REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

SURROGATE MODEL APPROACHES FOR PREDICTING BUILDING ENERGY PERFORMANCE

Ayşe Rüya EFE

MASTER OF SCIENCE THESIS
Department of Architecture
Program of Building Physics

Supervisor
Assoc. Prof. Suzi Dilara MANGAN

July, 2024

REPUBLIC OF TÜRKİYE
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

SURROGATE MODEL APPROACHES FOR PREDICTING BUILDING ENERGY PERFORMANCE

A thesis submitted by Ayşe Rüya EFE in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 05.07.2024 in Department of Architecture, Program of Building Physics.

Assoc. Prof. Suzi Dilara MANGAN
Yildiz Technical University
Supervisor

Approved By the Examining Committee

Assoc. Prof. Suzi Dilara MANGAN, Supervisor
Yildiz Technical University _____

Prof. Gülay ZORER GEDİK, Member
Yildiz Technical University _____

Assist. Prof. Arkadaş ÖZAKIN, Member
Bogazici University _____

I hereby declare that I have obtained the required legal permissions during data collection and exploitation procedures, that I have made the in-text citations and cited the references properly, that I haven't falsified and/or fabricated research data and results of the study and that I have abided by the principles of the scientific research and ethics during my Thesis Study under the title of "Surrogate Model Approaches for Predicting Building Energy Performance" supervised by my supervisor, Assoc. Prof. Suzi Dilara MANGAN. In the case of a discovery of false statement, I am to acknowledge any legal consequence.

Ayşe Rüya EFE

Signature

This work was supported by the project no “2210-c Domestic Graduate Scholarship Program carried out by TÜBİTAK Scientist Support Programs Presidency (BİDEB)”

Dedicated to my beloved husband, Onur
and my dearest family, Cahide and Bilal

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Assoc. Prof. Suzi Dilara MANGAN, for all the help and support she offered. From the beginning of my academic journey, she showed enthusiasm and interest in my ideas. Despite all the challenges, she guided me with her extensive knowledge and experience. Her innovative and open-minded personality has served as a role model for me, and working alongside her has been a true honor.

I want to express my gratitude to my beloved spouse, Onur, for his unending support and love. Our journey, which we began from scratch, was made possible by his understanding and patience. He was so helpful that he taught me everything that I know today, and I continue to learn so much from him every day.

I want to express my sincere gratitude to my dear family, especially my mother, Cahide Aksu, and my father, Bilal Aksu, who have always stood by me with their support and love. I am also thankful to my dear friend Zeynep Püren Yıldırım, who has shared this journey with me.

Ayşe Rüya EFE

TABLE OF CONTENTS

LIST OF SYMBOLS viii

LIST OF ABBREVIATIONS ix

LIST OF FIGURES xi

LIST OF TABLES xiii

ABSTRACT xiv

ÖZET xvi

1 INTRODUCTION 1

1.1 Motivation and Background 1
1.2 Research Questions 2
1.3 Research Aim and Objectives 3
1.4 Thesis Methodology and Structure 3

2 BUILDING ENERGY PERFORMANCE AND SURROGATE MODEL 5

2.1 Building Energy Performance 5
2.1.1 Complexity of Building Performance Simulations 6
2.2 Surrogate Modeling 7
2.2.1 Machine Learning 7
2.3 Surrogate Model for Building Performance 9
2.3.1 Data Collection 9
2.3.2 Data Preprocessing 9
2.3.3 Sampling Strategies 11
2.3.4 Train/Test Splits 12
2.3.5 Machine Learning Models 12
2.3.6 Hyper-parameter Tuning 14
2.3.7 Model Performance Analysis 17
2.3.8 Model Applications 17

3 SURROGATE MODEL APPROACHES FOR BUILDING ENERGY PERFORMANCE PREDICTION AND IMPLEMENTATION 20

3.1 Surrogate Model Development 20
3.1.1 Data Collection 22
3.1.2 Data Preprocessing 29

3.1.3 Design Space Sampling Strategy 29
3.1.4 Train/Test Splits 30
3.1.5 Machine Learning Models Development 35
3.1.6 Hyper-Parameter Tuning 37
3.1.7 Model Performance Analysis 38
3.2 Results and Discussion 39
3.2.1 Interpolation Results 39
3.2.2 Extrapolation Results 42
3.2.3 Effect of Increasing Size of Training Data 43

4 CONCLUSION **46**
4.1 Addressing Research Questions 47
4.2 Research Limitations 48
4.3 Contributions and Future Studies 48

REFERENCES **50**

A PERFORMANCE DETAILS OF SURROGATE MODELS **55**

B DETAILS OF TUNED MACHINE LEARNING MODELS **57**

C RELATION BETWEEN PREDICTED AND SIMULATION VALUES **59**

PUBLICATIONS FROM THE THESIS **62**

LIST OF SYMBOLS

R^2	Coefficient of Determination
y_true	Simulated primary energy consumption
σ	Standard deviation of the each design parameters
$y_predict$	Surrogate model's prediction for primary energy consumption
E_{cons_fuel}	The annual energy consumption based on type of fuel
E_{PV}	The annual amount of energy generated by the photovoltaic system
μ	The mean
X	The original value
$f_{p,fuel}$	The primary energy conversion coefficient by fuel type
$f_{p,PV}$	The primary energy conversion coefficient related to the electricity generated by the PV system
Z	The standardized value

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
BPS	Building Performance Simulation
BPNN	Back Propagation Neural Networks
CART	Classification and Regression Trees
CNN	Convolutional Neural Network
CV	Coefficients of Variance
CV	Cross Validation
DT	Decision Tree
DNN	Deep Neural Network
DoE	Design of Experiments
EBT	Ensemble Bagging Trees
FFNN	Feed Forward Neural Network
GPR	Gaussian Process Regression
GBRT	Gradient Boosted Regression Trees
IID	Independent and Identically Distributed
KPI	Key Performance Indicator
LR	Lasso Regression
ML	Machine Learning
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Squared Error
MLP	Multi-layer Perceptron
MARS	Multivariate Adaptive Regression Splines
NCA	Neighborhood Component Analysis
PEC	Primary Energy Consumption
PV	Photovoltaic
RBF	Radial Basis Function
RF	Random Forest

RNN	Recurrent Neural Networks
RMSE	Root Mean Square Error
SCOP	Seasonal Coefficient of Performance
SEER	Seasonal Energy Efficiency Ratio
SHGC	Solar Heat Gain Coefficient
SRM	Structural Risk Minimization
SVC	Support Vector Classification
SVM	Support Vector Machine
SVR	Support Vector Regression
WWR	Window To Wall Ratio

LIST OF FIGURES

Figure 2.1 Multi-layer feed forward neural network 14
Figure 2.2 Decision tree algorithm 15
Figure 3.1 Surrogate model development for building performance prediction 21
Figure 3.2 Settlement form (a) axonometric view (b) 23
Figure 3.3 The square floor plan used by TOKI(a) model building floor plan(b) 23
Figure 3.4 Workflow of simulation-based data generation 26
Figure 3.5 Histogram of primary energy consumption for DS1 30
Figure 3.6 Histogram of primary energy consumption for DS2 30
Figure 3.7 Histogram of primary energy consumption for DS3 31
Figure 3.8 Histogram of primary energy consumption for DS4 31
Figure 3.9 Correlation map of building design parameters 32
Figure 3.10 Design parameter value distributions of DS1 33
Figure 3.11 Design parameter value distributions of DS2 33
Figure 3.12 Design parameter value distributions of DS3 34
Figure 3.13 Design parameter value distributions of DS4 34
Figure 3.14 Histogram of primary energy consumption for DT.01 downsampled train data sets a)DS1 b)DS2 c)DS3 d)DS4 35
Figure 3.15 Histogram of primary energy consumption for DT.02 downsampled train data sets a)DS1 b)DS2 c)DS3 d)DS4 35
Figure 3.16 Histogram of primary energy consumption for DT.04 downsampled train data sets a)DS1 b)DS2 c)DS3 d)DS4 36
Figure 3.17 Histogram of primary energy consumption for DT.08 downsampled train data sets a)DS1 b)DS2 c)DS3 d)DS4 36
Figure 3.18 Histogram of primary energy consumption for downsampled test data sets a)DS1 b)DS2 c)DS3 d)DS4 37
Figure 3.19 Error distribution of interpolation for all the train sizes 41
Figure 3.20 Error distribution of extrapolation for all train sizes 44
Figure 3.21 RMSE for building energy performance prediction against number of total number of samples used for training 45
Figure A.1 Relationship between number of trees and RMSE 56
Figure C.1 Scatter plots of simulation and predicted values of interpolation for 10% and 20% train sizes 59
Figure C.2 Scatter plots of simulation and predicted values of interpolation for 40% and 80% train sizes 60

Figure C.3 Scatter plots of simulation and predicted values of interpolation for all the train sizes 61

LIST OF TABLES

Table 3.1 Characteristics of the climate region and model building 24
Table 3.2 Characteristics of the residential model building elements 24
Table 3.3 Characteristics of the design space input parameters 28
Table 3.4 Features of the specified glazings 29
Table 3.5 Dataset splits 32
Table 3.6 Performance comparison based on DS1 test data for %10 train size 39
Table 3.7 Performance comparison based on DS1 test data for %20 train size 39
Table 3.8 Performance comparison based on DS1 test data for %40 train size 40
Table 3.9 Performance comparison based on DS1 test data for %80 train size 40
Table 3.10 Performance comparison based on DS3 test data for %10 train size 42
Table 3.11 Performance comparison based on DS3 test data for %20 train size 42
Table 3.12 Performance comparison based on DS3 test data for 40% train size 42
Table 3.13 Performance comparison based on DS3 test data for 80% train size 42
Table A.1 Detailed neural network architecture 55
Table A.2 Random forest model architecture 56
Table A.3 Support vector machine model architecture 56
Table B.1 Relation between the hyperparameters of ANN and RMSE 57
Table B.2 Relation between the hyperparameters of RF and RMSE 58
Table B.3 Relation between the hyperparameters of SVM and RMSE 58

ABSTRACT

Surrogate Model Approaches For Predicting Building Energy Performance

Ayşe Rüya EFE

Department of Architecture
Master of Science Thesis

Supervisor: Assoc. Prof. Suzi Dilara MANGAN

Urbanization and industrialization accelerate climate change by increasing greenhouse gas emissions and resource consumption, necessitating more efficient urban energy policies and building designs. Accurate energy demand predictions and evaluations of building designs are critically needed to reduce environmental impact and improve sustainability, given that the building sector is a significant consumer of energy and emitter of greenhouse gases. The current building sector process often encounters the iterative trial-and-error method. As building performance targets become more ambitious due to these challenges, finding optimal solutions becomes even more significant. In addition, design parameter alternatives increase over time, consequently increasing the number and complexity of potential solutions. Thus, evaluating different design alternatives becomes more challenging. Building performance simulations should integrated into the decision-making phase in order to identify the effect of early design stage determinations on building performance. However, conducting millions of simulations to evaluate high-dimensional design spaces developed based on many different design variables brings the problem of an intensive computational load. This problem, which is called the 'curse of dimensionality', can be solved with surrogate models. This study investigates surrogate model approaches that will provide rapid and effective feedback to the decision-making process. For this purpose, we use datasets created by sampling high-dimensional design space whose energy consumption is simulated by EnergyPlus. Different design alternatives in terms of the envelope of the building, energy and renewable energy systems for typical multi-family apartment block within a hypothetical settlement form were evaluated. Building performance prediction analyses were carried out using three prediction models based on different algorithms (Artificial Neural Network(ANN), Random Forest(RF), and Support Vector Machines(SVM)) for this data set. The distribution of training and test sets was also considered with these algorithms,

and the interpolation and extrapolation capabilities of the models were evaluated. Results show that the total computational cost of exploring the design space can be decreased significantly without any considerable accuracy degradation. Furthermore, cross-validation shows that the variation of performance metrics between models is very low which signals the reliability of these approaches.

