neural Networks with Multivariate Time Series*. In Neural Inf. Process. Syst.

Benhaddi, M. & Ouarzazi, J. (2021) *Multivariate Time Series Forecasting with Dilated Residual Convolutional Neural Networks for Urban Air Quality Prediction*, Arabian journal for science and engineering (2011), vol. 46, no. 4, pp. 3423-3442.

Cai, M., Pipattanasomporn, M. & Rahman, S. (2019), *Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques*, Applied energy, vol. 236, pp. 1078-1088.

Chen, B. J., & Chang, M. W. (2004). *Load forecasting using support vector machines: A study on EUNITE competition 2001*. IEEE transactions on power systems, 19(4), 1821-1830.

Chen, Y., Xu, P., Gu, J., Schmidt, F., & Li, W. (2018). *Measures to improve energy demand flexibility in buildings for demand response (DR): A review*. Energy and Buildings, 177, 125-139.

Choi, H., Ryu, S. & Kim, H. (2018), *Short-Term Load Forecasting based on ResNet and LSTM*, 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, pp. 1.

Deb, C., Dai, Z. & Schlueter, A. (2021), *A machine learning-based framework for cost-optimal building retrofit*, Applied energy, vol. 294, pp. 116990.

Divina, F., García Torres, M., Goméz Vela, F.A. & Vázquez Noguera, J.L. (2019). *A Comparative Study of Time Series Forecasting Methods for Short Term Electric Energy Consumption Prediction in Smart Buildings*, Energies (Basel), vol. 12, no. 10, pp. 1934.

Djuric, N., Novakovic, V., Holst, J., & Mitrovic, Z. (2007). *Optimization of energy consumption in buildings with hydronic heating systems considering thermal comfort by use of computer-based tools*. Energy and Buildings, 39(4), 471-477

Eguizabal, M., Garay-Martinez, R., & Flores-Abascal, I. (2022). *Simplified model for the short-term forecasting of heat loads in buildings*. Energy Reports, 8, 79-85.

EnergyPlus. (2022) Computer software. https://energyplus.net/ Vers. 22.2.0. US DOE Office of Energy Efficiency and Renewable Energy (EERE), Energy Efficiency Office. Building Technologies Office.

Friedman, J. H. (2001). *Greedy function approximation: a gradient boosting machine*. Annals of statistics, 1189-1232.

Gan, V. J., Lo, I. M., Ma, J., Tse, K. T., Cheng, J. C., & Chan, C. M. (2020). *Simulation optimisation towards energy efficient green buildings: Current status and future trends*. Journal of Cleaner Production, 254, 120012.

Gao, Y., Fang, C. & Ruan, Y. (2019). *A novel model for the prediction of long-term building energy demand: LSTM with Attention layer*, IOP conference series. Earth and environmental science, vol. 294, no. 1, pp. 12033.

Garrett, A. & New, J. (2016). *Suitability of ASHRAE guideline 14 metrics for calibration*. ASHRAE transactions, vol. 122, no. 1, pp. 469.

George E. P. Box, Jenkins, G. M., & Reinsel, G. C. (2008). *Time series analysis: Forecasting and control*, fourth edition. Wiley.

Goel, H., Melnyk, I. & Banerjee, A. (2017). *R2N2: Residual Recurrent Neural Networks for Multivariate Time Series Forecasting*, arXiv:1709.03159.

Gonzalez-Vidal, A., Ramallo-Gonzalez, A.P., Terroso-Saenz, F. & Skarmeta, A. (2017). *Data driven modeling for energy consumption prediction in smart buildings*, IEEE, pp. 4562.

Graves, A. (2013). *Generating Sequences With Recurrent Neural Networks*. arXiv:1308.0850v5

Graves, A., Mohamed, A. & Hinton, G. (2013). *Speech recognition with deep recurrent neural networks*, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. pp. 6645-6649.

Hastie, T., Tibshirani, R., & Friedman, J. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (Second Edition)*. Springer New York.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). *Deep Residual Learning for Image Recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770.

Hygh, J. S., DeCarolis, J. F., Hill, D. B., & Ranji Ranjithan, S. (2012). *Multivariate regression as an energy assessment tool in early building design*. Building and Environment, 57, 165–175. https://doi.org/10.1016/j.buildenv.2012.04.021

Kämpf, J. H., Wetter, M., & Robinson, D. (2010). *A comparison of global optimization algorithms with standard benchmark functions and real-world applications using EnergyPlus*. Journal of Building Performance Simulation, 3(2), 103–120.

Khan, Z.A., Hussain, T., Ullah, A., Rho, S., Lee, M. & Baik, S.W. (2020). *Towards Efficient Electricity Forecasting in Residential and Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework*, Sensors (Basel, Switzerland), vol. 20, no. 5, pp. 1399.

Kochenderfer, M. J., & Wheeler, T. A. (2019). *Algorithms for optimization*. Mit Press.

Koutlis, C., Papadopoulos, S., Schinas, M. & Kompatsiaris, I. (2020). *LAVARNET: Neural network modeling of causal variable relationships for multivariate time series forecasting*. Applied soft computing, vol. 96, pp. 106685.

Lawrence Berkeley National Laboratory (2021). *Dataset Description and Data Cleaning for Building 59 Raw Data: The Benchmark Datasets Project*.

Mahdavi, A., & Tahmasebi, F. (2016). *The deployment-dependence of occupancy-related models in building performance simulation*. Energy and Buildings, 117, 313-320.

Melo, A. P., Versage, R. S., Sawaya, G., & Lamberts, R. (2016). *A novel surrogate model to support building energy labelling system: A new approach to assess cooling energy demand in commercial buildings*. Energy and Buildings, 131, 233-247.

Miller, C., Arjunan, P., Kathirgamanathan, A., Fu, C., Roth, J., Park, J.Y., Balbach, C., Gowri, K., Nagy, Z., Fontanini, A.D., Haberl, J. & National Renewable Energy Lab. (NREL), Golden, CO

(United States) (2020) *The ASHRAE Great Energy Predictor III competition: Overview and results*. Science & technology for the built environment, vol. 26, no. 10, pp. 1427-1447.

Mohandes, M. (2002). *Support vector machines for short-term electrical load forecasting*. International Journal of Energy Research, 26(4), 335-345.

Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. (2016), *WaveNet: A Generative Model for Raw Audio*. arXiv:1609.03499

Osborne, M. A., Rogers, A., Roberts, S. J., Ramchurn, S. D., & Jennings, N. R. (2011). *Bayesian Gaussian process models for multi-sensor time series prediction*. In Bayesian Time Series Models (pp. 341–362). Cambridge University Press.

Papadopoulos, S., Azar, E., Woon, W. L., & Kontokosta, C. E. (2018). *Evaluation of tree-based ensemble learning algorithms for building energy performance estimation*. Journal of Building Performance Simulation, 11(3), 322-332.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.

Prada, A., Gasparella, A., & Baggio, P. (2018). *On the performance of meta-models in building design optimization*. Applied Energy, 225, 814–826.

Rueda, F.D., Suárez, J.D. & Alejandro del Real Torres (2021), *Short-Term Load Forecasting Using Encoder-Decoder WaveNet: Application to the French Grid*, Energies (Basel), vol. 14, no. 2524, pp. 2524.

Sapankevych, N. I., & Sankar, R. (2009). *Time series prediction using support vector machines: a survey*. IEEE computational intelligence magazine, 4(2), 24-38.

Sehovac, L., Nesen, C. & Grolinger, K. (2019). *Forecasting Building Energy Consumption with Deep Learning: A Sequence to Sequence Approach*, IEEE, , pp. 108.

Serale, G., Fiorentini, M., Capozzoli, A., Bernardini, D. & Bemporad, A. (2018). *Model Predictive Control (MPC) for Enhancing Building and HVAC System Energy Efficiency: Problem Formulation, Applications and Opportunities*, Energies (Basel), vol. 11, no. 3, pp. 631.

Silver, N., (2012). *The Signal and the Noise: Why So Many Predictions Fail—But Some Don't*. Penguin.

Statistics Canada. (2016). *Survey of Commercial and Institutional Energy Use*, 2014. Monthly energy review. The Daily. https://www150.statcan.gc.ca/n1/daily-quotidien/160916/dq160916c-eng.pdf Accessed Feburary 23 2022.

Sutskever, I., Vinyals, O. & Le, Q.V. (2014). *Sequence to Sequence Learning with Neural Networks*. arXiv:1409.3215.

Touzani, S., Granderson, J., & Fernandes, S. (2018). *Gradient boosting machine for modeling the energy consumption of commercial buildings*. Energy and Buildings, 158, 1533-1543.

U.S. Department of Energy (2012). *2011 Buildings Energy Data Book*. Buildings Technologies Program. Energy Efficiency and Renewable Energy. https://ieer.org/wp/wp-content/uploads/2012/03/DOE-2011-Buildings-Energy-DataBook-BEDB.pdf. Accessed February 23 2022.

U.S. Department of Energy (2021). *The Buildings Data Platform*. Office of Energy Efficiency and Renewable Energy's Building Technologies Office. https://bbd.labworks.org/ds/bbd/lbnlbldg59. Accessed February 23 2022.

Voss, M., Bender-Saebelkampf, C., & Albayrak, S. (2018). *Residential short-term load forecasting using convolutional neural networks*. 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) (pp. 1-6). IEEE.

Waibel, C., Mavromatidis, G., Evins, R., & Carmeliet, J. (2019). *A comparison of building energy optimization problems and mathematical test functions using static fitness landscape analysis*. Journal of Building Performance Simulation, 12(6), 789-811.

Waibel, C., Wortmann, T., Evins, R., & Carmeliet, J. (2019). *Building energy optimization: An extensive benchmark of global search algorithms*. Energy and Buildings, 187, 218–240

Werbos, P. (1974). *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA.

Westermann, P., & Evins, R. (2019). *Surrogate modelling for sustainable building design–A review*. Energy and Buildings, 198, 170-186.

Westermann, P., Christiaanse, T. V., Beckett, W., Kovacs, P., & Evins, R. (2021). *besos: Building and Energy Simulation, Optimization and Surrogate Modelling*. Journal of Open Source Software, 6(60), 2677.

Wetter, M., & Wright, J. (2004). *A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization*. Building and environment, 39(8), 989-999.

6 Appendix

6.1 Appendix A

This section documents the details of the neural nets used in this work.

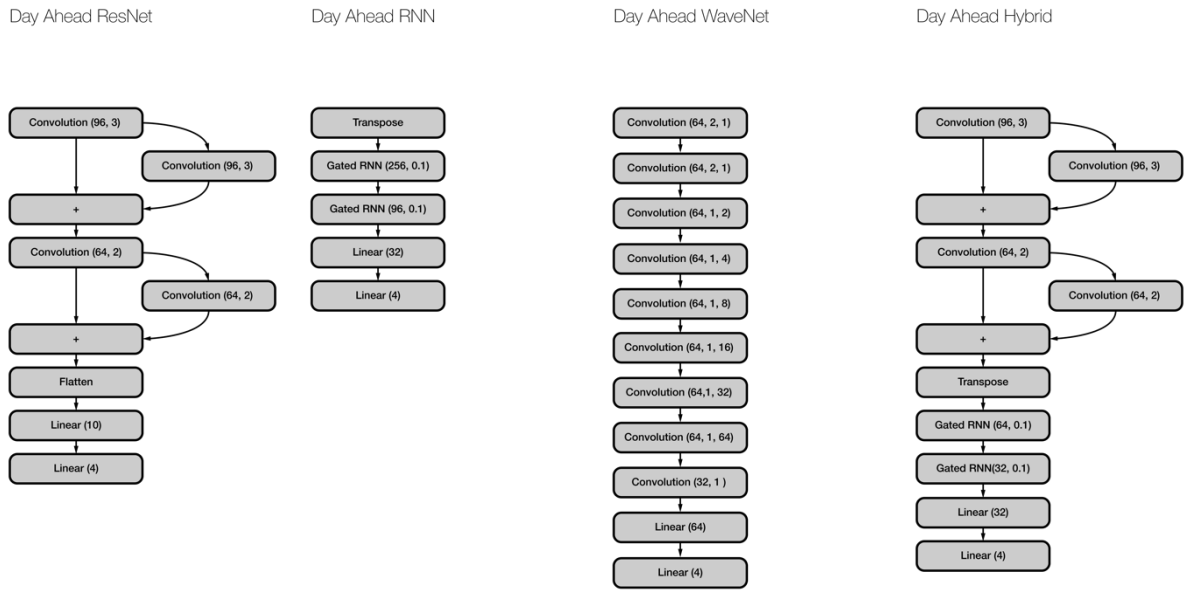

Figure 6.1 Detailed representation of day-ahead neural nets.