Keywords: Building performance simulation, machine learning, building energy performance prediction, surrogate model.

ÖZET

Bina Enerji Performans Tahmininde Vekil Model Yaklaşımları

Ayşe Rüya EFE

Mimarlık Anabilim Dalı
Yüksek Lisans Tezi

Danışman: Doç. Dr. Suzi Dilara MANGAN

Kentleşme ve sanayileşme, doğal kaynak tüketimini ve sera gazı emisyonlarını artırarak iklim değişikliğini önemli ölçüde hızlandırmakta, daha verimli kentsel enerji politikaları ve bina tasarımlarını zorunlu hale getirmektedir. Bina sektörü, enerji tüketiminde ve sera gazı emisyonlarında önemli bir paya sahip olduğundan, sürdürülebilir bir gelecek için enerji talebinin tahmini ve bina tasarımlarının değerlendirilmesi büyük önem taşımaktadır. Günümüzde mevcut bina üretim süreci genellikle yinelemeli deneme-yanılma yöntemine dayanmaktadır. Fakat çevresel koşulların etkisiyle iddialı hale gelen bina performans hedefleri bu yaklaşımın uygulanabilirliğini yitirmesine sebep olmuş ve bununla birlikte optimal tasarım çözümlerini bulmanın gerekliliği giderek kendini göstermiştir. Ayrıca, tasarım parametre alternatifleri zamanla artmakta ve bu durum, potansiyel çözümlerin sayısını ve karmaşıklığını artırmaktadır. Bu nedenle, farklı tasarım alternatiflerini değerlendirmek daha da zorlaşmakta ve erken tasarım aşamasında alınan kararların bina performansı üzerindeki etkisini belirlemek için bina performans simülasyonlarını karar alma sürecine entegre etmek gerekmektedir. Ancak, birçok tasarım parametresi içeren yüksek boyutlu tasarım alanlarını değerlendirmek için milyonlarca simülasyonun yapılması, yoğun bir hesaplama yükü problemini beraberinde getirmektedir. Literatürde "boyutsallığın laneti" olarak adlandırılan bu durum, karar alma sürecine hızlı ve etkili geri bildirim sağlayabilecek olan vekil modeller ile aşılabilmektedir. Bu çalışmada, bina enerji tüketimi EnergyPlus tarafından simüle edilen yüksek boyutlu tasarım alanı örneklenerek oluşturulmuş veri setleri kullanılmıştır. Tipik çok aileli apartman bloğu için bina kabuğu, enerji sistemleri ve yenilenebilir enerji sistemleri açısından farklı tasarım alternatifleri değerlendirilmiştir. Bu veri seti için farklı algoritmalara (Yapay Sinir Ağı (YSA), Rastgele Orman (RO) ve Destek Vektör Makineleri (DVM)) dayalı üç tahmin modeli kullanılarak bina performans tahmin analizleri gerçekleştirilmiştir. Ayrıca, algoritmalarla eğitim ve test setlerinin dağılımı da dikkate alınmış ve modellerin

interpolasyon ve ekstrapolasyon yetenekleri değerlendirilmiştir. Sonuçlar, tasarım alanını keşfetmenin toplam hesaplama maliyetinin, belirgin bir doğruluk kaybı olmadan önemli ölçüde azaltılabileceğini göstermektedir. Ayrıca, çapraz doğrulama ile elde edilen performans ölçütlerindeki varyasyonun çok düşük olması, bu yaklaşımların güvenilirliğine işaret etmektedir.

Anahtar Kelimeler: Bina performans simülasyonu, makine öğrenmesi, bina enerji performansı tahmini, vekil model.

1 INTRODUCTION

1.1 Motivation and Background

In the 21st century, climate change is caused by the engagement of the population in urban areas and the increase in greenhouse gases, which is a significant result of industrialization. the current global urban population stands at 55%. The United Nations predicts that by 2050, approximately 68% of the world's population will live in urban areas [1]. This increase creates major problems such as environmental (excessive use of natural resources, environmental pollution, and lack of green areas), housing (increase in housing prices, lack of housing), and standard of living (increase in living standards as a result of modernization). According to IEA 2023 data, the building sector has a significant percentage when analyzing the sectors responsible for energy consumption (34%) and greenhouse gas emissions (37%) in many countries [2]. Therefore, an increase in energy demand has emerged, which led city governments to plan and gradually adapt their energy policies and strategies and energy-efficient approaches [3]. In order to achieve these requirements, existing and potential building designs need to be evaluated. Also, the energy demand needs to be predicted so that it can be reduced.

Building energy consumption estimation is calculated at the early design stage with Building Performance Simulation (BPS) tools such as EnergyPlus [4], eQUEST [5] or TRNSYS [6]. However, as the challenges increases, the number of building design parameters also increases, and building energy performance targets become more ambitious. As building design parameters increase, the volume of design domain space expands exponentially. This means that millions of simulations should be run to cover high dimensional design space accurately [7]. This problem is referred to in the literature 'curse of dimensionality' [7, 8] and is a significant obstacle to integrating BPS for the exploration of design space. This puts a major limitation of using BPS for design space exploration. BPS tools have advanced significantly lately but still demand considerable computational time [9]. Furthermore, the complexity of simulation tools and the need for detailed information can be intimidating for decision-makers [10] and human expertise remains essential. As a result, it is necessary to explore Artificial Intelligence potential to meet the demands of interdisciplinary studies.

As a solution to these challenges, surrogate models or meta-models are used [7, 11]. Surrogate models can learn relations between data variables without additional building information data and predict building energy performance values for unseen parameter combinations with the learned model. They predict energy models accurately in the order of microseconds, which is significantly faster than traditional energy models, which may take minutes to hours [10]. In practice, many researchers rely on ML techniques to study the impact of various building parameters on objectives because it is easier and quicker [12]. Most surrogate models are formed using machine learning (ML) methods such as supervised learning, and they imitate the input and output relationships inherent in the data. Also, the models are more accessible to evaluate since they are based on analytical formulations that can be made explicit [7].

ML models' effectiveness is significantly influenced by distribution of training and test datasets. When the test dataset shares the same distribution as the training dataset, it is called interpolation, and when it does not, it is called extrapolation [13]. ML algorithms performs well when interpolating because the hypothesis based on identical distribution, but when predicting out of data range models mostly shows high-error predictions [14]. This thesis also focused on extrapolation approach by a case study. The case study building model was trained using two different values (30%, 40%) of the window-to-wall ratio (WWR) parameter, which significantly affects the building energy performance, and later trained model estimated 50% WWR simulation value outputs to ensure extrapolation capacity of the model.

1.2 Research Questions

Taking into account the study gaps identified in the earlier section, this research addresses the questions below:

RQ1: Is it possible to use BPS for feasible design space exploration with the help of ML methods? How much performance gain can be obtained using surrogate models, and what is the compromise?

RQ2: How does the distribution of training and test datasets affect the performance of ML-based surrogate models, particularly under interpolation and extrapolation scenarios?

RQ3: How does varying training sets' size impact the ML model's performance in estimating energy performance of buildings?

RQ4: Which ML techniques are more suitable for creating surrogate models?

1.3 Research Aim and Objectives

This thesis aspires to create an accurate and quick prediction method for building energy performance using advanced AI techniques. To ensure that ML techniques computationally feasible, this thesis aims to develop surrogate model for rapidly also accurately assessing buildings' energy performance.

Objective 1: To develop supervised ML-based surrogate models with a synthetic database for estimation of primary energy performance of building

Objective 2: To explore the efficacy of surrogate models in addressing the 'curse of dimensionality' in building performance simulation

Objective 3: To investigate how the data distribution affects the implementation of machine-learning-based surrogate models

Objective 4: Suggest a wise method for training and testing the speed rate of the process

Objective 5: Assessing building energy implementation for ML models with an application on a case study

1.4 Thesis Methodology and Structure

This study dealing with the high dimensionality of the design space created by BPS, leading to the curse of dimensionality, which complicates the evaluation of building energy performance. To tackle this issue, a surrogate model method based on ML is offered to support performance-based design decisions as an innovative solution. The overall methodology of this thesis can be outlined in four steps;

Chapter 1 introduces the research problem identification by tendering motivation and background for study. Based on the knowledge gaps identified, the study's questions are established.

Chapter 2 provides an overview of related studies in the literature. The chapter highlights the necessity and complexity of assessing the energy consumption of buildings and claims that surrogate models will address this problem using ML algorithms. It also explains the hypothesis of interpolation and extrapolation approaches for creating surrogate models. It also provides an essential explanation of three leading ML methods, which include artificial neural networks (ANN), random forest (RF), and support vector machine (SVM), all extensively applied in predicting the primary energy consumption of buildings.

Chapter 3 was dedicated to developing ML-based surrogate models that could predict building primary energy consumption from BPS in the early design stages with a case study. The simulation databases were employed to train and test surrogate models. Later, the model results were presented and discussed.

Chapter 4 provides a thesis overview, emphasizing its substantial contribution to knowledge and practical implications. We delve into the potential and significance of the research by reviewing research questions. This chapter demonstrates the feasibility of the offered method and its potential to building energy performance prediction. Moreover, it offers valuable recommendations for future research, laying the groundwork for further advancements in the field.

2
BUILDING ENERGY PERFORMANCE AND SURROGATE MODEL

The need for more efficient, better-designed buildings is increasing, as mentioned in Chapter1. Decision-makers have responded to this need by proposing "performative" designs and more performant design tools. These design tools often include parametric modeling, simulations, and sometimes optimization [15]. Parametric modeling digitally models mathematical relationships and can control design parameters by creating a series of design iterations. This way, it is possible to examine design spaces for various design combinations. Woodbury et al. [16] highlighted the need for computational design tools that support design space exploration. However, design space exploration, an activity that uncovers design variations, is often limited due to its time-consuming nature. Even on today's computers, simulations take a long time, and evaluation becomes difficult [15].