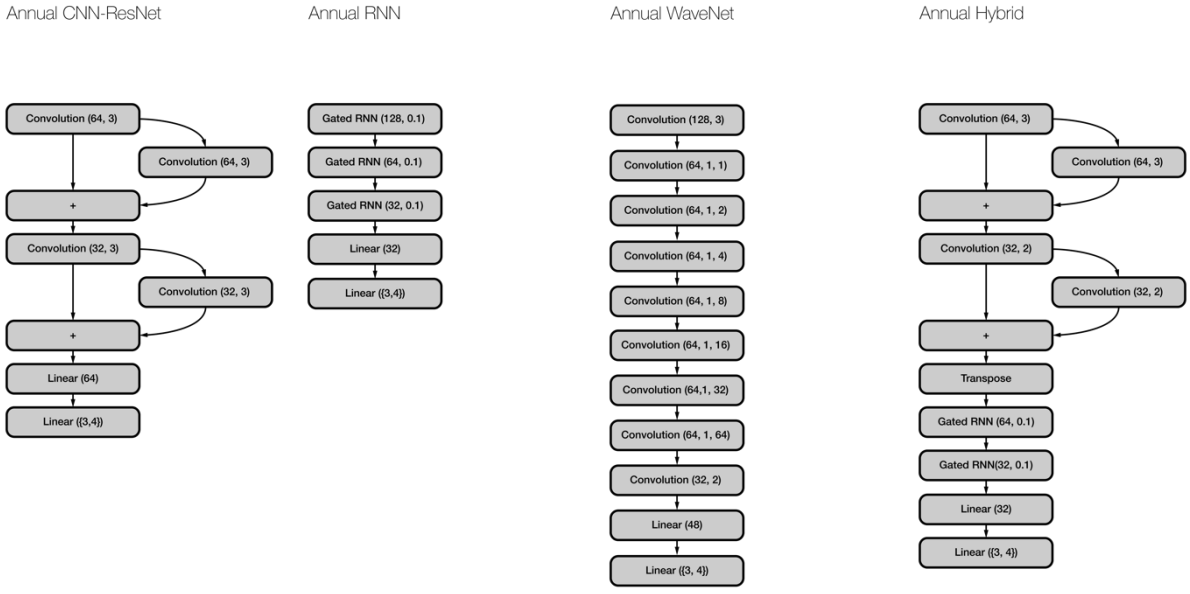

Figure 6.2 Detailed representation of annual neural nets.

6.2 Appendix B

Surrogate Modeling performance for building design problems

Birdsell, B. & Evins, R. (2023)
Accepted to 18th IBPSA International Conference and Exhibition. Shanghai, China.

Abstract

Simulating building energy performance is a key part of design. Surrogate modeling has emerged as a method to reduce the computational load of simulating potential building designs. To better understand the relationship between problem characteristics and surrogate model performance, this study investigated two building design problems and four types of surrogate models. The study documents the strong performance of gradient boosted decision trees in predicting energy use and representing the original fitness landscape of the simulated data. These findings provide insight into the relationship between problem characteristics and surrogate model performance, and serve as a foundation for future research into surrogate modeling for building energy performance.

Introduction

Building energy simulations predict the energy and thermal performance of a building, making them essential to evaluate its performance during the design phase. These simulations help improve building design by identifying energy efficiency measures, reducing energy consumption, lowering carbon emissions, and improving occupant comfort (Gan et al. 2020). They provide valuable information for building owners and operators, who can use this data to make informed decisions about energy management, identify retrofit opportunities, and improve building energy performance (Westermann et al. 2019). However, the high computational demands of physics-based building energy simulations are a significant impediment to their widespread implementation. Surrogate modelling (SM), which is an application of machine learning, can greatly reduce the computational load of predicting building performance. This paper aims to investigate the accuracy and changes in fitness landscape associated with surrogate modelling and its causes on each design problem. The results will enable decision-makers to balance performance and accuracy in their surrogate models effectively.

The BESOS platform

The BESOS platform is a comprehensive tool that offers a suite of integrated modules for building energy simulation, optimization, and surrogate modeling. It enables architects, engineers, and other building professionals to model building energy performance with a high degree of accuracy, while reducing computational load. The platform is flexible, modular, and offers a range of different optimization algorithms, allowing users to find the optimal design solution for their specific building project. The BESOS platform represents a significant advancement in the field of building energy simulation and optimization, with applications in building design, construction, and operation, because it allows programmatic interaction via Python with EnergyPlus and EnergyHub, facilitating the building optimization and the generation of training and testing data used in machine learning applications. A detailed description of the platform's development and capabilities can be found in Westermann et al. (2021).

Complimenting the results of this study, a goal of this project was to extend the functionality of BESOS to include a set of building design problems. This standard set of problems opens research to novel machine learning and optimization approaches where the use of physics-based simulations could be an obstacle.

The entire workflow of generating building simulation data, conducting surrogate modelling, and analysing the results was accomplished on the BESOS platform, utilizing Jupyter notebooks. Final analysis and graphics were produced in Wolfram Mathematica 13.

Related Work

This section expands on the algorithms chosen to model the EnergyPlus building simulations, which were implemented in Python's scikit-learn module (Pedregosa et al. 2011).

Gaussian Processes

To understand Gaussian processes (GP), a common starting point is linear regression, which models the relationship between a dependent variable and one or more independent variables. Gaussian processes, in contrast, assume the error is a Gaussian distribution with a linear combination of parameters, allowing for greater flexibility in modeling complex relationships between variables (Kochenderfer 2019). To address the design problems encountered in building simulations, ridge regression with an *alpha* parameter was an appropriate choice for the Gaussian process model, as each variable has a measurable effect on the output. The use of Gaussian processes has been elaborated on in multi-sensor time series prediction, relevant to the types of sensor systems found in modern buildings, by Osborne et al. (2010).

Melo et al. (2016) further extended the application of Gaussian processes to model cooling demand for several example buildings.

Support Vector Machines

Support vector machines (SVM) are widely recognized for their superior classification performance. However, SVMs are also applicable to regression problems such as time series prediction. The main mechanism at work in SVMs involves utilizing a kernel function to transform the input data points from their original feature space to a higher-dimensional feature space. This kernel function calculates the similarity or distance between pairs of input data points, enabling the SVM to implicitly operate in the higher-dimensional space without explicitly computing the transformed feature vectors. By doing so, the SVM bypasses the computational burden associated with explicitly calculating the transformed features, thus enhancing computational efficiency. In their 2009 survey, Sapankevych describes the use of SVMs in predicting financial data time series and environmental parameters such as air quality. Furthermore, the performance of SVMs in forecasting energy loads has been extensively studied, as documented by Mohandes (2002) and Chen et al. (2004). For this study, a radial basis function was chosen as the SVM kernel because of their ability to capture non-linear behaviour. When used as a kernel in SVM, they can produce excellent results. An example of this is in predicting the cooling demand of buildings with a mean absolute percent error of 4.0% (Melo et al. 2016). During testing, the radial basis function's *gamma* parameter can be adjusted. This parameter represents how distance to neighbouring points influences the approximation.

Gradient Boosted Trees

Decision trees are a popular choice for prediction models due to their human-like reasoning and interpretability. Random Forest and Gradient Boosted Trees are two prominent types of decision trees. In the random forest model, multiple agents build a decision tree, and the results are averaged at the end. Gradient boosted trees (GBT), however, combine results from the start, leading to better computational efficiency and accuracy (Friedman 2001). This study employed gradient boosted trees as the method of analysis since the training and testing data were expected to be free of random noise, which would have required averaging in a random forest model. The depth of the tree and the number of estimators were the primary parameters examined in this study, as they significantly influence the model's performance. The tree depth determines the model's ability to learn complex data relationships, while the number of estimators determines the number of boosting rounds used to improve model accuracy. Studies by Papadopoulos et al. (2018) and Touzani et al. (2018) provide strong evidence of the GBT model's ability to estimate building energy performance.

Multi-Level Perceptron

A multilayer perceptron (MLP) is an artificial neural network composed of linear layers that mimics the structure of the human brain (Werbos 1974). MLP models are well-suited for solving problems with non-linear characteristics, and their performance improves with an increasing amount of data due to their adaptability. In contrast, GP models have a limited set of static parameters, and their performance does not improve significantly with increased training data. The performance of an MLP model depends on the various configurations of layers, including the number of hidden layers and the width of each layer, which can affect the model's ability to learn data characteristics. Goel et al. (2017) and Khan et al. (2020) have each looked at neural nets' efficacy at predicting energy consumption. Deb et al. (2021) used neural nets to predict cost-optimized retrofit solutions. Here the results show the neural net model was able to provide retrofit solutions with greater computational efficiency than if all potential designs were simulated.

Methodology

The process of surrogate modeling entails the use of training data to develop an approximate model. Given the well-established impact of sample size on model performance, two distinct categories of training data were considered. Specifically, 130 and 1300 input samples were selected for each design problem and subsequently simulated in EnergyPlus 9.1.2. When standardizing the inputs, the values were shifted and rescaled to achieve a zero mean and unit sample variance. As such, each model, with varying parameters, was trained on four categories of data to acquire knowledge of the cost function. These categories included a large and small data set, with both standardized and non-standardized inputs. The models were trained on 104 and 1040 samples, respectively, and tested against 26 and 260 samples.

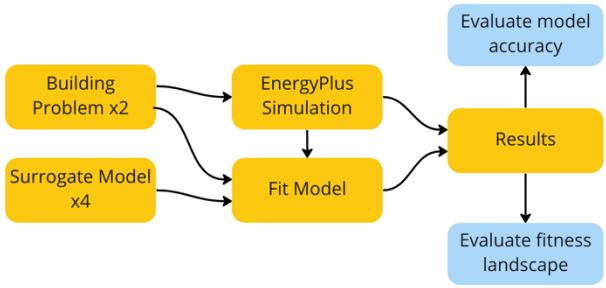

Figure 3. Overview of surrogate modeling methodology.

The process of analyzing the results from EnergyPlus and the SM for each building design problem is represented in figure 3. The returned normalized values of each of the chosen accuracy metrics makes comparison between building design problems possible where the absolute cost errors might be orders of magnitude different between problems. The mean absolute percent error subtracts the predicted cost value from the real EnergyPlus cost value and then divides the results by the real cost value to get the error as a percent. The R-squared value divides the variance of the predicted values by the variance of the real values, resulting in a ratio of how much of the variance in the results is explained by the model.

Taking direction from Waibel et al. (2019a), three measurements of fitness landscape were evaluated on each of the 49 models: Fitness distance correlation, variance of state, and variance of cost. Based on sampling the model, fitness landscape analysis through evaluation gives insight into the nature of the problem surface. Each is briefly described below.