Surrogate models belong to the most widespread category of meta-models, seen as a solution to this challenges [7, 11]. They are widely used in numerous disciplines, and the literature offers specific techniques in many variations. Also, most of these techniques are accessible through various toolboxes and are implemented in high-level programming environments such as MATLAB [17], Python [18], RStudio [19]. This chapter presents a literature review of existing knowledge on surrogate (or meta) modeling, followed by an assessment of ML algorithms for estimating building energy performance.

2.1 Building Energy Performance

The building design process begins in the early/conceptual design stage, and decision-makers derive design concepts. At this stage, the design concept is highly flexible and can be changed quickly and radically [11] so that building energy performance can be considered. Improving building energy performance at the early stage is essential for sustainability and creates opportunities to ensure energy efficiency, prevent waste, and save resources. Building energy performance is compared through key performance indicators (KPIs), generally consisting of primary energy consumption, electricity consumption, lighting, CO_2 emissions, heating, and cooling loads [20]. Improving KPIs creates the most significant impact on building energy performance [21].

In order to evaluate and ensure building energy performance criteria, the energy consumption value must be calculated [22]. There are three main different approaches to predict building energy performance, physical models ("white box") [23], hybrid methods ("grey box") and data-driven approaches ("black box"). Physical models can simulate and forecast actual implementation of buildings based on detailed physical properties of building materials and characteristics [24], as long as calibration is completed correctly [25]. Simulation of physical models are carried out with BPS tools. Advanced simulation tools such as EnergyPlus [4], eQUEST [5], or TRNSYS [6] are used to evaluate energy performance [24]. BPS tools use a mathematical model to predict energy flows and building loads dynamically. Ostergard et al. [7] explained that BPS provides a comparative and informative calculation, helping decision-makers in design space exploration for energy consumption evaluation. Thomas et al. [26] described that BPS ensures design parameters fulfill building criteria to evaluate building performance and have systems to predict the output for a given input(s). Decision-makers often rely on modeling and simulation to test various design strategies to guide the design of buildings. At the same time, they put in necessary efforts without compromising indoor environmental conditions, which impact occupants' health and overall well-being [10].

2.1.1 Complexity of Building Performance Simulations

Integrating BPS into design space exploration allows decision-makers to receive a very high degree of accuracy of prediction feedback for the building design process [27]. However, there are many drawbacks in the process of creating and developing BPS models. Firstly, decision-makers need powerful computational resources because simulations can take a second to several days, weeks, or even months, depending on the model's complexity. Despite their potential, computational intensity might be a limiting factor [28] and may interrupt the creativity process [11]. Shortening calculation time by reducing the model fidelity is not an option, which might lead to less reliable results [28]. Since building parameter values are uncertain and variable, millions of high-fidelity energy models need to be simulate. Probabilistic energy predictions involve defining uncertain parameters by their possible value ranges and sampling these to create multiple energy models. These models are then simulated to predict a range of possible energy demands, accounting for variability and uncertainty [29]. The high-dimensional design space created by millions of simulation called the "curse of dimensionality" [7] Lastly, the long computation times makes it difficult to explore design space [30]. When consider all the disadvantages, researchers have stated that simple and fast energy estimation tools are needed to evaluate and compare building design scenarios [7].

2.2 Surrogate Modeling

Researchers have proposed using surrogate models or also known as meta-models [11] to overcome computational challenges that is created by high-dimensionality of simulation-based methods. Surrogate models can emulate high-fidelity models, in that case, an expensive dynamic building simulation model. Surrogate models act as a bridge to simplify communication between BPS and decision-makers. The intent of surrogate model is to obtain approximate results to simulated data and provide high-fidelity without running all of the expensive simulation codes called primary sources. Forrester et al. [8] explained that BPS tools can take a long to run millions of simulations due to computational intensity so the surrogate model is an alternative model that allows results to be predicted quickly (< 0.01 s) and can provide rapid point and distribution estimates of the building performance effectively. Ali et al. [10] stated that surrogate model is a low-computational alternative created by training a mathematical model to complement the capabilities of BPS and expand their applications. Van Gelder et al. [28] explained that instead of time-consuming, expensive-coded simulation models, surrogate models can be considered because they can mimic the original complex simulation models. Edwards et al. [9] mentioned that surrogate models provide fast approximations to the actual model and require low computational resources. According to stated benefits, surrogate models allows decision-makers to quickly evaluate the design concept, provide immediate feedback and investigate the design space sufficiently [7].

2.2.1 Machine Learning

There are several methods for estimate building energy performance: engineering, statistics, ML methods [31]. ML, branch of artificial intelligence, is extensively used among these approaches. It enables tool to learn from historical or simulation data to forecast future performance without the need for explicit programming [23]. ML methods can handle more complex and nonlinear relationships between variables, allowing for more accurate predictions. ML methods are more general and adjustable modeling techniques for predicting [25].

Using supervised learning, which also overlaps with the regression analysis procedure, is as follows: the model is trained with inputs (design parameters) and desired outputs (design targets); the outputs are known and labeled as data. Once trained, the system can predict the values of unknown output variables for other design combinations [22]. ML models trained with supervised learning can capture complex interactions with low prediction errors, even though the relationship between input and output parameters is not defined. Like BPS tools, models

perform as black box [7, 22]. In other words, it does not require any information about building systems, it can predict various unknown values in a very short time and with a high level of accuracy by using historical [23] or simulation data, which usually have a high computational load [32].

Interpolation and Extrapolation Approaches

A dataset is needed for train the models and another to characterize their performance in forming ML models. The dataset used for training the models is named the training dataset, while the dataset used for characterizing the performance is called the test dataset [33]. The distribution of the training and test datasets over the design space profoundly affects the model's performance. In this manner, two scenarios characterize the model performance. If the test dataset is sampled from the same distribution as the training dataset or sampling from a distribution from a more challenging test dataset. Testing the model with a dataset that has the same distribution as the training dataset is called interpolation, while the latter is called the extrapolation [13].

ML model predicts variables beyond the ranges of training data, in fact, this is seen as one of the important limitations of ML predictions and needs to be clarified to see what extent the ML model can be utilized beyond the training design cases [32]. Most ML algorithms are built on the independent and identically distributed (IID) hypothesis, which hypothesizes that the data sampling is independent and the training and testing sets are identically distributed. ML algorithms perform well when maintaining the IID hypothesis, but the training and testing sets are not identically distributed when extrapolating [14]. The main concern is that prediction models may not perform well outside of their training range. For instance, a model trained on a limited dataset, such as data from a small group of buildings, may struggle to perform accurately when applied to different contexts. In order to perform well, the data set used for training must be representative of the range of applications and contain high quality data, but comprehensive research that includes sufficient diversity may take time and effort [32]. However, extrapolation can be improved by teaching the model the underlying generalizable functional relationships of the physical principle model instead of memorizing the values of target variables. Singaravel et al. [34] have tested to predict energy performance of test set that has different number of floors than training set with deep learning. Zhou et al. [14] proposed a combination of two-level (interpolation and extrapolation) model that predicts the large-scale performance of unknown input parameter combinations.In the first step, the authors predicted high-accuracy small-scale performances with interpolation. Later, with these predictions, they

generated scalability models, which is referred to as extrapolation. Robinson et al. [35] suggested generalized method; conducted training with national survey data, validated with authentic United State (US) city benchmarking data and predicted the performance of the US commercial real estate buildings. The authors considered five common features for traning so that wide range of areas can use the model without requiring however, the data can is only available for the US country in which they were trained.

2.3 Surrogate Model for Building Performance

Generalization in ML refers to a model's ability to accurately predict outputs on new, unseen data, that is not part of the training set. Several techniques can enhance generalization, including preprocessing techniques, selecting an appropriate learning algorithm, and fine-tuning hyperparameters [36].

2.3.1 Data Collection

There are two main ways of obtaining data for training surrogate models. One way is to use data obtained from the observations (actual data), e.g. utility bills, energy consumption surveys, and energy consumption statistics and reports [32, 37]). However, collecting detailed real data comes with challenges and is sometimes unavailable. In this case, another option is to create models using BPS. In the literature, surrogate models are generally created from a large set of simulations made with validated and detailed BPS, which is often computationally heavy but most cost-effective method when time considered [9]. They are widely used for the early predictions and when physical building experiments cannot be performed. Unfortunately, this approach still requires some BPS time, but the computation can be decreased significantly with the surrogate models [7]. Some researchers had access to real building feature datasets. Neto et al. [38] compared performances of ML model and EnergyPlus simulation predictions based on real-world data. The results revealed that ML model performed better at prediction.

2.3.2 Data Preprocessing

Data preprocessing is critical in ML methods; ensures input data is clean, organized and ready for analysis [39]. Also it means cleaning, integrating, transforming or reduction of the data. Datasets are usually contain errors, for instance, missing, duplicated, inconsistent, noisy data and outliers. Missing values can be replaced with using mean, median, or mode [40]. Data cleaning is the part where incomplete, incorrect, inaccurate, irrelevant, and/or noisy parts of the data is detected and corrected. Zekic-Susac et al. [41] performed data preprocessing and explained in detail; replaced missing values and removed outliers from the data

and some of the records because of human made data entry. Amasyalı et al. [33] used neighborhood component analysis (NCA), to select the discriminating and non-redundant features. Data integration is the part that datasets are taken from different sources and combined [39]. Data transformation is converting raw data into form that can be easily interpreted by the ML methods, including encoding categorical variables, scaling (normalize or standardize (2.1) numerical ones [40], and managing imbalanced data to avoid biased models through oversampling and undersampling. Ahmad et al. [42] normalized input and output values in order to improve surrogate model accuracy. Lastly, data reduction is the process of decreasing the dimensionality of a dataset. As a result, this method helps to identify patterns more easily and improve the prediction accuracy for outputs [32].

$$Z = \frac{X - \mu}{\sigma} \tag{2.1}$$

Where X is the original value, the mean is μ, σ is the standard deviation of the each design parameters. Z is the standardized value.

Simple but helpful descriptive statistical analysis visualizations of especially high-dimensional design spaces, can be used for present the data meaningfully and concisely [43]. The method includes graphical data visualizations through scatterplots, boxplots, charts, and histograms. Histogram plots shows the symmetry of the data distribution, revealing whether the plot follows the Gaussian (normal) distribution [12]. Scatter plots show the connection of both linear and non-linear correlations between two numerical variables, data is mostly normalized to interpret easily. Boxplots show the smallest and largest values, representing the distribution of a dataset [37, 39]. Also, the authors suggest using the parallel coordinate plot with histograms to investigate the design space fast [44]. Quantitative sensitivity analysis methods can be computed (by utilizing either Pearson's product-moment correlation or Spearman's rank correlation) to consider monotonic, non-monotonic, linear, and asymmetric variable correlations [11].

Data format of inputs and outputs must be appropriate for the model. Categorical variables can be used if a feature cannot be described with continuous numerical ranges or a linear relationship between target variable is not expected [39]. Different methods exist for operating categorical variables, such as one-hot encoding and label encoding [45]. Each option in a category feature is re-encoded into "dummy" binary features if using one-hot encoding. If the option is 1, it is selected, if the option is 0, it is not [11]. Amasyalı et al. [33] used one-hot encoding to convert the categorical variables to numerical.

Data preprocessing and developing predictive models require programming tools. There are several software packages, such as MATLAB, providing various ML models easy to use [46]. Ali et al. [40] analysed that 24% of the papers used the Python tool, 18% used the R tool, and 13% preferred MATLAB. Also, Seyedzadeh et al. [46] advised using for simple models, such RF or SVM, a sophisticated programming, for example Rstudio and Python. Robinson et al. [35] implemented all models with scikit-learn Python library and use the default parameter settings. Gelder et al. [28] applied all algorithms with MATLAB toolboxes.

2.3.3 Sampling Strategies

Another important matter addressed is the sizes of the data. In the early design stage, the main idea is to run a large set of BPS that possibly multidimensional represent design space [44]. A small dataset may not fully capture the data, whereas a large dataset demands significant computational resources [32]. In such a case, sampling should be done considering data sets. If design parameters are divided into discrete values with equal probability, design combinations could be investigated with full-factorial sampling [47]. However, the required simulations grow exponentially with the number of design variables and quickly becomes an overwhelming task [44]. Sampling strategies, also known as design of experiments (DoE), aim to select points within the design space that maximize knowledge for each simulation while decreasing the duration required for sampling [11].

Randomness in data splitting can create bias, and poorly learned observations by the ML can dominate the selected test dataset. A n-fold CV procedure is used to dealing with many issues, ensure more solid outcomes, and balance bias and variance. Ali et al. [40] stated that 10-fold CVs are commonly applied but there are other types in the literature, 4 folds, 5-fold, and 8-fold CV [40]. For example, 10-fold CV means the training data is divided into 10 equal parts. The model is trained 10 times, each time using 9 parts for training and 1 part for validation. The performance metrics from each fold are averaged to give an overall performance measure. It is essential to validate models and evaluate how the model will generalize to an independent dataset and prevent overfitting. Ostergard et al. [7] mentioned that using CV or bootstrapping is beneficial especially if the data is scarce. Ali et al. [40] mentioned that there are some other validation techniques that authors used, nested CV, and leave-one-cross-validation.

Westermann et al. [11] reviewed that the 28 majority of papers used Latin hypercube sampling, 81% used a static sampling technique compared to 19% that used an adaptive sampling technique. In static sampling, all sample locations

are defined in one shot before model fitting, making general ML method precise for entire design space. Standard methods include pseudo-random sampling like Monte Carlo sampling, quasi-random sampling like Hammersly, Halton, or Sobol's sequences, and stratified pseudo-random sampling like stratified Monte Carlo sampling, Latin-hypercube sampling (LHS), or orthogonal array sampling. Gelder et al. and Østergård et al. [7, 28] used Monte Carlo sampling in their studies. Amasyalı et al. [33] used conditional Latin hypercube sampling (cLHS) method.

The algorithms that perform the best are unclear; the outcome depends on the number of variables and samples. In literature, LHS is the most applied sampling scheme [11]. Latin Hypercube Sampling creates random input values evenly distributed between 0 and 1 used in conjunction with the feature ranges to create a random set of training examples [8].

2.3.4 Train/Test Splits

The datasets are divided into training and testing. In the literature, there is not a certain rule for how to divide these data sets and no consistency between studies. This section offers an overview of the sample sizes and the techniques employed to divide the data into different sets. Westermann et al. [11] mentioned that, typically, distribution of 20% data points randomly for test dataset is appropriate. Olu-Ajayi et al. [39] randomly split the data and the training group containing 70% of total data. Gelder et al. [28] chose a maximin Latin Hypercube sampling scheme to create training and validation samples, 500 samples in total for model training and CV. Ali et al. [40] mentioned in the review, common ratio is 80-20 followed by 70-30 and 75-25, train and test set, respectively. Also indicate that ML methods typically require 100–1000 samples. Amasyalı et al. [33] trained the model with 1000, 2000, 5000, 10,000, 20,000, 50,000, 100,000, 200,000, 500,000, and 1,000,000 data instances and split into training (65%), validation (10%) , and testing datasets (25%).

2.3.5 Machine Learning Models

Most of the studies in the literature involves multiple models applied at the same time to solve a problem. The authors [7, 48] mentioned about how difficult choosing the most suitable ML model and there is not a best model for every data set. Therefore, the choice of method depends mainly on the problem and analysing the data. In order to understand which model perform better in specific problem, the authors suggest to try several methods [42]. Widely employed three primary algorithms for the prediction of energy consumption for supervised learning are Artificial Neural Network(ANN), Random Forest(RF), and Support

Vector Machine(SVM) [39]. According to Amasyalı et al. [32] review, 47% and 25% of the studies used ANN and SVM, respectively, to predict building energy consumption. Also, just 4% of the studies examine decision trees. Seyedzadeh et al. [22] and Ahmad et al. [42] revealed that ANN is the most common algorithm for building energy prediction.

ANN is a non-linear problem solver algorithm that is also good at discovering complex hidden relations between inputs and outputs [23]. The typical form of ANN has three successive layers: an input layer that trains the model, a hidden layer that combines input and output layers, and an output layer that gives the outcome [49], as illustrated in Figure 2.1. There are several subsets of neural networks, such as Feed-forward Neural Networks(FNN), Back Propagation Neural Networks(BPNN), Recurrent Neural Networks(RNN). Feed-forward networks are the most applied from all these algorithms in the literature [39]. A Multi-layer Perceptron(MLP) is a type of DNN that employs a feed-forward propagation mechanism, featuring one hidden layer where latent and abstract features are obtained. [39]. A significant disadvantage of ANNs is their black-box nature, i.e., lack of transparency in physical performance. On the other hand, black-box systems do not require knowledge about the building system; they learn the relation between inputs and outputs using historical datasets [42]. Another disadvantage is that their tendency to overfit, i.e., performing well on specific datasets but poorly on others. When dealing with overfitting, it is important to remember that the risk increases with model complexity, especially when aiming for higher accuracy. However, modifying the structure of hidden layers during training and employing regularization techniques have proven to be successful strategies for overfitting. Another approach is to reduce input data complexity by pre-selecting impactful variables [49].

SVM is a data mining model and non-linear problem solver [39]. Kernel function is an important feature for model performance, applied if the data is not linearly separable. The model identifies the optimal (best fit) hyperplane to separate the data into two sub-datasets; the hyperplane maximizes the margin between classes defined by support vectors and transforms it into a higher-dimensional space where linear separation is possible [23]. In the literature, common kernel selections are linear, polynomial, or a radial basis function(RBF) [49], and sigmoid function. In order to the Structural Risk Minimization (SRM) principle, SVM minimizes generalization errors [25]. Also, it is not affected by data imbalance even if the data amount is limited, giving accurate results [40]. One major limitation of SVM is its substantial computation time [25].

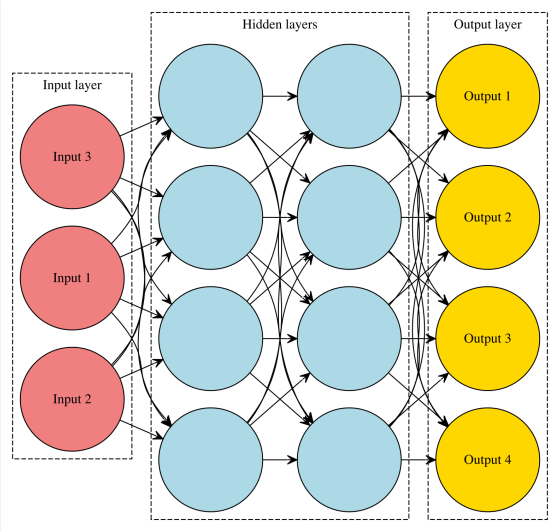

Figure 2.1 Multi-layer feed forward neural network

RF algorithm is based on Decision Tree (DT) [23] and firstly, DT algorithm should be understood. DT is applied for both regression and classification analysis for prediction. The algorithm has a tree-like flowchart structure[39, 42]. The root node is the leading node at the top of a decision tree. The root node is divided into smaller sub-nodes until a specific criterion is met, and leaf nodes reach the predicted output [12]. Finally, a path can be followed according to the relationship between the data set parameters as illustrated in Figure 2.2 [49]. DTs are effortless to interpret, easily visualize, and handle the missing data values [40], but they are not very accurate or robust regarding data changes. To overcome this issue, bootstrapping techniques, such as bagging or random forests, are proposed [7].

RF is a tree partitioning method with a principle called ensemble learning, which randomly combines weak predictions to become a strong learner [50, 51]. RF grows with multiple trees and trained independently with random samples to make their prediction. The final prediction result averaged all of these predictions [23]. RO algorithm has superiority since each tree structure is trained on multiple different data; it prevents the data from overfitting [40, 51], and thus, the prediction results have high accuracy [12] and stability [41]. RF model provides some benefits; there is no need to perform feature reduction and feature normalization [14], and it can overcome unbalanced and missing data [51]. As a disadvantage, it has a high computational time when created with large data sets. [7].

2.3.6 Hyper-parameter Tuning

ML model characteristics are determined by parameters. However, before the training process, the best performance hyper-parameters would found for finding optimal parameters. This process is called by the authors "hyperparameter