Fitness Landscape Analysis

Fitness landscape analysis (FLA) aims to quantify the surface characteristics of design problems in terms of their input-output relation, specifically the patterns that arise from the decision variables x in relation to the cost value f(x). Fitness refers to the cost value and Landscape refers to the multi-dimensional surface which is defined by the unknown cost-response f(x) to the input decision variables x (Waibel et al. 2019a). Problem landscapes deemed difficult can be more challenging to optimize.

Fitness Distance Correlation (FDC) is a well-known measure of problem difficulty. The evaluation starts by defining the input values of the minimal cost and then calculating the Euclidean distance of various samples to that minimum. More challenging surfaces will show less correlation between the inputs, where as highly correlated surfaces will show similar inputs have similar costs and approach an FDC value of one.

Variance of state is normalized to make comparison across problems possible. The evaluation begins by categorizing or binning the cost values and then analyzing the variance of the inputs in each bin. Smoother, less challenging surfaces will have closely related input values if they are in the same bin, whereas when high variance is detected within a bin, it means that similar cost values can have widely varying inputs.

Lastly, a more familiar evaluation of the straightforward reporting of the variance of the cost values is given, normalized again so that problems of different magnitude can be similarly compared.

Review of building design problems

The initial set of building design problems was from Waibel et al. (2019a) with the original sources cited in the table below.

	N	Variables	Building Type	Source
Building A	13	continuous	Office	Wetter et al. (2004)
Building B	10	continuous	School	Djuric et al. (2007)

Table 1. Summary of Building design problems.

Building design problem A

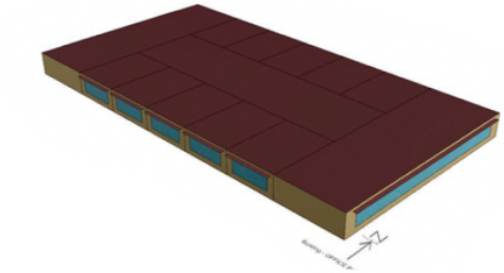

Figure 1. Representation of Building A.

Building design problem A is an office located in Houston (figure 1) comprising of two large zones in the East and West direction, along with five smaller zones oriented towards the North and South. The floors and ceilings are considered adiabatic, and the 13 decision variables include window widths, overhang depths, external shading setpoints, setpoints for the zone air temperature, and the cooling design supply air temperature for the HVAC system.

The aim of these tasks is to minimize the annual primary energy consumption, expressed in kWh/m^2, for heating and cooling coils, fans, and zone lighting. The objective function is defined as

$$f(x) = \frac{\left[PEG_{el}\left(E_{el}(x) + E_c(x)\right) + PEF_{gas}\, E_h(x)\right]}{A}$$

where E_{el} represents energy consumption for fans and zone lighting, E_h denotes heating coil energy consumption, and E_c represents cooling coil energy consumption. PEF_{el}=3 and PEF_{gas}=1 signify primary energy factors for electricity and gas, respectively. All of which is divided by the building's area. (Waibel et al. 2019b)

Building design problem B

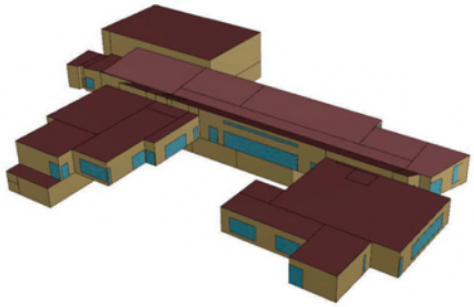

Figure 2. Representation of Building B.

Building design problem B focuses on cost optimization of energy, insulation, and radiator sizing for a school building in Belgrade, Serbia. The building model includes nine thermal zones, and each zone has a radiator that can be optimized. In addition, the insulation thickness of external walls is a decision variable. The objective function aims to minimize the total cost in € for heating energy (CE), insulation (C_{ins}), and radiator (C_{rad}), and is represented as follows:

$$f(x) = CE(x) + C_{in}(x) + C_{rad}(x)$$

Results & Interpretation

In this section a summary of the results will make way for a more in-depth analysis of the error and fitness landscape of each building design problem.

Summary of Results

Table 2 of 1040 training samples shows how the SVM and MLP models struggle with non-standardized inputs. Even with standardized inputs, there are still model parameter configurations whose performance was equal or worse than random. GP, while not leading in overall performance, did improve marginally with the

	Non–Staandardized Error Values	Building A		Building B	
		R2	MAPE	R2	MAPE
Gaussian Process	alpha: 0.0	0.5910	5.8244	0.9325	1.8973
	alpha: 0.0714	0.5910	5.8243	0.9310	1.8851
	alpha: 0.1429	0.5910	5.8243	0.9292	1.8754
	alpha: 0.2143	0.5909	5.8242	0.9273	1.8671
	alpha: 0.2857	0.5909	5.8242	0.9251	1.8619
	alpha: 0.3571	0.5909	5.8241	0.9228	1.8579
	alpha: 0.4286	0.5909	5.8241	0.9203	1.8561
	alpha: 0.5	0.5908	5.8240	0.9175	1.8550
	alpha: 0.5714	0.5908	5.8240	0.9146	1.8554
	alpha: 0.6429	0.5908	5.8239	0.9115	1.8570
	alpha: 0.7143	0.5907	5.8239	0.9083	1.8609
	alpha: 0.7857	0.5907	5.8239	0.9048	1.8679
	alpha: 0.8571	0.5907	5.8238	0.9012	1.8760
	alpha: 0.9286	0.5906	5.8238	0.8975	1.8856
	alpha: 1.0	0.5906	5.8237	0.8936	1.8953
Support Vector Machine	gamma: 5e–05	0.1669	10.0000	–0.1000	10.0000
	gamma: 0.0001	–0.1000	10.0000	–0.1000	8.6416
	gamma: 0.0005	–0.1000	10.0000	–0.1000	6.4344
	gamma: 0.001	–0.1000	10.0000	–0.1000	6.5472
	gamma: 0.005	–0.1000	10.0000	–0.1000	6.5944
	gamma: 0.01	–0.1000	10.0000	–0.1000	6.5942
	gamma: 0.05	–0.1000	10.0000	0	6.5942
	gamma: 0.1	–0.1000	10.0000	0	6.5942
	gamma: 0.5	0	10.0000	0	6.5942
	gamma: 1.0	0	10.0000	0	6.5942
Gradient Boosted Trees	depth: 5 agents: 25	0.8483	2.9419	0.5870	2.9690
	depth: 5 agents: 50	0.9849	1.1549	0.9046	1.8342
	depth: 5 agents: 60	0.9897	0.9829	0.9289	1.6422
	depth: 5 agents: 80	0.9929	0.8358	0.9496	1.4294
	depth: 5 agents: 100	0.9938	0.7875	0.9595	1.2997
	depth: 5 agents: 100	0.9946	0.7044	0.9688	1.1770
	depth: 5 agents: 125	0.9944	0.7446	0.9656	1.2113
	depth: 5 agents: 150	0.9947	0.7214	0.9688	1.1607
	depth: 3 agents: 100	0.9923	0.8727	0.9361	1.6121
	depth: 10 agents: 25	0.8598	2.8997	0.6683	2.8159
	depth: 10 agents: 50	0.9812	1.2772	0.8770	2.1483
	depth: 15 agents: 25	0.8556	2.9480	0.6375	2.9826
	depth: 30 agents: 25	0.8566	2.9261	0.6294	3.0081
Multilayer Perceptron	(64,)	–0.1000	10.0000	–0.1000	6.7339
	(100,)	–0.1000	10.0000	–0.1000	6.7414
	(150,)	–0.1000	10.0000	–0.1000	6.8058
	(64, 64)	–0.1000	10.0000	–0.1000	6.7261
	(100, 100)	–0.1000	10.0000	–0.1000	6.8667
	(25, 25, 25)	–0.1000	10.0000	–0.1000	6.7916
	(64, 64, 64)	–0.1000	10.0000	–0.1000	6.7838
	(96, 96, 96)	–0.1000	10.0000	–0.1000	6.8095
	(25, 25, 25, 25)	–0.1000	10.0000	–0.1000	6.8016
	(64, 64, 64, 64)	–0.1000	10.0000	–0.1000	6.8156

	Standardized Error Values	Building A		Building B	
		R2	MAPE	R2	MAPE
Gaussian Process	alpha: 0.0	0.6644	5.9223	0.9223	1.9771
	alpha: 0.0714	0.6643	5.9221	0.9223	1.9770
	alpha: 0.1429	0.6643	5.9219	0.9223	1.9769
	alpha: 0.2143	0.6643	5.9217	0.9222	1.9768
	alpha: 0.2857	0.6642	5.9215	0.9222	1.9766
	alpha: 0.3571	0.6642	5.9213	0.9222	1.9765
	alpha: 0.4286	0.6641	5.9211	0.9222	1.9764
	alpha: 0.5	0.6641	5.9209	0.9222	1.9763
	alpha: 0.5714	0.6641	5.9206	0.9222	1.9761
	alpha: 0.6429	0.6640	5.9204	0.9222	1.9760
	alpha: 0.7143	0.6640	5.9202	0.9222	1.9759
	alpha: 0.7857	0.6639	5.9200	0.9222	1.9758
	alpha: 0.8571	0.6639	5.9198	0.9222	1.9756
	alpha: 0.9286	0.6639	5.9196	0.9221	1.9755
	alpha: 1.0	0.6638	5.9194	0.9221	1.9754
Support Vector Machine	gamma: 5e–05	0.6521	5.1176	0.3485	10.0000
	gamma: 0.0001	0.7329	4.4407	0.0414	10.0000
	gamma: 0.0005	0.9768	1.3381	–0.0312	10.0000
	gamma: 0.001	0.9792	1.2938	–0.0050	10.0000
	gamma: 0.005	0.9741	1.5829	0.9509	1.4743
	gamma: 0.01	0.9641	1.8496	0.9967	0.3733
	gamma: 0.05	0.9576	1.8768	0.9940	0.4599
	gamma: 0.1	0.9125	2.3731	0.9805	0.7568
	gamma: 0.5	–0.1000	8.4960	–0.0873	3.6241
	gamma: 1.0	–0.1000	10.0000	–0.1000	6.3901
Gradient Boosted Trees	depth: 5 agents: 25	0.8083	3.1012	0.5752	3.0592
	depth: 5 agents: 50	0.9792	1.2824	0.9013	1.9262
	depth: 5 agents: 60	0.9869	1.0772	0.9258	1.7290
	depth: 5 agents: 80	0.9917	0.8930	0.9490	1.4875
	depth: 5 agents: 100	0.9933	0.8107	0.9590	1.3575
	depth: 5 agents: 100	0.9945	0.7255	0.9690	1.1842
	depth: 5 agents: 125	0.9939	0.7668	0.9655	1.2582
	depth: 5 agents: 150	0.9943	0.7406	0.9688	1.1978
	depth: 3 agents: 100	0.9921	0.9016	0.9354	1.6317
	depth: 10 agents: 25	0.8209	3.0616	0.6004	3.0999
	depth: 10 agents: 50	0.9743	1.3811	0.8490	2.4250
	depth: 15 agents: 25	0.8171	3.0784	0.5647	3.2779
	depth: 30 agents: 25	0.8159	3.0941	0.5607	3.2746
Multilayer Perceptron	(64,)	–0.1000	10.0000	0.9666	1.2405
	(100,)	–0.1000	10.0000	0.9867	0.7459
	(150,)	–0.1000	10.0000	0.9949	0.4483
	(64, 64)	0.8595	3.6875	0.9918	0.5981
	(100, 100)	0.9372	2.5621	0.9917	0.5718
	(25, 25, 25)	0.9707	1.7303	0.9986	0.2311
	(64, 64, 64)	0.9658	1.8863	0.9915	0.5745
	(96, 96, 96)	0.9686	1.8453	0.9937	0.5062
	(25, 25, 25, 25)	0.9568	2.1576	0.9929	0.5686
	(64, 64, 64, 64)	0.9746	1.6095	0.9918	0.5696

Table 2. Error values for building design problems A and B for 1040 training samples. Left represents non-standardized input values, right standardized. Red indicates better accuracy and grey values are clipped.

standardization of the inputs. The GP model's performance increased on Building B because of the

	FLA Metrics N = 104	Building A			Building B		
		FDC	var of state	var of costs	FDC	var of state	var of costs
Gaussian Process	alpha: 0.0	0.4490	0.0123	0.0591	0.6904	0.0009	0.0897
	alpha: 0.0714	0.4337	0.0123	0.0592	0.7020	0.0009	0.0897
	alpha: 0.1429	0.4304	0.0123	0.0592	0.6895	0.0009	0.0897
	alpha: 0.2143	0.4181	0.0123	0.0592	0.6927	0.0009	0.0897
	alpha: 0.2857	0.4341	0.0123	0.0592	0.6984	0.0009	0.0897
	alpha: 0.3571	0.4318	0.0123	0.0592	0.6882	0.0009	0.0897
	alpha: 0.4286	0.4321	0.0123	0.0592	0.6938	0.0009	0.0898
	alpha: 0.5	0.4229	0.0123	0.0592	0.6927	0.0009	0.0898
	alpha: 0.5714	0.4416	0.0123	0.0592	0.6899	0.0009	0.0898
	alpha: 0.6429	0.4356	0.0123	0.0592	0.6914	0.0009	0.0898
	alpha: 0.7143	0.4312	0.0123	0.0592	0.6863	0.0009	0.0898
	alpha: 0.7857	0.4245	0.0123	0.0592	0.6900	0.0009	0.0898
	alpha: 0.8571	0.4357	0.0123	0.0592	0.6931	0.0068	0.0898
	alpha: 0.9286	0.4306	0.0123	0.0592	0.6991	0.0068	0.0898
	alpha: 1.0	0.4411	0.0123	0.0592	0.6904	0.0068	0.0898
Support Vector Machine	gamma: 5e–05	0.4576	0.0018	0.0801	0.3797	0.0126	0.0694
	gamma: 0.0001	0.5237	0.0077	0.0729	0.5284	0.0426	0.0481
	gamma: 0.0005	0.4451	0.0015	0.0708	0.3320	0.0013	0.0617
	gamma: 0.001	0.4327	0.0011	0.0907	0.6386	0.0078	0.0830
	gamma: 0.005	0.2499	0.0198	0.0902	0.6487	0.0078	0.0833
	gamma: 0.01	0.3205	0.0180	0.1051	0.6628	0.0075	0.0875
	gamma: 0.05	0.6688	0.0005	0.0958	0.7677	0.0079	0.0858
	gamma: 0.1	0.7480	0.0075	0.0804	0.7846	0.0032	0.0829
	gamma: 0.5	0.1329	0.0220	0.0281	0.6640	0.0090	0.0454
	gamma: 1.0	0.0743	0.0268	0.0291	0.4289	0.0339	0.0262
Gradient Boosted Trees	depth: 5 agents: 25	0.1363	0.0477	0.0611	0.0815	0.0031	0.0751
	depth: 5 agents: 50	0.1428	0.0464	0.0616	0.3249	0.0077	0.0806
	depth: 5 agents: 60	0.1272	0.0477	0.0615	0.3401	0.0080	0.0803
	depth: 5 agents: 80	0.1483	0.0477	0.0607	0.5220	0.0077	0.0784
	depth: 5 agents: 100	0.2075	0.0477	0.0624	0.5671	0.0077	0.0754
	depth: 5 agents: 100	0.1868	0.0477	0.0594	0.4983	0.0026	0.0767
	depth: 5 agents: 125	0.1275	0.0464	0.0621	0.5232	0.0077	0.0772
	depth: 5 agents: 150	0.2062	0.0477	0.0608	0.5147	0.0077	0.0768
	depth: 3 agents: 100	0.1286	0.0355	0.0634	0.5207	0.0077	0.0975
	depth: 10 agents: 25	0.0286	0.0477	0.0613	0.3177	0.0095	0.0676
	depth: 10 agents: 50	0.1070	0.0477	0.0598	0.0797	0.0349	0.0676
	depth: 15 agents: 25	0.0704	0.0477	0.0592	0.2147	0.0135	0.0666
	depth: 30 agents: 25	0.0154	0.0477	0.0591	0.0997	0.0135	0.0682
Multilayer Perceptron	(64,)	0.7037	0.0270	0.0573	0.7692	0.0231	0.0506
	(100,)	0.6422	0.0068	0.0708	0.7694	0.0230	0.0449
	(150,)	0.7086	0.0069	0.0895	0.8256	0.0029	0.0675
	(64, 64)	0.7375	0.0006	0.0864	0.7890	0.0332	0.0475
	(100, 100)	0.7461	0.0320	0.0694	0.8265	0.0117	0.0501
	(25, 25, 25)	0.7743	0.0068	0.0734	0.8003	0.0074	0.0452
	(64, 64, 64)	0.7865	0.0328	0.0549	0.6995	0.0078	0.0579
	(96, 96, 96)	0.7753	0.0324	0.0625	0.8557	0.0024	0.0698
	(25, 25, 25, 25)	0.7205	0.0015	0.0848	0.8168	0.0260	0.0460
	(64, 64, 64, 64)	0.8045	0.0226	0.0477	0.7896	0.0071	0.0662

	FLA Metrics N = 1040	Building A			Building B		
		FDC	var of state	var of costs	FDC	var of state	var of costs
Gaussian Process	alpha: 0.0	0.3464	0.0002	0.0677	0.6683	0.0013	0.0409
	alpha: 0.0714	0.3588	0.0002	0.0677	0.6713	0.0013	0.0409
	alpha: 0.1429	0.3585	0.0002	0.0677	0.6738	0.0013	0.0409
	alpha: 0.2143	0.3588	0.0002	0.0677	0.6729	0.0013	0.0409
	alpha: 0.2857	0.3624	0.0002	0.0677	0.6733	0.0013	0.0409
	alpha: 0.3571	0.3547	0.0002	0.0677	0.6682	0.0013	0.0409
	alpha: 0.4286	0.3478	0.0002	0.0677	0.6671	0.0013	0.0409
	alpha: 0.5	0.3721	0.0002	0.0677	0.6643	0.0013	0.0409
	alpha: 0.5714	0.3507	0.0002	0.0677	0.6684	0.0013	0.0409
	alpha: 0.6429	0.3661	0.0002	0.0677	0.6669	0.0013	0.0409
	alpha: 0.7143	0.3452	0.0002	0.0677	0.6644	0.0013	0.0409
	alpha: 0.7857	0.3715	0.0002	0.0677	0.6779	0.0013	0.0409
	alpha: 0.8571	0.3719	0.0002	0.0677	0.6677	0.0013	0.0409
	alpha: 0.9286	0.3607	0.0002	0.0677	0.6669	0.0013	0.0409
	alpha: 1.0	0.3467	0.0002	0.0677	0.6750	0.0013	0.0409
Support Vector Machine	gamma: 5e–05	0.4859	0.0002	0.0624	0.3128	0.0142	0.0289
	gamma: 0.0001	0.6131	0.0002	0.0612	0.1851	0.0167	0.0251
	gamma: 0.0005	0.3069	0.0018	0.0608	0.1157	0.0101	0.0310
	gamma: 0.001	0.3188	0.0011	0.0610	0.0075	0.0117	0.0175
	gamma: 0.005	0.2390	0.0009	0.0569	0.4266	0.0063	0.0353
	gamma: 0.01	0.2905	0.0059	0.0468	0.6691	0.0106	0.0335
	gamma: 0.05	0.3910	0.0056	0.0588	0.7671	0.0105	0.0344
	gamma: 0.1	0.1617	0.0057	0.0502	0.7940	0.0020	0.0383
	gamma: 0.5	0.5467	0.0107	0.0209	0.6223	0.0077	0.0321
	gamma: 1.0	0.1392	0.0299	0.0070	0.3557	0.0163	0.0176
Gradient Boosted Trees	depth: 5 agents: 25	0.1419	0.0090	0.0504	0.3540	0.0005	0.0492
	depth: 5 agents: 50	0.1845	0.0100	0.0494	0.4590	0.0005	0.0453
	depth: 5 agents: 60	0.1431	0.0084	0.0491	0.5140	0.0005	0.0447
	depth: 5 agents: 80	0.1441	0.0084	0.0485	0.5053	0.0005	0.0443
	depth: 5 agents: 100	0.1499	0.0084	0.0485	0.5469	0.0008	0.0439
	depth: 5 agents: 100	0.1472	0.0084	0.0482	0.6511	0.0006	0.0440
	depth: 5 agents: 125	0.1903	0.0100	0.0481	0.5950	0.0011	0.0435
	depth: 5 agents: 150	0.1282	0.0084	0.0479	0.5969	0.0007	0.0436
	depth: 3 agents: 100	0.1701	0.0142	0.0456	0.5350	0.0019	0.0390
	depth: 10 agents: 25	0.1922	0.0078	0.0522	0.1151	0.0021	0.0435
	depth: 10 agents: 50	0.2051	0.0054	0.0522	0.5388	0.0007	0.0448
	depth: 15 agents: 25	0.2405	0.0055	0.0522	0.3995	0.0009	0.0437
	depth: 30 agents: 25	0.1643	0.0055	0.0522	0.2085	0.0019	0.0439
Multilayer Perceptron	(64,)	0.7058	0.0068	0.0352	0.8401	0.0015	0.0379
	(100,)	0.7290	0.0205	0.0299	0.8288	0.0018	0.0330
	(150,)	0.7938	0.0072	0.0401	0.7327	0.0015	0.0329
	(64, 64)	0.6951	0.0006	0.0423	0.6855	0.0026	0.0320
	(100, 100)	0.6970	0.0008	0.0493	0.6569	0.0070	0.0302
	(25, 25, 25)	0.4742	0.0022	0.0460	0.7002	0.0026	0.0328
	(64, 64, 64)	0.5019	0.0057	0.0475	0.6937	0.0016	0.0331
	(96, 96, 96)	0.4717	0.0009	0.0479	0.7673	0.0013	0.0334
	(25, 25, 25, 25)	0.4034	0.0013	0.0469	0.6379	0.0013	0.0365
	(64, 64, 64, 64)	0.4354	0.0021	0.0497	0.7185	0.0027	0.0302

Table 3. Fitness landscape values for building design problems A and B. Left represents 104 training samples, and right 1040 training samples. Red indicates high values and blue lower.

linear nature of the design problem. The standout performer is GBT whose standardized and non-standardized accuracy is similar.

Evaluating the error for 104 training samples showed overall weaker model performance with GP and GBT performing the best with fewer training samples. Whether 104 standardized or non-standardized training samples were used, the MPL model's performance deteriorated significantly.

Moving to the results of the fitness landscape in table 3, here the number of training samples had a noticeable effect on the measured fitness landscape. GP, GBT, and MLP models all show evidence of smoothing when trained on a lower number of training examples. When considering the fitness landscape defined by the SVM model, the distance parameter *gamma* shows a measurable, though not predictable, effect on the fitness landscape as it sweeps through the possible parameter values between zero and one.