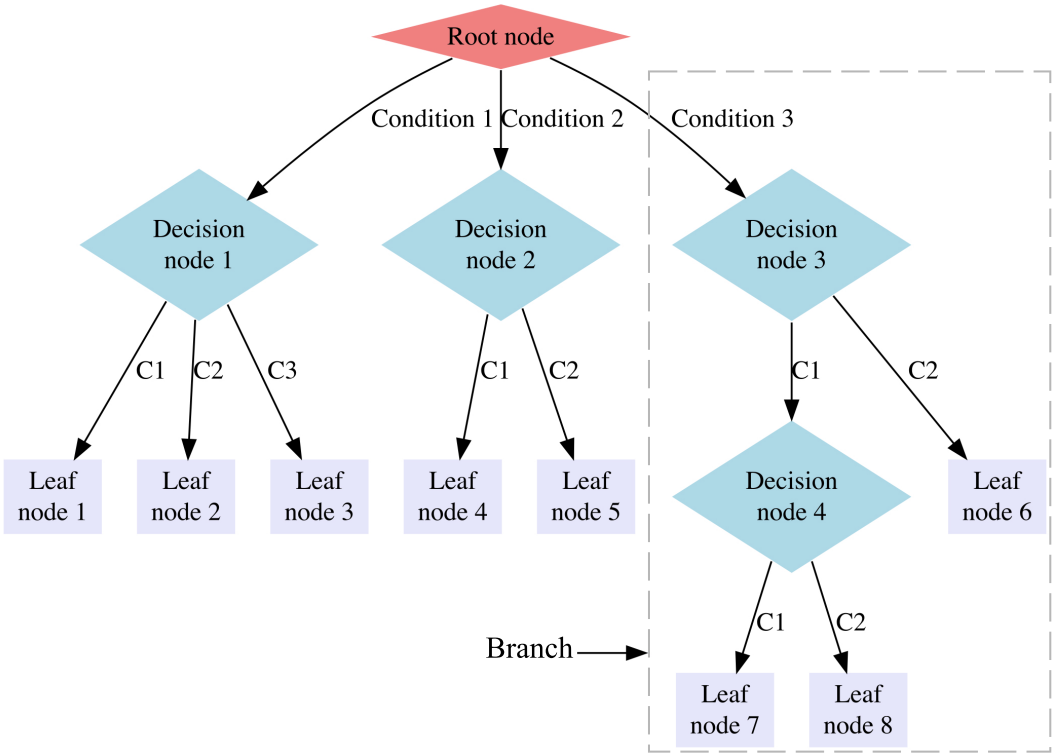

Figure 2.2 Decision tree algorithm

optimization" or "tuning" [51]. Hyper-parameters enable the algorithm to effectively map input features to the desired targets, leading to the acquisition of meaningful insights. The 'hyper' means that model parameters that are determine and control the learning capacity, are at higher-level. Hyper-parameters are set manually in the first place before the training of the model and cannot change itself during training process [52]. It should be noted that a parameter is a variable trained on a data set that makes predictions based on new data. The hyper-parameter is set before the training process starts and controls the behavior of the learning algorithm. Unlike parameters, hyper-parameters do not learn from data but are set by the user or optimized [51].

ML requires substantial technical expertise and models handled with several software packages mostly have default values for hyper-parameters in the ML library [40] (these packages mostly use Bayesian methods [52]), which helps to provide quick and easy results when considering their predictive performance. However, tuning hyper-parameters is still critical as it controls ML models accuracy and consistency. It involves evaluating the model's performance across various combinations of hyper-parameter values and selecting the models with the lowest error. Multiple different arrangements are compared for to choose the hyper-parameters [11]. Grid (Cartesian hyperparameter) and random search are the two standard fine-tuning methods. Grid-search investigates whole possible

combinations which is exhaustive search, the motivation is not find the optimal configurations. However, this method suffers from the curse of dimensionality [51], consumes so much time and not applicable if the setting are not done properly [7]. In order to deal with high-dimensionality, random search proposed. It evaluates just a predefined number of combinations, which are only a few of them, to find equally good or better models. Bergstra et al. [53] mentioned that random search explores effectively and finds superior combination with the same computational budget. Another effective optimization algorithm is the Bayesian [51, 52]. These methods belong to a category of sequential model-based optimization algorithms, enabling us to leverage the outcomes from previous iterations to enhance our sampling approach for subsequent experiments [51]. Seyedzadeh et al. [46] carried out comprehensive study for tuning models and used grid-search coupled with k-fold CV for choosing best hyper-parameters and predicted the loads of cooling and heating. According to the results, hyper-parameter tuning become more crucial if the data set size increase. Ali et al. [40] reviewed that genetic algorithms, step-wise search method, and shuffled frog leaping optimization approach is rarely applied.

ANN types have similar hyper-parameters because their architecture is also similar. Since the model complexity mainly depends on the number of hidden layers and the number of hidden neurons per layer, these parameters should be tuned. Hyper-parameter size specifies the number of units (neurons) in the hidden layer and based on defined numbers, model will be trained and assessed. Weight decay helps to prevent overfitting. If the output of the model is continuous, especially for the regression problems, linout should be used [54].

SVM algorithm require tuning these hyper-parameters, the C (Regularization parameter) controls the trade-off between misclassification of training examples and the complexity of the decision surface. The Kernel Type defines the type of hyperplane used to separate the data, with typical choices being linear, polynomial, RBF, and sigmoid. The Gamma parameter, controls the spread of the RBF kernel for model. Parameter names represent e1071 package in R [55], can be different for other programming languages.

RF requires tuning a small number of hyper-parameters compared to other ML algorithms [42] for reaching optimum accuracy [41]. The first parameter; independent tree numbers in the forest, which refers to ntree parameter. A trade-off exists between the model's accuracy and the computational time. The second parameter is mtry, which is the number of parameters to consider when deciding the best split, the most critical hyperparameter. The third parameter is nodesize, the minimum size of the terminal nodes (i.e., the number of samples in a leaf) [56].

Papadopoulos et al. [57] and some authors used Tsanas's dataset [12] to evaluate tree-based ensemble learning methods. Papadopoulos applied an exhaustive grid search to tune hyper-parameters of RF model, and results showed that hyperparameters significantly improved the model compared to Tsanas' RF model, for which tuning was not discussed in his work.

2.3.7 Model Performance Analysis

While the surrogate model is not always accurate, a "confidence index" must indicate its quality [15]. Several standard metrics are used for model performance evaluation [39]. These measurements are designed to assess prediction error and accuracy. The performance and accuracy of surrogate models are generally reported in test and validation datasets. However, to determine if the model is overfitting the training dataset, it is important to evaluate and compare it to the validation/test dataset [52].

Ali et al. [40] reported that Root Mean Square Error(RMSE) used 29% of regression cases, followed by the Coefficient of Determination R-squared(R^2) 27% and Mean Absolute Error(MAE) at nearly 20%. Amasyalı et al. [32] analyzed that Mean absolute percentage squared error(MAPE) 29% and CV 41% are mainly used in the studies. Many studies use more than one metric because employing multiple performance metrics can offer deeper insights into the model's behavior. Coefficient of determination, R^2, which is a unitless measure of the global error with an optimal value of 1 making the measure practical for comparison and a measure to assess how much of the variance in the actual energy consumption levels are explained by the model. R^2 is correlated with other common metric, RMSE. RMSE is the standard deviation of the residuals between the predicted and the actual energy consumptions. The lower the MAE and RMSE and the higher the R^2, the more similar dispersions are between the predicted and the actual consumptions [46]. Robinson et al. [35] used stratified k-folds CV to see how well the model can generalize. Also, some authors used graphs to evaluate the model. They visually examined patterns in the residuals (differences between observed and predicted values) to determine whether they were normally distributed [58].

2.3.8 Model Applications

Ahmad et al. [59] reviewed the benefits and drawbacks of ANN, SVM, and hybrid-based methods. Also, the ANN is one of the most common algorithms used in building energy consumption prediction. Edwards et al. [9] used Multi-Layer Feed Forward Neural Network (MLFFNN) and Lasso regression (LR) to create EnergyPlus based surrogate models for three large-scale datasets that have 7 to

156 model inputs and 90 output variables. Models predicted an entire yearly simulation outputs in 3 seconds, greatly reduced the required simulation time. Olu-Ajayi et al. [39] carried comprehesive study and used many models including ANN, SVM, DT, LR, RF, DNN for predicting yearly residential building energy consumption using a large dataset(285,000 data points). Model results with 5000 samples showed that the most accurate model was DNN (R^2 0.95) with a 5.2 training time, following that ANN model showed also good performance (R^2 0.94) with a better training time 3.7 s. DT algorithm have the lowest training time with 1.2 s but the R^2 value was 0.74. Ahmad et al. [42] applied ANN and RF models for the HVAC electricity consumption of a hotel in Spain and found that ANN model (R^2 0.95) slightly performed better than RF (R^2 0.92). Amasyalı et al. [33] predicted the cooling energy consumption of office buildings with ANN, DNN, classification and regression trees (CART), and ensemble bagging trees(EBT) while considering occupant behavior and mentioned that the behavior of occupants highly affects energy consumption. All methods showed accurate prediction results depending on the sample data size, for example ANN model required minimum 1000 sample but EBT model need at least 5000 sample to provide good accuracy. NN models showed better performance than tree-based models but the training times were longer. Tsanas et al. [12] predicted the heating (0.5 points deviation) and cooling (1.5 points deviation) load of 768 different residential building samples with RF model with only 8 input variables. Dong et al. [60] first one to used the SVM algorithm to predicted the monthly electricity consumption of four randomly selected non-domestic buildings based on meteorological data in Singapore, which has a tropical climate. The authors indicated that SVM with the radial basis function (RBF) kernel, performed better than neural networks with the coefficients of variance (CV) evaluation metric to under 3% and percentage error within 4%, and. Similarly, Li et al. [61] used SVM algorithm to predict the hourly cooling demand for a non-residential building in China, as model input variables; using humidity, solar radiation, and outdoor temperature. Also in the study, SVM and and back-propagation(BP) neural network model compared and SVM model achieved better accuracy. Østergård et al. [7] presented a comparison of metamodeling methods and evaluated the six most used models; including NN, RF and Support Vector Regression (SVR). Despite their high dimensionality and complexities, models applied 13 cases created with BPS. The study found that Gaussian Process Regression (GPR) is the most accurate, while NN and Multivariate Adaptive Regression Splines (MARS) also perform well, achieving high accuracy ($R^2 > 0.99$) with fewer training points (32 to 256). Seyedzadeh et al. [46] applied five common ML models including ANN, SVM, RF, GPR, and Gradient Boosted Regression Trees(GBRT) for predicting loads of cooling and heating consumptions instead of

EnergyPlus and Ecotect tools. According to the results, SVM performed best due to simple structure and rapid calculations. If consideration was on the data set size, GBRT predicted most accurately for a small dataset. On the other hand, multi-layer NNs are more efficient and faster for complex data sets that require extensive energy simulations.

ML algorithms compared with different methods for building energy prediction. Neto et al. [38] compared performances of ANN trained on real data and EnergyPlus prediction of daily energy usage for commercial building with six floors. The results revealed that ANN performed better. Ali et al. [40] reviewed 56 journal articles that have applied statistical and ML techniques in building performance and found that ML algorithms performed better, especially RF and SVM methods shows goody performance in the most works.

3
SURROGATE MODEL APPROACHES FOR BUILDING ENERGY PERFORMANCE PREDICTION AND IMPLEMENTATION

Based on previous similar studies learned from the literature review, this chapter developed surrogate models using interpolation and extrapolation approaches to predict building energy performance. An identified gap in the literature review was the creation of method for building energy performance prediction using surrogate models.

3.1 Surrogate Model Development

At this stage, we will discuss how surrogate models are used to estimate residential building primary energy consumption. Different steps were considered to build a surrogate model: collecting the data, data preprocessing, and choosing a design space sampling strategy; later, division of training and testing datasets by size; selecting the model algorithm; tuning algorithm hyper-parameters; and finally, evaluating the model performance.

Figure 3.1 summarized the interpolation and extrapolation approaches used in the surrogate model development. This method clearly defined the process, allowing other researchers to compare their models quickly. Firstly, a simulation-based method was applied to generate data. Building simulations were carried out using EnergyPlus. An initial building model and dataset acquired from Mangan's study were used [62].