Best Performers

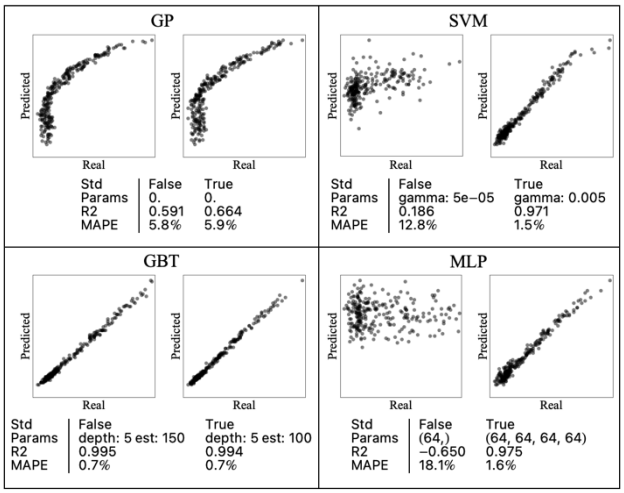

Figure 4. Best Building A model performers with 1040 training samples.

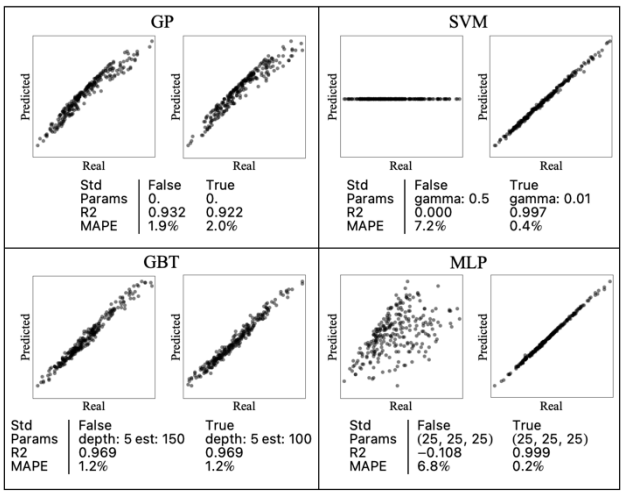

Figure 5. Best Building B model performers with 1040 training samples.

Taking the top performers of each model by R-squared value reveals important information applicable to surrogate modeling. The non-linear nature of building design problem A meant models with more layers or more estimators performed well. The results document how the performance of SVM with RBF kernels and MLP models improve when inputs are standardized. With the more linear nature of building design problem B, SVM and MLP return extremely high accuracy, whereas the GP is limited in accuracy because of the fundamental assumption Gaussian noise is present.

The duration of training for each model constitutes a final data point when comparing models' appropriateness for surrogate modeling. In scenarios where the number fo training samples is low, GP, SVM, and GBT models exhibit near-instantaneous training, while MLP models demonstrate comparatively slow performance. As the number of training samples increase, SVM's performance declines, and the MLP's training time continues to increase. Conversely, the training duration of GP and GBT models remains relative constant even with the addition of more training samples.

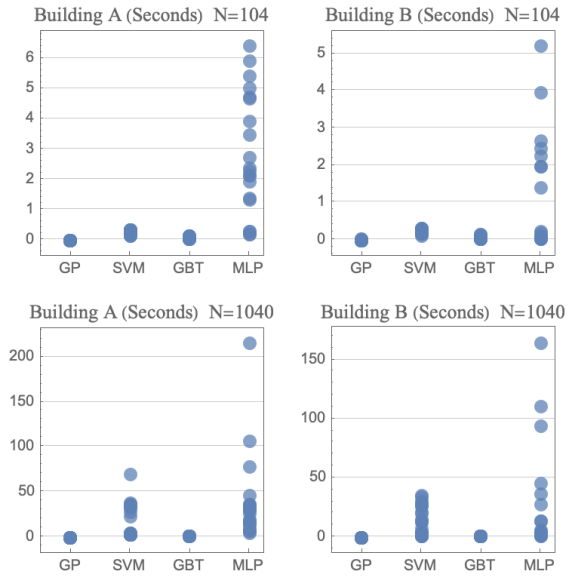

Figure 6. Training time in seconds.

Conclusions

The GBT model demonstrates superior performance due to its high computational efficiency and its ability to maintain accuracy across varying input conditions. In the testing phase, the GBT model exhibited MAPE values ranging from 0.7 to 1.2% and R2 values ranging from 0.969 to 0.995. This level of performance is evident in the EnergyPlus fitness landscape metrics, where the GBT models achieved some of the lowest error rates when compared to the actual EnergyPlus data (figure 7). Specifically, the GBT model demonstrated low error rates for both buildings in terms of the FDC and variance of state metrics.

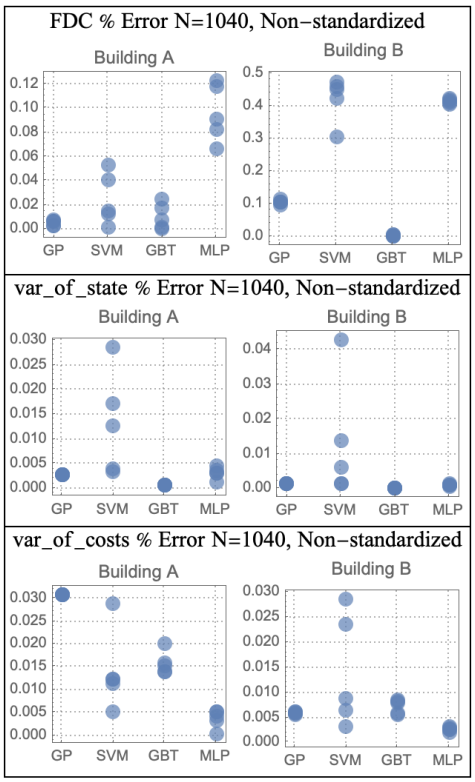

Figure 7. Absolute Error (%) between the measured fitness landscape metrics for the EnergyPlus data and the surrogate model.

In figure 8, a combination of the best performance models for both problems is shown. A negative correlation is observed for the FDC value. This suggests that as model accuracy declines, the problem design space becomes smoother. A similar pattern is found with the variance of cost metric, as an increase in model accuracy corresponds to an increase in measured variance. However, the correlation between accuracy and the variance of state metrics is found to be minimal.

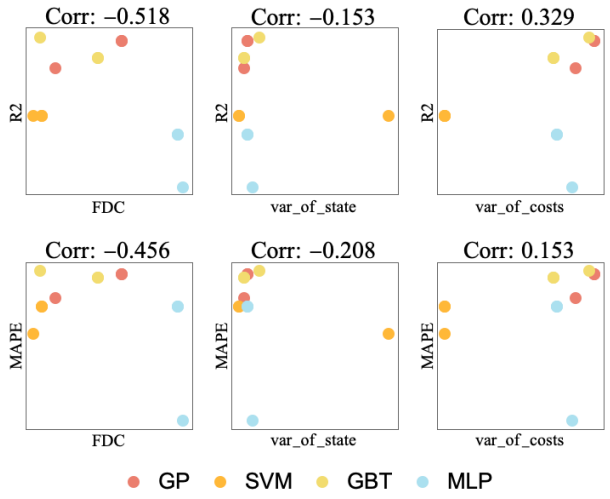

Figure 8. N=1040 pair plots comparing error and fitness landscape values for best performing models with correlation recorded above.

To summarize the fitness landscape results for each model type, the GP model tends to converge on a surface that is dissimilar to the original. SVM can alter the problem design surface depending on the parameters chosen. MLP models have the ability to replicate the output distribution closely, and lastly, GBT can accurately predict output values and represent the original design problem surface.

Future work

Using the same methodology as outlined above, more building design problems should be developed for the BESOS platform. This expanded dataset has the potential to reveal further novel characteristics of model selection on the building design problems' fitness landscape. Better understanding the relationship between a design problem's fitness landscape and performance can lead to better surrogate modeling.

Effort is also being put toward formatting the set of building design problems as a Python module for use on the BESOS platform and be made available publicly.

References

Chen, B. J., & Chang, M. W. (2004). Load forecasting using support vector machines: A study on EUNITE competition 2001. IEEE transactions on power systems, 19(4), 1821-1830.

Deb, C., Dai, Z. & Schlueter, A. (2021), A machine learning-based framework for cost-optimal building retrofit, Applied energy, vol. 294, pp. 116990.

Djuric, N., Novakovic, V., Holst, J., & Mitrovic, Z. (2007). Optimization of energy consumption in buildings with hydronic heating systems considering thermal comfort by use of computer-based tools. Energy and Buildings, 39(4), 471-477

EnergyPlus. (2022) Computer software. https://energyplus.net/ Vers. 22.2.0. USDOE Office of Energy Efficiency and Renewable Energy (EERE), Energy Efficiency Office. Building Technologies Office.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. Annals of statistics, 1189-1232.

Gan, V. J., Lo, I. M., Ma, J., Tse, K. T., Cheng, J. C., & Chan, C. M. (2020). Simulation optimisation towards energy efficient green buildings: Current status and future trends. Journal of Cleaner Production, 254, 120012.

Goel, H., Melnyk, I. & Banerjee, A. (2017). R2N2: Residual Recurrent Neural Networks for Multivariate Time Series Forecasting, arXiv:1709.03159.

Khan, Z.A., Hussain, T., Ullah, A., Rho, S., Lee, M. & Baik, S.W. (2020). Towards Efficient Electricity Forecasting in Residential and Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework, Sensors (Basel, Switzerland), vol. 20, no. 5, pp. 1399.

Kochenderfer, M. J., & Wheeler, T. A. (2019). Algorithms for optimization. Mit Press.

Melo, A. P., Versage, R. S., Sawaya, G., & Lamberts, R. (2016). A novel surrogate model to support building energy labelling system: A new approach to assess cooling energy demand in commercial buildings. Energy and Buildings, 131, 233-247.

Mohandes, M. (2002). Support vector machines for short-term electrical load forecasting. International Journal of Energy Research, 26(4), 335-345.

Osborne, M. A., Rogers, A., Roberts, S. J., Ramchurn, S. D., & Jennings, N. (2010). Bayesian Gaussian process models for multi-sensor time-series prediction. Inference and Learning in Dynamic Models.

Papadopoulos, S., Azar, E., Woon, W. L., & Kontokosta, C. E. (2018). Evaluation of tree-based ensemble learning algorithms for building energy performance estimation. Journal of Building Performance Simulation, 11(3), 322-332.

Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Touzani, S., Granderson, J., & Fernandes, S. (2018). Gradient boosting machine for modeling the energy consumption of commercial buildings. Energy and Buildings, 158, 1533-1543.

Waibel, C., Mavromatidis, G., Evins, R., & Carmeliet, J. (2019). A comparison of building energy optimization problems and mathematical test functions using static fitness landscape analysis. Journal of Building Performance Simulation, 12(6), 789-811.

Waibel, C., Wortmann, T., Evins, R., & Carmeliet, J. (2019). Building energy optimization: An extensive benchmark of global search algorithms.

Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA.

Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA.

Westermann, P., & Evins, R. (2019). Surrogate modelling for sustainable building design–A review. Energy and Buildings, 198, 170-186.

Westermann, P., Christiaanse, T. V., Beckett, W., Kovacs, P., & Evins, R. (2021). besos: Building and Energy Simulation, Optimization and Surrogate Modelling.

Wetter, M., & Wright, J. (2004). A comparison of deterministic and probabilistic optimization algorithms for nonsmooth simulation-based optimization. Building and environment, 39(8), 989-999.

6.3 Appendix C

A comparison of machine learning functions for time-series prediction in buildings

Birdsell, B. & Evins, R. (2023)
Accepted to 18th IBPSA International Conference and Exhibition. Shanghai, China.

Abstract

This work undertakes a comprehensive comparison of different neural net architectures for both day-ahead and annual predictions with specific focus on their application in buildings. This comparison of four types of nets under two prediction conditions documents and describes methods that will be impactful to building designers and operators. The neural net architectures tested were convolutional-based residual neural nets, recurrent neural networks, WaveNet, and a hybrid residual-RNN model. The results showed that while all nets are suitable for time-series prediction, Convolutional nets have strengths in forecasting sequences. This has implications when trying to predict the short-term behavior of building systems. When annual performance was estimated, convolutional-based nets showed a distinct advantage in accurately predicting the expected monthly minimum and maximum values. This result can contribute to better decision making in the design and operation of buildings.

Introduction

The goal of this project is to inform best practice in regard to predicting building performance using machine learning. Like many papers on this topic, it is fundamentally motivated by the fact that in North America buildings are responsible for consuming 30-40% of all energy produced (U.S. Department of Energy 2012, Statistics Canada 2016). Therefore, improving building operations has the potential to make a significant impact reducing energy consumption and greenhouse gas emissions.

Characteristics inherent in building data also make time series prediction challenging. These are important characteristics to understand because they are often tied to the structure of the data the model is trying to learn and make predictions from. Notably, these challenges are similar to the characteristics with which traditional statistical methods struggle.

Included in long-term time series are patterns of different length such as seasonal, weekly and daily variations. Certain high resolution building time series, taken in 15 minutes or shorter time steps, can contain high frequency noise that present a difficult environment to model. Trends in the data can also represent non-linear relationships. Systems can evolve in ways that go beyond linearly proportional effects.

In recognition of the increasing amount of data a monitored building can produce, and the rapidly evolving ecosystem of machine learning algorithms, building owners and operators are left with a lot of building data and few answers about how best to use it. Instead of modifying an existing neural net architecture until it's considered novel, adding to the confusing landscape of choices, this project instead undertakes a detailed study of well-known and well-established network architectures and compares their accuracy and computational performance. The research looks at four neural net architectures under two types of prediction conditions, day-ahead forecasting and long-term predictions. This body of work can contribute to the decision-making process in several of the applications described below.

Prediction & Forecasting

Seeking a robust comparison of the algorithms in question, both day-ahead and annual predictions are considered. This brings up a subtle and informal distinction that should be explicitly noted between these two terms and why they were chosen. Nate Silver in his 2012 book *The Signal and the Noise* states that in many fields the terms "forecast" and "prediction" are used interchangeably. In this paper they are not used interchangeably. Forecasting will be taken to mean a short-term estimate, while prediction will refer to a longer-term aggregated estimate. Using energy consumption as an example, an energy consumption forecast would look at the past week of data and estimate the next day's usage in high-resolution. Prediction, on the other hand, would estimate the energy expected to be consumed over a year.

The main applications for these architectures which frame this work are Smart Buildings and Model Predictive Control. With the decreasing cost of computation and sensor deployment, it's becoming more common for buildings to implement modern building management systems, of which

predictive control is a major benefit. Intelligent management systems have two goals: Firstly, achieving a high degree of comfort for users concerning the thermal, air quality, or visual performance of the space. And secondly, achieving high energy efficiency while minimizing operational costs (Serale et al. 2018). Chen (2018) and Zhan (2022) draw a connection between the importance of model predictive control and the increased use of renewable resources in buildings. These sources of energy have a large amount of variation in their availability over time. Having intelligent building systems which can predictively adapt to conditions is an important characteristic in utilizing renewable resources effectively.

Model Predictive Control with neural nets addresses several criteria that traditional control methods based on simulation cannot. MPC has the ability to handle uncertainties and interdependence among the parameters, track slow-moving processes that might be time delayed within the evolving system, capture nonlinear dynamics, and, finally, predict multiple objectives at the same time. The underlying thrust of the benefits of MPC is that it can use the system for anticipatory control rather than corrective control (Afram 2014).

Turning to the benefits of better prediction algorithms in the long-term, the existing building stock offers a rich target in terms of potential retrofits. Building owners and operators have started to generate large volumes of data regarding the performance and behaviour of their assets over time (Gonzalez-Vidal et al. 2017). Previous work with RNNs and multivariate time series have looked at the classification of buildings (Baasch 2019). Deb (2021) looked at machine-learning applications for cost-optimal building retrofits.

Literature Review

Approximating Functions

Classical prediction methods for time series data involve regression, differences, and moving averages of residual errors, as described in Box and Jenkins' 1970 book *Time Series Analysis; Forecasting and Control*. However, these methods assume linear dependence over time and do not account for non-uniform causal interrelationships among variables. Additionally, selecting the model's parameters for good results is challenging, often leading to ad hoc adjustments. In contrast, neural networks (NNs) are black-box models that do not require physical simulations of the building. They are composed of linear layers which mimic the structure of the human brain (Werbos 1974). NNs rely solely on data, using current indicators such as temperature and humidity to predict the target variable. Studies have shown that NN techniques outperform traditional ARIMA methods in terms of accuracy (Divina et al. 2019, Cai et al. 2019).

Recurrent Neural Networks

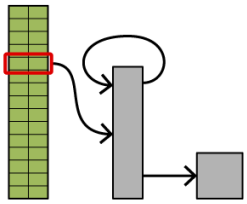

Figure 1. Schematic of RNN.

Recurrent neural networks (RNNs) are widely used for time series modeling and are currently state-of-the-art for natural language processing, speech recognition, text generation, and language translation (Graves et al. 2013, Graves 2013, Sutskever 2014). RNNs are well-suited for time series because they are designed to capture dependencies among consecutive input features. This is achieved by using different configurations of state memory depending on the specific type of RNN layer used. During training, the sequence is fed into the net one by one and the layer weights are updated toward the target value. RNNs have been studied in the field of building performance and energy consumption by Gao (2019) and Sehovac (2019) due to their success in the temporal domain.

Convolutional Based Residual Networks

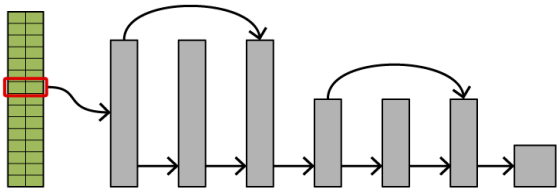

Figure 2. Schematic of CNN-based ResNet.

The basic idea of a convolution is to calculate the product of a function (the kernel) applied to another function (the time series). This product can reveal latent structure in the data. Time series are not often thought of as containing spatial information, but, under certain assumptions, that is one way they can be interpreted. The growth of the method follows from the increased availability of computational power where the kernel can be applied repeatedly to update arrays of weights to approximate complex behaviour.

In this project, convolutional layers are packaged into blocks allowing for skipped connections between layers that have been shown to improve accuracy on time series prediction and classification (He et al. 2016, Wang et al. 2019). While a linear network tries to learn the output function $H(x)$, the layers in a residual block aim to learn the residual: $H(x)=F(x)-x$ (Choi 2018, Benhaddi 2021). This structure allows the net to extract more characteristic information from the data.

Temporal Dilated Causal Convolutional Networks

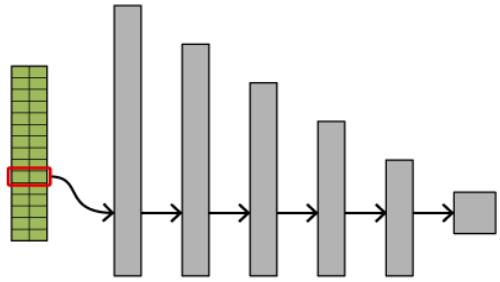

Figure 3. Schematic of WaveNet

Similar to how convolutional neural networks first saw success in image processing, the development of Temporal Dilated Causal Convolutional Networks sought to address some of the challenges of audio processing. Oord et al. (2016) describes the original idea, shortened to WaveNet, and its application to speech generation and how the network deals with high-noise environments to extract meaningful structure at different scales. At the smallest scales, the transient movement of the waveforms can appear quite chaotic but it's in these instantaneous sounds that language is communicated. At the other end of the scale, we understand how meaning can be built up over the length of sentences and paragraphs to express complex ideas. (Oord et al. 2016)

The WaveNet architecture still uses convolutional layers, but instead of using the kernel width to drive the process, the WaveNet, over a of series layers, expands its dilation parameters, effectively expanding its receptive field. After Oord's 2016 publication, Voss et al. (2018) applied the method to residential short-term load forecasting where the technique bettered linear regression methods and simple ANNs. Rueda et al. (2021), continued to apply the method to forecasting electrical loads in France. The authors note in regards to the behaviour of the network, the expanding receptive window is responsible for helping deal with data sparsity, long-term sequence relations, and multi-resolution.

Hybrid Neural Net

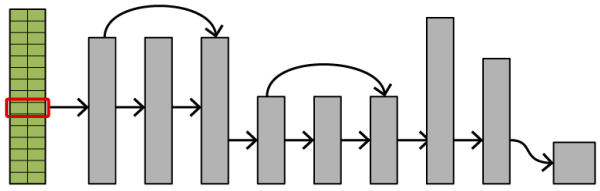

Figure 4. Schematic of Hybrid Convolutional-RNN network.

A hybrid model is also considered for this comparison, drawing on the strengths of both Residual and RNN architectures. Time series analysis and electricity forecasting with hybrid neural nets has seen two in-depth reports from Goel et al. (2017) and Khan et al. (2020) each looking at short term predictions. Another study that exposes the hybrid neural net to a rich set of building data is Deb et al.'s (2021) retrofit prediction research. Here the hybrid neural network was trained on the energy simulation data of several different potential building retrofit scenarios. The hybrid ResNet-RNN then predicted cost-optimized retrofit solutions with greater computational efficiency than if all potential designs were simulated.

Methodology

Data Source

In this study, Building 59 at Lawrence Berkeley National Laboratory in Berkeley, California was used for both annual and day-ahead predictions. It is a 112, 000 square foot office building containing two office floors, one mechanical equipment floor, and one floor dedicated to a data and computing center. Only the datasets for the office floors were released.

The data that comes directly from sensors or mechanical units has several drawbacks that if left unaddressed could lead to substandard results for any machine learning application. One requirement for the types of neural nets researched in this study is that the input data is required to have uniform time steps. Gaps in data collection of varying lengths, from hours to months, can also occur when a sensor or mechanical unit fails or is down for maintenance etc. Unpredictable transient events are also common in building data. These events can be caused by an influx of users at the site, such as a special event, or by powering up large mechanical units. To a certain degree, depending on the requirements of the project, these sorts of artifacts can be mitigated by pre-processing the data.

Building data was collected from January 2018 to December 2021. The dataset was analyzed and prepared by LBNL and addressed data file aggregation and categorization, diagnostic gap filling, and outlier modification. Detailed information about the raw data post-processing can be found in the dataset's accompanying documentation (Lawrence Berkeley National Laboratory 2021).

Data Processing

Two further steps were taken to prepare the dataset for the training and testing undertaken in this experiment. After selecting which features would be used and targeted, the data was down sampled to 30-minute time steps.

The second step was to standardize the input features. This was done by subtracting the mean of the feature from each value and dividing it by the standard deviation of the feature. No modification was done to the prediction targets. When training for the annual predictions the dataset was further down sampled to a 1-hour resolution to ease the computational burden of the training phase.