Since the same steps were followed for these two approaches, the method was not separated to avoid confusion. The only difference in the method was that the WWR30 dataset (DS1) was used for the interpolation approach. DS1 was internally divided into training and test sets, and the model was trained with the training data from the same dataset, and predictions were made using the test data from the same dataset. In the extrapolation approach, a dataset was randomly sampled from the WWR30 dataset (DS1) and WWR40 dataset (DS2), and then the WWR30_40 dataset (DS4) was created. However, the test data was formed from the WWR50 dataset (DS3), which the model had never seen, and predictions were made.

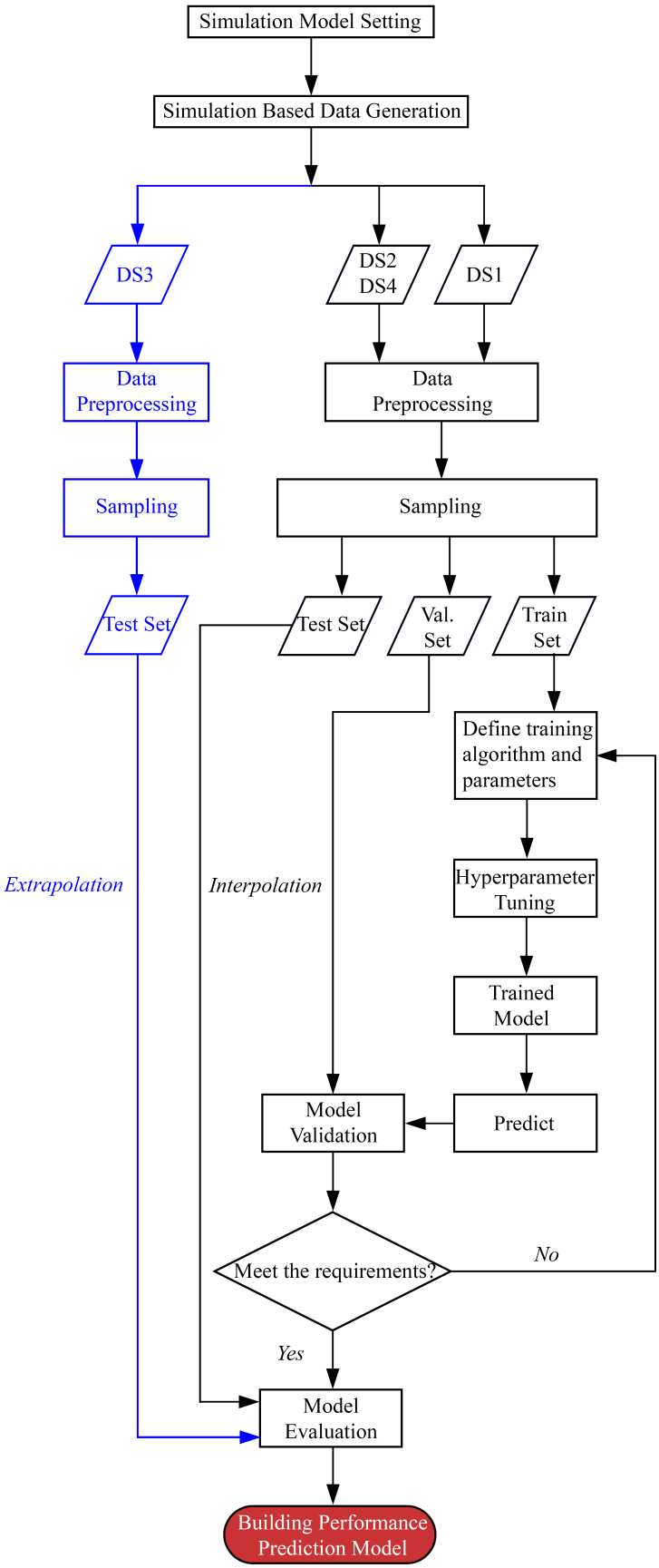

Figure 3.1 Surrogate model development for building performance prediction

3.1.1 Data Collection

The first stage of the study consisted of two main steps: setting the simulation model and generating simulation-based data. It showed how the building model was created and how the data was collected.

Simulation Model Setting

ML required a highly accurate database to create a surrogate model. If real-world data were effortlessly available, using them would have been advantageous, yet simulations were also helpful. A synthetic database was used because of the need for actual data. It was impossible to guarantee that simulation results would ultimately reflect real-world data, but simulating building performance was commonly used with that method. Thereby, even though the simulation data could have been biased, it represented real-world data with high probability. Furthermore, any discrepancy between simulated and real-world data did not affect the methodology developed.

This study examined the modeling and simulation of a residential building in a hypothetical settlement context in Istanbul. Multi-family houses (i.e., apartments) covered most of the total residential floor area, at around 80%. Istanbul was chosen as the location when creating the model building because the city was extraordinarily dynamic and rapidly developing, and it was stated that there were more than 5.4 million buildings. Urban metropolitan growth was highly experienced. Thus, there was an increasing need to solve the problem of low awareness of energy-efficient residential buildings. To close this information gap, it was beneficial to develop energy-focused solutions enabling informed decisions regarding sustainable housing production [62]. The settlement form was designed using a 3-by-3 matrix, adhering to a uniform layout of a 9-point block with identical characteristics on a hypothetical site. All blocks in the defined hypothetical site had the same properties as the model building. The model building was at the center of the matrix layout. Based on the results of Mangan et al.'s comprehensive parametric analysis, 1.00 (uniform canyon) and 90° values were taken for the aspect ratio (building height to street width (H/W)) and orientation, respectively [63]. The model building's settlement form and axonometric view were illustrated in Figure 3.2.

In defining building form parameters, the existing residential building form parameters that reflected the effect of current national regulations in housing production and included commonly utilized construction technologies and design standards were considered. Thus, mass housing investments carried out through

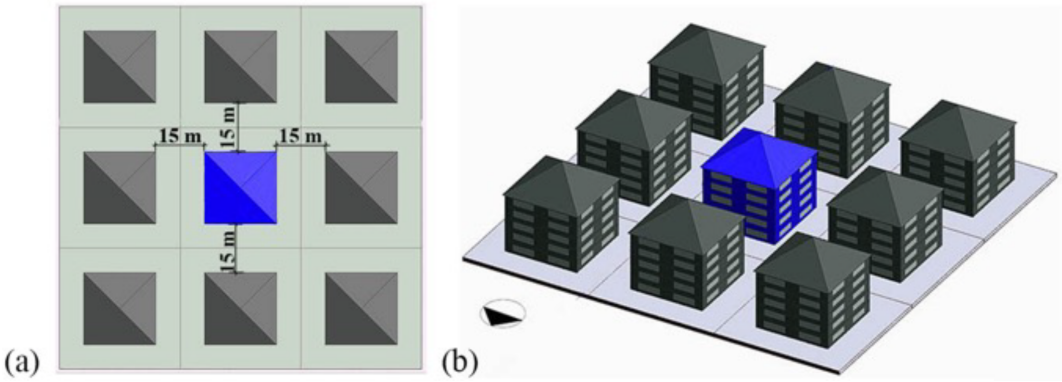

Figure 3.2 Settlement form (a) axonometric view (b)

TOKI (Housing Development Administration of Türkiye) across the entire country, a significant change in the building sector in Türkiye, were also considered. Based on this review, a module having a building floor area of 100 m^2 was employed to characterize the typical apartment block. For the type of plan, a square building floor plan which was designed using four modules featuring a form factor (length/depth of the building in the plan) of 1.00, as seen in Figure 3.3.

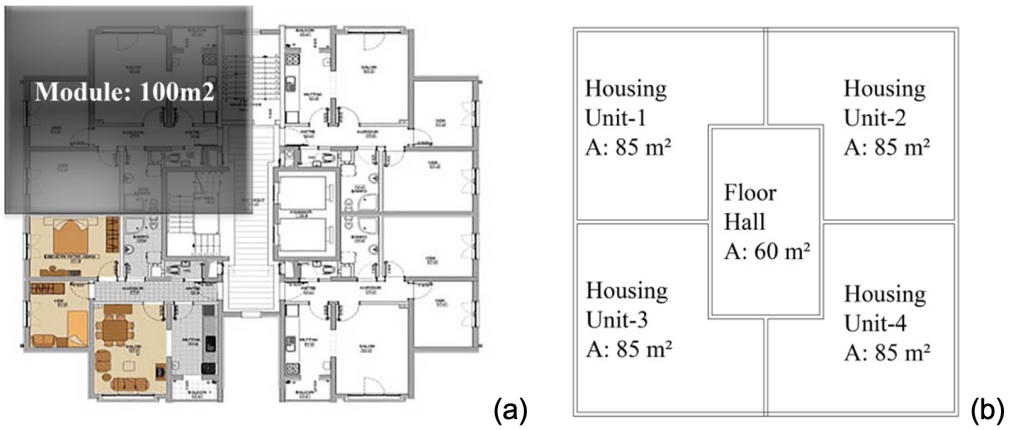

Figure 3.3 The square floor plan used by TOKI(a) model building floor plan(b)

A 5-storey building having a floor-to-floor height of 3.00 m was employed. In Türkiye, the average household comprises 3.6 people, and households comprising of nuclear families with children outnumber (53%) of those without children [64, 65]. Thus, for each housing unit, a family consisting of parents and two children was assumed, and the occupancy schedules were defined according to official surveys concerning the Turkish family structure. Design parameters that concerned the climate, occupants, building and active building subsystems were given in Tables 3.1 and 3.2 . The layer details for the opaque and transparent elements of the model building are shown in Table 3.2 and the limit values of overall heat transfer coefficients (U values) (W/m^2K) for Istanbul, as specified in the standard TS-825, 'Thermal Insulation Requirements for Buildings' [66], were used to create them.