Use Case

The dataset was used to predict the flow rate of the four roof top units. Roof top units providing treated fresh air presents an energy intensive load worthy of study for greater efficiency. The prediction model has no underlying

knowledge of the physical system. During the training process the predicted flow rate is deduced from the RTUs previous state, with the approximating function created by the neural nets used to map the inputs to the flow rate at t+1. Predicting the required flow rate of the RTUs in the future given the system's previous state is representative of what needs to be predicted and controlled by the model in a building management system.

To ensure the day-ahead neural net architectures were sufficiently robust, four days were chosen three months apart. This tests the day-ahead architecture across different environmental conditions and mechanical demands.

Neural Net Training

The four RTUs were each treated as a feature which created a multivariable input at each time step. In the annual version a fifth feature was added, the month plus fraction of the month, so that the model could match up the preceding years day efficiently during training to improve the annual prediction of the test data.

For day head, a 72-hour window was used on 8 weeks of training data. The output of each window at 30 min timesteps was the next 30 minutes of RTU flow rate values. The annual training data was segmented into 3-hour periods with the previous 72 hours of data. This makes the target an array of 12 values for each of the 4 rooftop units.

Training was conducted on an Apple M1 Studio with 32GB of RAM operating macOS Ventura. All coding and neural net construction was completed in Wolfram Mathematica 13.

Inputs	Outputs
Roof top unit 1 standardized flow rate	Roof top unit 1 (CFM)
Roof top unit 2 standardized flow rate	Roof top unit 2 (CFM)
Roof top unit 3 standardized flow rate	Roof top unit 3 (CFM)
Roof top unit 4 standardized flow rate	Roof top unit 4 (CFM)
Month + Fraction of Month	

Table 1. List of model parameters.

Performance Metrics

This study utilizes Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as performance metrics for day-ahead predictions. MAPE has been widely used in building performance studies and has been reported to range between 5-10% for CNNs and RNNs (Gonzalez-Vidal 2017, Koutlis 2020). More recent studies have reported <4% mean absolute percent error and an R-squared value between 0.92 and 0.94 for short-term forecasting of heating loads (Eguizabal 2022). To calculate a final value, MAE and MAPE were averaged across RTUs and seasons. R-squared is a better metric for explanatory models and linear regression, but is included in this study for comparison purposes. ASHRAE Guideline 14 allows annual predictions to vary by up to 20%, but this may not be sufficient or meaningful in practice (A. S. Committee et al., 2002, Garrett 2014, Gonzalez-Vidal 2017). To evaluate annual predictions, the monthly min and max values are calculated and compared between the estimated and historical values. A naïve approach, which assumes only basic knowledge of the preceding data in the form of a moving average, is also included for additional context. The 6-hour moving average is used for short-term forecasting, and the average of the preceding year is used as a naïve annual estimate.

Results

Day-ahead Forecast

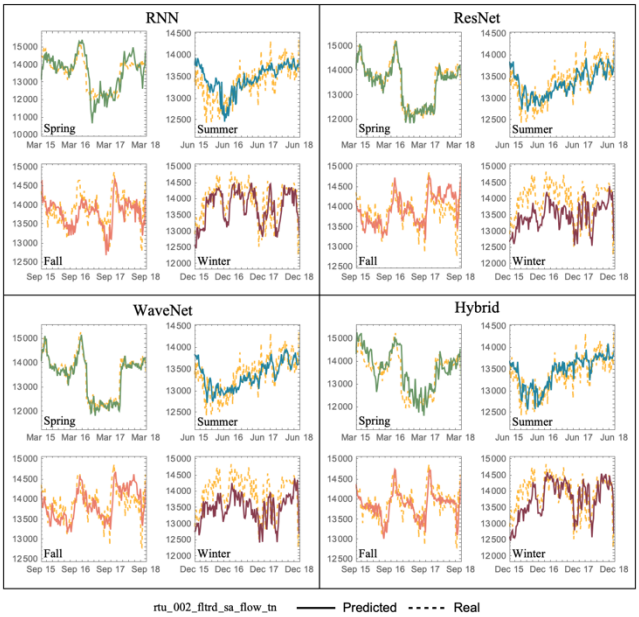

Figure 5. Day ahead results overview of RTU 2 in cubic feet per minute.

	24 hour			72 hour		
	MAPE (%)	**MAE (CFM)**	**R2**	**MAPE** (%)	**MAE (CFM)**	**R2**
RNN	2.7	338	0.94	2.9	362	0.93
ResNet	2.4	301	0.95	2.4	302	0.95
WaveNet	2.2	281	0.96	2.3	295	0.95
Hybrid	2.4	309	0.95	2.5	312	0.95
Naive	4.9	603	0.83	5.4	676	0.75

Table 2. Day ahead forecast results summary.

When the MAPE and MAE values are averaged among the four RTUs for the 24 hours after the target date, all four architectures show promising results, bettering the naïve model, and ended up with between 2.2% ~ 2.7% mean percent error. Except for the ResNet, the results show accuracy decrease as the forecast window extends past the 24-hour estimate and the model begins to encounter more data it's never encountered before. This is realistic of real-world building applications where time series are continually added to over time and only ever see small periods of new data as the forecast window moves and

model updated. Visually, both the ResNet and WaveNet did a reasonable job approximating the noise seen in the real building data. Closer inspection of the raw data shows that RTU 4 was the most difficult target to train with the error consistently higher across tests.

Annual Prediction

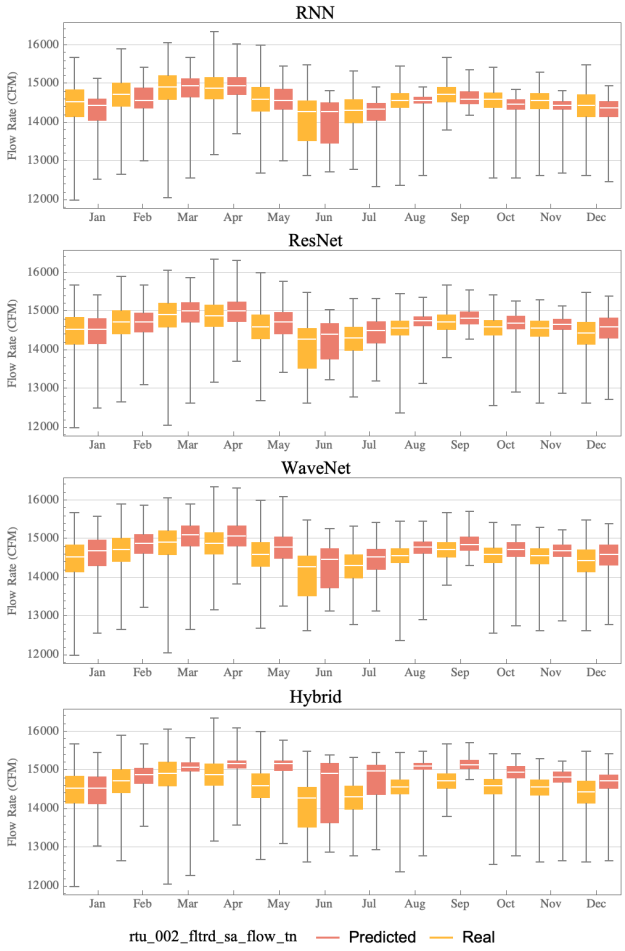

Figure 6. Monthly distribution of real and predicted values of RTU 2.

	MAPE (%)	**MAE (CFM)**	**R2**
RNN	2.2	331	0.56
ResNet	1.8	264	0.70
WaveNet	1.9	283	0.67
Hybrid	2.7	406	0.36
Naive	4.6	695	-1.00

Table 3. Annual prediction result summary.

From seeing three years of hourly training data, the ResNet and WaveNet excelled at generalizing to the fourth year testing data, representing the high-frequency noise of the original data closely. When inspecting the annual results visually, the ResNet and WaveNet did an admirable job predicting the distribution of monthly values and the monthly minimum and monthly values. The RNN was on its way to converge but with the early stopping method used to prevent overfitting ended up with smaller predicted values than the recorded historic values. The Hybrid net also tended to constrain values unrealistically.

Analysis & interpretation

Day-ahead Forecast

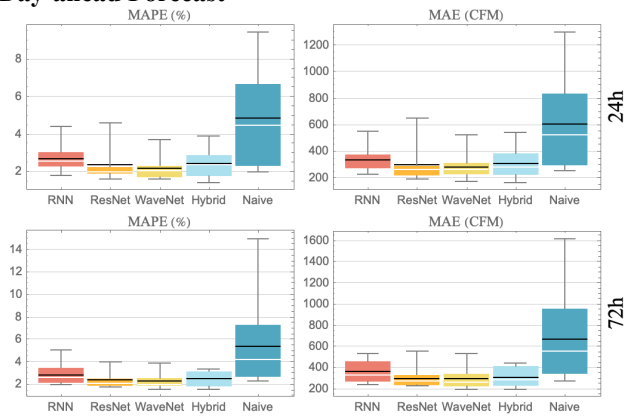

Figure 7. 24h (top) and 72h results (bottom). White line represents median values and black line the mean.

The Box-Whisker diagrams presented in Figure 7 provide a comprehensive overview of the four RTU error values obtained for each of the four target dates, resulting in a total of 16 measurements each. These diagrams enable a direct comparison between models based on the aggregated MAPE and MAE values, as well as their variance. Notably, both ResNet and WaveNet models exhibit tight clusters of error values, with 75% of values falling within the coloured box. Conversely, the RNN and Hybrid models exhibit slightly greater variance for these error metrics across different RTUs and seasons. Furthermore, the naive approach demonstrated inferior performance relative to other models.

Annual Prediction

The annual results show the RNN model only slightly behind the performance of the ResNet and WaveNet. The cause of deteriorating performance of the RNN and Hybrid model is seen in the lower portion of figure 8. The Hybrid model has poor accuracy throughout the year but especially RTU 4 in the months January to May. Through the latter half of the prediction period, it is the RNN model which struggles for accuracy on RTU 4's 2021 testing data, creating a cluster of high error values in the later months of the year.

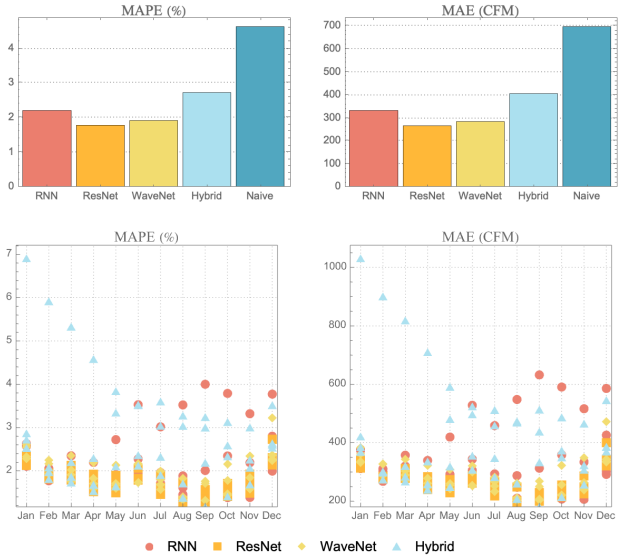

Figure 8. Top row shows averaged performance across all RTUs with the lower half representing each metric by month.

Figure 9. Annual min and max error (%) summary by RTU and month.

Figure 9 provides a detailed representation of model performance by graphing the minimum and maximum absolute percent error from the actual minimum and maximum values of the RTU. Notably, the ResNet and WaveNet models, which exhibited the best performance overall, tend to slightly overestimate the minimum value and demonstrate excellent predictive ability for the maximum value. This latter estimate is particularly crucial for the use case, as accurate predictions of expected maximum demand are critical in calculating the design loads of mechanical systems. Conversely, the min and max absolute percent error of the RNN model is observed to correspond to the model's performance as depicted in Figure 6.

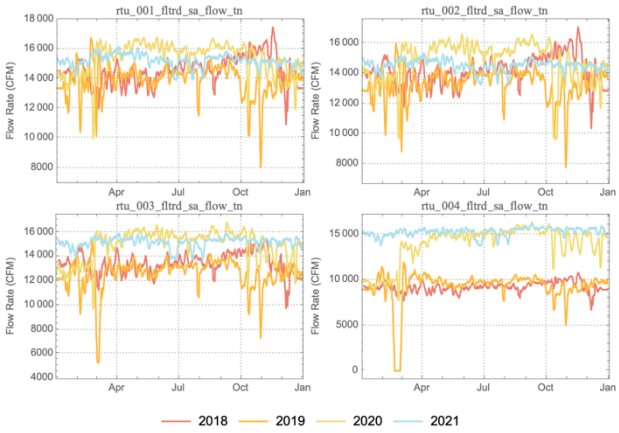

Figure 10. 7-day moving average overlay of annual training and testing data.

It's worth at this point stepping back slightly to revisit the details of the 2018 to 2020 training data to explain some of the outliers in the annual predictions. With the first three RTUs we see year-to-year consistency between the training data and 2021's testing data. RTU 4, however, presents much more complex behaviour for any model to learn because of the unexpected rise in flow rate values in 2020/2021 (figure 10).

In the context of annual predictions, it is pertinent to introduce a final performance metric. Decision-makers and designers rely on annual predictions to estimate the minimum and maximum loads expected to be encountered by building systems, making similarity between predicted and actual distributions critical to potential design decisions. The Jensen-Shannon Divergence, an effective yet underutilized tool for comparing distributions, has been previously used to assess the energy performance of building occupancy models (Mahdavi et al. 2016) and time series predictions in building energy simulations (Baasch et al. 2021). This metric yields lower values for more closely matching distributions, with a score of zero indicating an exact match. The ResNet and WaveNet models exhibit exceptional performance, closely approximating the actual distribution. In contrast, the RNN and Hybrid models demonstrate unrealistic peaks of values, resulting in a less accurate fit relative to the real data.

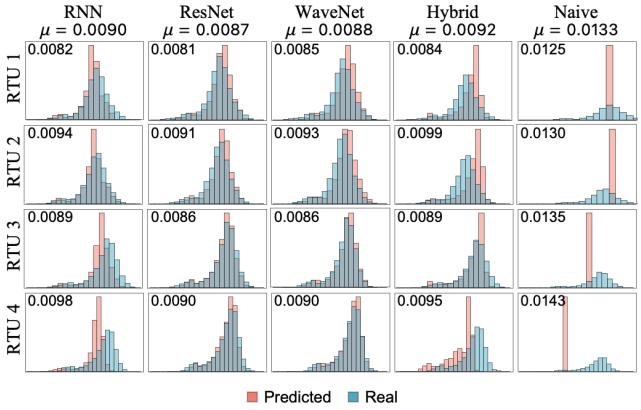

Figure 11. Jensen-Shannon Divergence. Lower values represent similar distribution.

Conclusion

Overview

This study evaluates the efficacy of ResNets, RNNs, WaveNet, and Hybrid architectures for model predictive control in building management systems and annual demand estimation. While ResNets and RNNs are well-established models in the field of time series modeling, this study expands upon these models by including WaveNet and Hybrid architectures. The findings provide valuable insights into the performance of these neural networks under both short- and long-term prediction conditions, making them applicable to building design and operations. Notably, the inclusion of convolutional layers in the ResNet and WaveNet models resulted in superior short-term sequence forecasting performance and estimation of the distribution of real monthly values. Thus, these models should be given greater consideration in the smart building industry when selecting a time series forecasting model. Although CNN-ResNets exhibited slower convergence rates, they have an exceptional ability to extract latent structures from the data, making them a strong contender for forecasting and prediction tasks. However, for applications where computational efficiency is a higher priority, models with a lighter computational load as measured by training duration, such as RNN and WaveNet, could be beneficial, with very little loss of accuracy in day-ahead forecasting (figure 12).

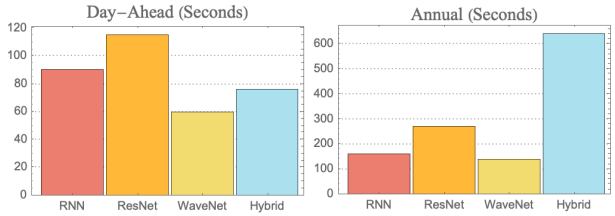

Figure 12. Training time in seconds for the model. Day-ahead results are averaged across four target dates.

It is important to note that the neural networks examined in this study were capable of achieving high levels of performance despite their lack of any inherent knowledge regarding the physical systems that produced the data. However, it is advisable to incorporate domain knowledge when applying these models in practice, in order to mitigate errors of interpretation and application. This aligns with the findings of Miller et al. (2020), who also advocate for a hybrid approach combining data science and engineering expertise for modeling building data.

Future work on building time series prediction could include additional neural network architectures. One compelling comparison to explore is that of a using gradient boosted tree model for time series prediction. Based on subsequence work from the author, gradient boosted trees have shown excellent performance characteristics across a range of prediction tasks.

References

A. S. Committee et al., (2002). *Ashrae guideline 14*, measurement of energy and demand savings. Atlanta.

Afram, A. & Janabi-Sharifi, F. (2014). *Theory and applications of HVAC control systems – A review of model predictive control (MPC)*, Building and environment, vol. 72, pp. 343-355.

Baasch, G., Rousseau, G., & Evins, R. (2021). *A Conditional Generative adversarial Network for energy use in multiple buildings using scarce data*. Energy and AI, 5, 100087.

Baasch, G.M. and Evins, R., (2019). *Targeting Buildings for Energy Retrofit Using Recurrent Neural Networks with Multivariate Time Series*. In Neural Inf. Process. Syst.

Benhaddi, M. & Ouarzazi, J. (2021) *Multivariate Time Series Forecasting with Dilated Residual Convolutional Neural Networks for Urban Air Quality Prediction*, Arabian journal for science and engineering (2011), vol. 46, no. 4, pp. 3423-3442.

Cai, M., Pipattanasomporn, M. & Rahman, S. (2019), *Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques*, Applied energy, vol. 236, pp. 1078-1088.

Chen, Y., Xu, P., Gu, J., Schmidt, F., & Li, W. (2018). *Measures to improve energy demand flexibility in buildings for demand response (DR): A review*. Energy and Buildings, 177, 125-139.

Choi, H., Ryu, S. & Kim, H. (2018), *Short-Term Load Forecasting based on ResNet and LSTM*, 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids, pp. 1.

Deb, C., Dai, Z. & Schlueter, A. (2021), A machine learning-based framework for cost-optimal building retrofit, Applied energy, vol. 294, pp. 116990.

Divina, F., García Torres, M., Goméz Vela, F.A. & Vázquez Noguera, J.L. (2019). *A Comparative Study of Time Series Forecasting Methods for Short Term Electric*

Energy Consumption Prediction in Smart Buildings, Energies (Basel), vol. 12, no. 10, pp. 1934.

Eguizabal, M., Garay-Martinez, R., & Flores-Abascal, I. (2022). *Simplified model for the short-term forecasting of heat loads in buildings*. Energy Reports, 8, 79-85.

Gao, Y., Fang, C. & Ruan, Y. (2019). *A novel model for the prediction of long-term building energy demand: LSTM with Attention layer*, IOP conference series. Earth and environmental science, vol. 294, no. 1, pp. 12033.

Garrett, A. & New, J. (2016). *Suitability of ASHRAE guideline 14 metrics for calibration*. ASHRAE transactions, vol. 122, no. 1, pp. 469.

Goel, H., Melnyk, I. & Banerjee, A. (2017). *R2N2: Residual Recurrent Neural Networks for Multivariate Time Series Forecasting*, arXiv:1709.03159.

Gonzalez-Vidal, A., et al. (2017). *Data driven modeling for energy consumption prediction in smart buildings*, IEEE, pp. 4562.

Graves, A. (2013). *Generating Sequences With Recurrent Neural Networks*. arXiv:1308.0850v5

Graves, A., Mohamed, A. & Hinton, G. (2013). *Speech recognition with deep recurrent neural networks*, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. pp. 6645-6649.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). *Deep Residual Learning for Image Recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770.

Khan, Z.A., Hussain, T., Ullah, A., Rho, S., Lee, M. & Baik, S.W. (2020). *Towards Efficient Electricity Forecasting in Residential and Commercial Buildings: A Novel Hybrid CNN with a LSTM-AE based Framework*, Sensors (Basel, Switzerland), vol. 20, no. 5, pp. 1399.

Koutlis, C., Papadopoulos, S., Schinas, M. & Kompatsiaris, I. (2020). *LAVARNET: Neural network modeling of causal variable relationships for multivariate time series forecasting*. Applied soft computing, vol. 96, pp. 106685.

Lawrence Berkeley National Laboratory (2021). *Dataset Description and Data Cleaning for Building 59 Raw Data: The Benchmark Datasets Project*.

Mahdavi, A., & Tahmasebi, F. (2016). *The deployment-dependence of occupancy-related models in building performance simulation*. Energy and Buildings, 117, 313-320.

Miller, C et al. *The ASHRAE Great Energy Predictor III competition: Overview and results*. Science & technology for the built environment, vol. 26, no. 10, pp. 1427-1447.

Oord, A.v.d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. & Kavukcuoglu, K. (2016), *WaveNet: A Generative Model for Raw Audio*. arXiv:1609.03499

Rueda, F.D., Suárez, J.D. & Alejandro del Real Torres (2021), *Short-Term Load Forecasting Using Encoder-Decoder WaveNet: Application to the French Grid*, Energies (Basel), vol. 14, no. 2524, pp. 2524.

Sehovac, L., Nesen, C. & Grolinger, K. (2019). *Forecasting Building Energy Consumption with Deep Learning: A Sequence to Sequence Approach*, IEEE, , pp. 108.

Serale, G., Fiorentini, M., Capozzoli, A., Bernardini, D. & Bemporad, A. (2018). *Model Predictive Control (MPC) for Enhancing Building and HVAC System Energy Efficiency: Problem Formulation*, Applications and Opportunities, Energies (Basel), vol. 11, no. 3, pp. 631.

Statistics Canada. (2016). *Survey of Commercial and Institutional Energy Use*, 2014. Monthly energy review. The Daily. https://www150.statcan.gc.ca/n1/daily-quotidien/160916/dq160916c-eng.pdf Accessed Feburary 23 2022.

Sutskever, I., Vinyals, O. & Le, Q.V. (2014). *Sequence to Sequence Learning with Neural Networks*. arXiv:1409.3215.

U.S. Department of Energy (2012). 2011 Buildings Energy Data Book. Buildings Technologies Program. Energy Efficiency and Renewable Energy. https://ieer.org/wp/wp-content/uploads/2012/03/DOE-2011-Buildings-Energy-DataBook-BEDB.pdf. Accessed February 23 2022.

Voss, M., Bender-Saebelkampf, C., & Albayrak, S. (2018, October). *Residential short-term load forecasting using convolutional neural networks*. 2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm) (pp. 1-6). IEEE.

Wang, K., Li, K., Zhou, L., Hu, Y., Cheng, Z., Liu, J. & Chen, C. (2019). *Multiple convolutional neural networks for multivariate time series prediction*. Neurocomputing (Amsterdam), vol. 360, pp. 107-119.

Zhan, S., Lei, Y., Jin, Y., Yan, D., & Chong, A. (2022). *Impact of occupant related data on identification and model predictive control for buildings*. Applied Energy, 323, 119580.