Table 3.1 Characteristics of the climate region and model building

Parameter	**Value**
Climate region / Representative city	Temperate-humid / Istanbul
Heating / Cooling degree days	1886 / 2152
Global horizontal radiation	1465 kWh/m^{2}y
No. of storey	5
Floor to floor height	3.00 m
Building height	15 m
Floor area	400 m^2 (square floor plan)
Building length / building depth in the plan	1.00 (square floor plan)
Total exterior surface area / building volume	0.33
Total transparent area / total facade area	30% (north, south, east, west)
Constant air change rate	0.5 ac/h
Occupant density	0.04 m^2/person
Occupant clothing type	1 clo (heating period), 0.5 clo (cooling period)
Heating set point	20°C (07:00-23:00), 13°C (other hours)
Cooling set point	26°C (07:00-23:00), 32°C (other hours)
Minimum fresh air	10 l/s

Table 3.2 Characteristics of the residential model building elements

Element	**Construction Material Layers (from outside to inside)**	**U Value** (W/m^{2}K)
Exterior Wall	0.006m cement plaster+0.04m rockwool+0.03m cement rendering+0.19m brick+0.02m gypsum plaster	U_w= 0.57
Ground Floor	1m reinforced concrete + 0.03m concrete + 0.04m extruded polystyrene + 0.03m concrete + 0.05m screed + 0.01m timber flooring	U_f= 0.53
Hipped Roof	gravel + roofing felt + 0.08 m extruded polystyrene + EPDM + 0.04 m concrete + 0.14 m reinforced concrete + 0.02 m gypsum plaster	U_r= 0.36
Window	0.06 m PVC frame, 0.004 m Low-e coated glazing + 0.012 m air + 0.004 m clear glass	U_w=1.80

Simulation-based Data Generation

The building energy model was implemented via the DesignBuilder simulation program [67],which serves as the graphical interface for the EnergyPlus dynamic thermal simulation engine. The DesignBuilder program is recognized as among the most comprehensive methods for calculating energy input and output in detail using dynamic parameters. The geometry of both settlements and buildings was defined,

assuming each building in a settlement shares identical design parameters with the model building (Figure 3.2). The housing units and the floor halls of the model building were agreed as an independent zone in the zoning criteria.(Figure 3.3). In addition, the following were set for the building energy model: the thermo-physical characteristics of the building envelope (Table 3.2), the profiles of building use within individual thermal zone regarding hourly timelines of occupancy, activities, ventilation needs, lighting power density, electrical equipment power density, and domestic hot water demand; and the operation of HVAC(Heating, Ventilation, and Air Conditioning) systems via the assignment of hourly schedules of set-point temperatures according to the type of HVAC system (Table 3.1).

Figure 3.4 illustrated the workflow used to generate the simulations. According to the fixed settlement form, three IDF files were created using DesignBuilder. The first file was used for the building designed without solar control elements; the second was designed with fixed aluminum overhangs for the south and overhangs and fins for the east and west facades; and the third was designed with a Venetian blind system for the south, east and west facades. Then, by the written coupling function for communication between EnergyPlus and MATLAB, the IDF files were manipulated according to the design variables relevant to the envelope in the MATLAB environment, after which EnergyPlus was launched to perform a dynamic energy simulation by employing a proper weather data file. IDFs were prepared by putting a search string (e.g., parameter_name) in the place of a desired parameter, and each EnergyPlus simulation then selected parameter inputs based on a set of predefined values for evaluation. Each of the three IDFs were simulated with 2,048 different combinations of parameters, resulting in 6,144 total simulations. At the end of the automated simulation, the resulting 6,144 CSV files containing the simulation outputs were processed under the MATLAB environment to calculate the objective functions, with a total of 147,456 iterations considering the different energy and renewable energy systems being recorded for each iteration. The total simulation time of each IDF file consisting of 2,048 alternatives was around five days, while the total time for calculating the objective functions based on the 6,144 CSV files under the MATLAB environment was around 40 hours using an Intel Core i7 9750H processor with a speed of 2.60 GHz [68].

While creating the data set, design parameters (also known as “features”), the simulation inputs, and KPI, the simulation outputs, were defined. Some parameters were kept constant including the plan type (square floor plan), the number of storeys (5 storeys), the roof type (hipped roof with a roof slope of 33%) [62]. In the synthetic data set, input parameters were classified into two groups: fixed parameters (settlement level design parameters and building scale: plan type,

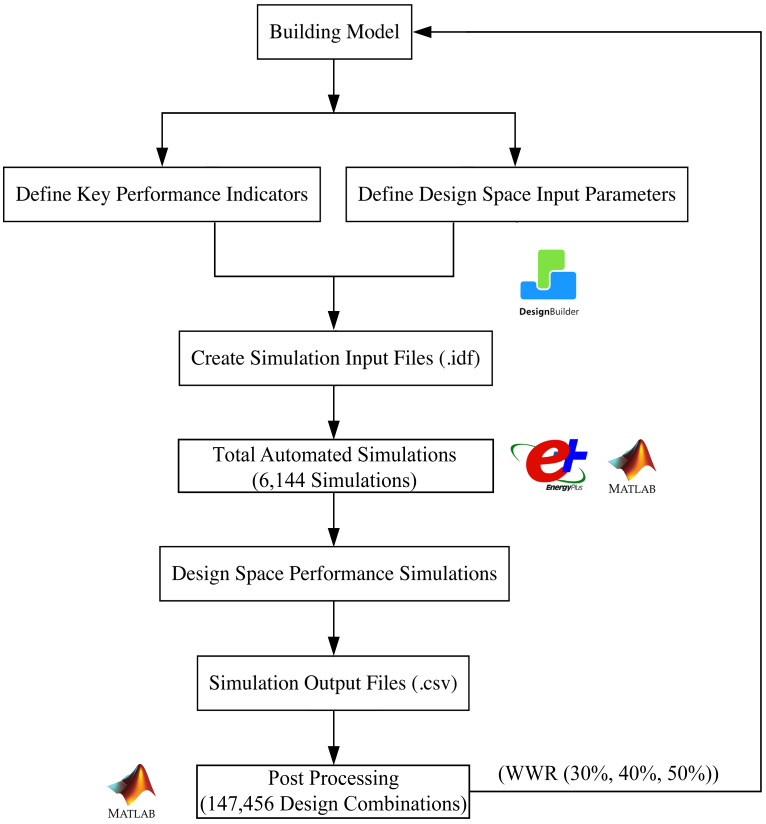

Figure 3.4 Workflow of simulation-based data generation

building height (number of floors) and roof type) and variable parameters (building level design parameters: transparency ratio (total window area/total facade area), building envelope, energy systems, renewable energy systems). It was assumed that WWRs are the same for each facades [62].

In this study, 13 design parameters were considered as input. As presented in Table 3.3, there was 1 design parameter for building geometry, 8 design parameters for the building envelope, 3 for energy systems, and 1 for renewable energy system. The input parameters in building performance studies generally consisted of discrete data [44] (a finite set of values defines each parameter) and depending on parameter ranges design alternatives (a set of combinations) were formed [69].

While defining the building envelope range variables, the maximum U value was defined according to the limit value of the Turkish standard (including three versions), and the minimum U value was defined according to the Passive House standard. The solar heat gain coefficient (SHGC), overall heat transfer coefficient (U value, (W/m^2K)) of the glazing system, the U values of the frame, and the passive (Fixed Solar Control Element, overhangs, and fins) and active (External Venetian Blinds) solar control elements were defined in Table 3.4 [68].

The building's energy system related variables describe the following issues: i:Heating systems: Provide only heating, and heating energy load is fulfilled by a central hot water boiler that runs on natural gas. This system is further categorised according to the Seasonal Coefficient of Performance (SCOP) of the boiler (Table 3.3); ii.Cooling systems: Provide only cooling (in this approach, only electrical systems are assumed to be available). This system is further categorised according to the Seasonal Energy Efficiency Ratio (SEER) of the split air conditioner. The hot-water system was assumed as consisting of standalone electric water heaters each having 80% thermal performance. Heating–cooling systems: Provide both space heating and cooling (only electrical systems are assumed to be available). Concerning the renewable energy system, both the high potential of solar energy in Türkiye and a photovoltaic (PV) system capable of contributing to a more distributed and efficient system in which buildings represent but one element within the energy supply infrastructure were considered. A grid-connected, roof-mounted PV system was used to assess the energy potential of such systems as related to the energy improvement of buildings in terms of a reduction in electrical energy consumption. To properly design a PV system, it is necessary to define the available application area for the roof according to architectural and solar suitability. A PV system was installed on the condition that it faced south in accordance with a roof slope of 33°, the common roof slope application in Türkiye [68].

The primary energy consumption (PEC) value (kWh/m^2-year) was calculated as a simulation output, represented the building energy performance. PEC values for design space sample combinations were calculated with the following Equation 3.1;

$$PEC = \sum (E_{cons_fuel} \times f_{p,fuel}) - \sum (E_{PV} \times f_{p,PV}) \tag{3.1}$$

E_{cons_fuel}, represents the yearly energy consumption according to the fuel type, (kWh/m^2-year); E_{PV} is the yearly energy produced from the photovoltaic (PV) system (kWh/m^2-year); $f_{p,fuel}$ is the primary energy conversion coefficient according to fuel type; $f_{p,PV}$ is the primary energy conversion coefficient for the electricity produced from the PV system. Depending on the fuel type, the primary energy conversion coefficients used in the equation are 1.00 for natural gas and 2.36 for electricity in Türkiye. PV system's primary energy conversion coefficient for electricity production was considered equal to the primary energy conversion coefficient for electricity specified in Türkiye. The yearly decline rate of the PV modules' power output was assumed to be 0.5% per year [68].

Table 3.3 Characteristics of the design space input parameters

Input Parameters	**Category**	**Sub-category**	**Parameter Range**
Geometry	Settlement Level Design Parameters	Settlement Form	Square
		Aspect Ratio	1.00
		Orientation	90°
	Building Level Design Parameters	Plan Type	Square
		Building Height	15m
		Roof Type	Hipped
		p1 Window Wall Ratio (WWR)	30%/ 40%/ 50%
Building Envelope	Exterior wall	*p2* Core Material	Horizontal Coring Brick / Aerated Concrete Block
		p3 Insulation type	Extruded Polystyrene Foam (XPS) / Rock Wool (RW)
		p4 Insulation thickness (m)	0.04 / 0.10 / 0.16 / 0.22
	Roof	*p5* Insulation type	Extruded Polystyrene Foam (XPS) / Rock Wool (RW)
		p6 Insulation thickness (m)	0.10 / 0.16 / 0.22 / 0.28
	Ground floor slab	*p7* Insulation thickness (m)	0.04 / 0.10 / 0.16 / 0.22
	Glazing	*p8* Glazing type	GL1 / GL2 GL3 / GL4
	Solar Control Element	*p9* Type of Solar Control Element	Fixed Solar Control Element (overhangs (South), overhangs and fins (East, West)) / External Venetian Blinds (South, East, West) / Not available
Energy Systems	Heating system	*p10* Efficiency	Radiator heating, boiler η: 0.86 / 0.95 (fuel type: natural gas) / Air-Source Heat Pump (ASHP) SCOP: 3.29 (Heating); 2.25 (Cooling) / Variable Refrigerant Flow (VRF) SCOP: 7.20 (Heating); 4.20 (Cooling), (fuel type: electricity)
	Cooling system	*p11* Efficiency	Split Air Conditioning System (fuel type: electricity) SEER: 5.80 / 8.50 / Air-Source Heat Pump (ASHP) SCOP: 3.29 (Heating); 2.25 (Cooling) / Variable Refrigerant Flow (VRF) SCOP: 7.20 (Heating); 4.20 (Cooling), (fuel type: electricity)
	Hot water system	*p12* Efficiency	Stand Alone Water Heater (η) 0.86 / Hot Water Boiler (η) 2.41 (fuel type: electricity)
Renewable Energy System	Photovoltaic system	*p13* PV-rooftop	Mono crystalline module (250 Wp); η_{module}:15% ; $\eta_{inverter}$:96%) / Not available

Table 3.4 Features of the specified glazings

No	Description	SHGC	U_{gl} (W/m^2K)	U_{window} (W/m^2K)
GL1	Low-e (heat cont.) (4 + 12 air + 4 mm)	0.436	1.628	1.80 (U_{window_limit}: 1.80 [66])
GL2	Low-e (heat cont.) (4 + 12 argon + 4 mm)	0.430	1.260	1.50
GL3	Low-e (heat + solar cont.) (4 + 12 argon + 4 mm)	0.296	1.204	1.40
GL4	Low-e (heat + solar cont.) (4+16 argon +4+16 argon+ 4 mm)	0.253	0.550	0.80

3.1.2 Data Preprocessing

As stated in the literature review, data preprocessing increased the models' prediction accuracy. Firstly, the data had to be checked to ensure that the input features were suitable for the model, so incorrect or noisy data were removed. Then, input data were translated into a form that algorithms could process. Input data mostly consisted of discrete values and were hot-coded. Hot encoding is a common method used for categorical inputs in ML models, as mentioned in Chapter 2. Model output value and some of the input values, which are numeric (p1, p4, p6, p7), were treated as continuous, with no preprocessing applied. For ANN and SVM models, the numerical input features for each training set were normalized so that the features were centered with a mean of zero and a variance of 1.0. Rescaling the input causes the features to be on a similar scale, thereby reducing the risk of the model's inaccurately assigning importance to the input features. This also allows for more balanced training. The same procedure was applied to test datasets. RF algorithm did not require this process.

All datasets histograms were shown in Figures 3.5, 3.6, 3.7, and 3.8 . The correlation heatmap in Figure 3.9 illustrated the relationships among various features, particularly those related to building physics. Also, the distribution of the design parameters for all datasets were shown in Figures 3.5, 3.6, 3.7, and 3.8 .

3.1.3 Design Space Sampling Strategy

This study's input parameters were discrete, and design alternatives were created with the full-factorial method. A full-factorial design space exploration approach, which covered all configurations in the design space, offers flexibility and estimates the energy performance for each configuration by executing an automatic sequential assessment through the design space [70]. As a result, 147,456 building design samples were created for each WWR parameter. However, since the number

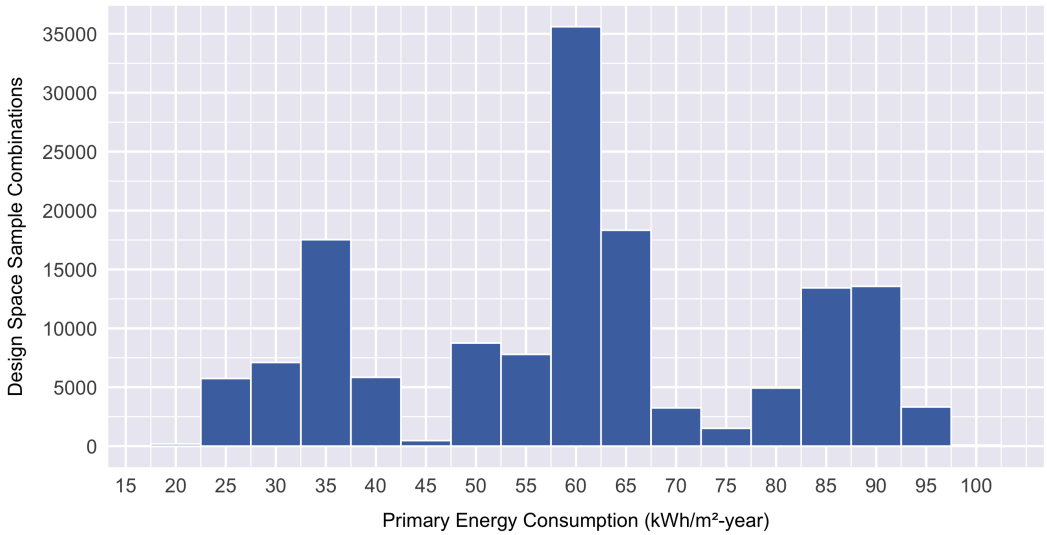

Figure 3.5 Histogram of primary energy consumption for DS1

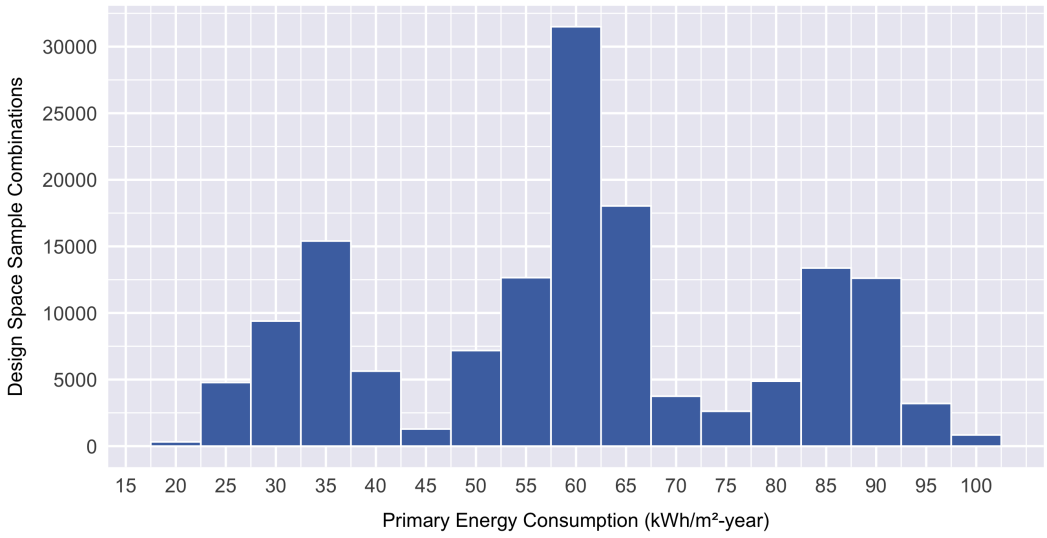

Figure 3.6 Histogram of primary energy consumption for DS2

of samples was relatively high (the literature usually considers the number of samples between 1000 and 5000 adequate), it was not possible to calculate it due to computational cost of resources. Instead, the data were downsampled to reduce a subspace that still represents the entire design space. All datasets were reduced to 36,864 samples, meaning models were created with 25% of the entire dataset.

3.1.4 Train/Test Splits

The characteristic that made the interpolation and extrapolation approach fundamentally distinct was the data set separation in the training and test sets. Interpolation tried to predict output values that had the same data set. On the

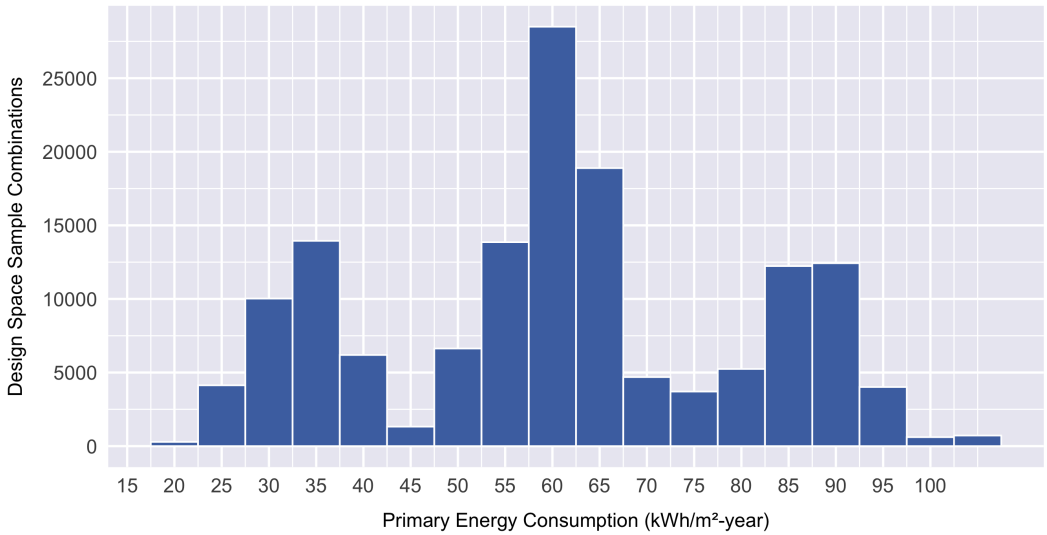

Figure 3.7 Histogram of primary energy consumption for DS3

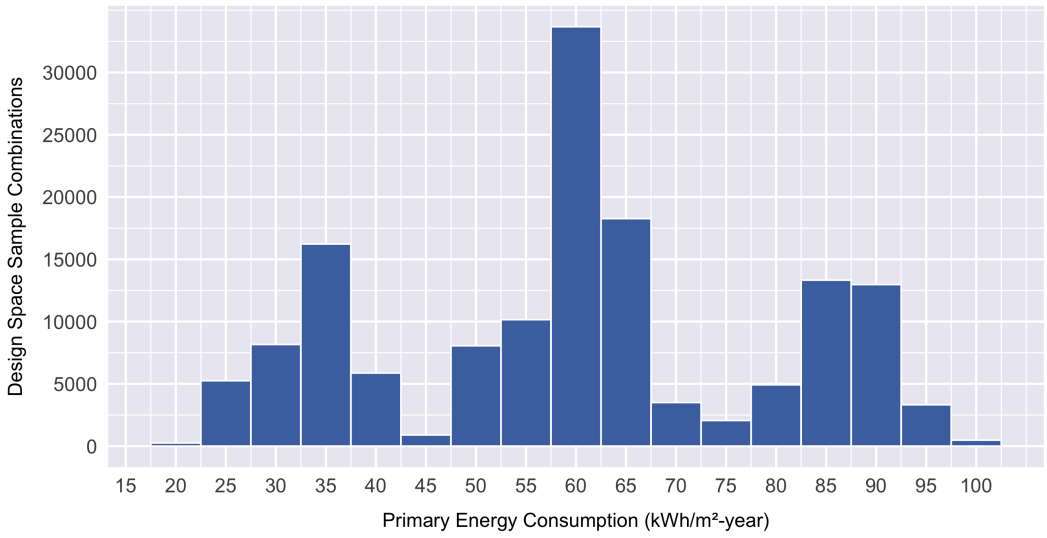

Figure 3.8 Histogram of primary energy consumption for DS4

other hand, extrapolation tried to predict the values of unknown data by establishing various relationships using different data sets. Since models tended to overfit during training (they tend to have better performance metrics while they perform poorly on other datasets), when to cease training is critical to their generalization performance. A validation set (val. set) is used to objectively monitor model performance during training and determine where to stop it. Also, a random seed number was applied to ensure the results were reproducible, using the 'set.seed()' function, 0 value used. The training set, a subset of the data set, was selected with random sampling, using the sample function.

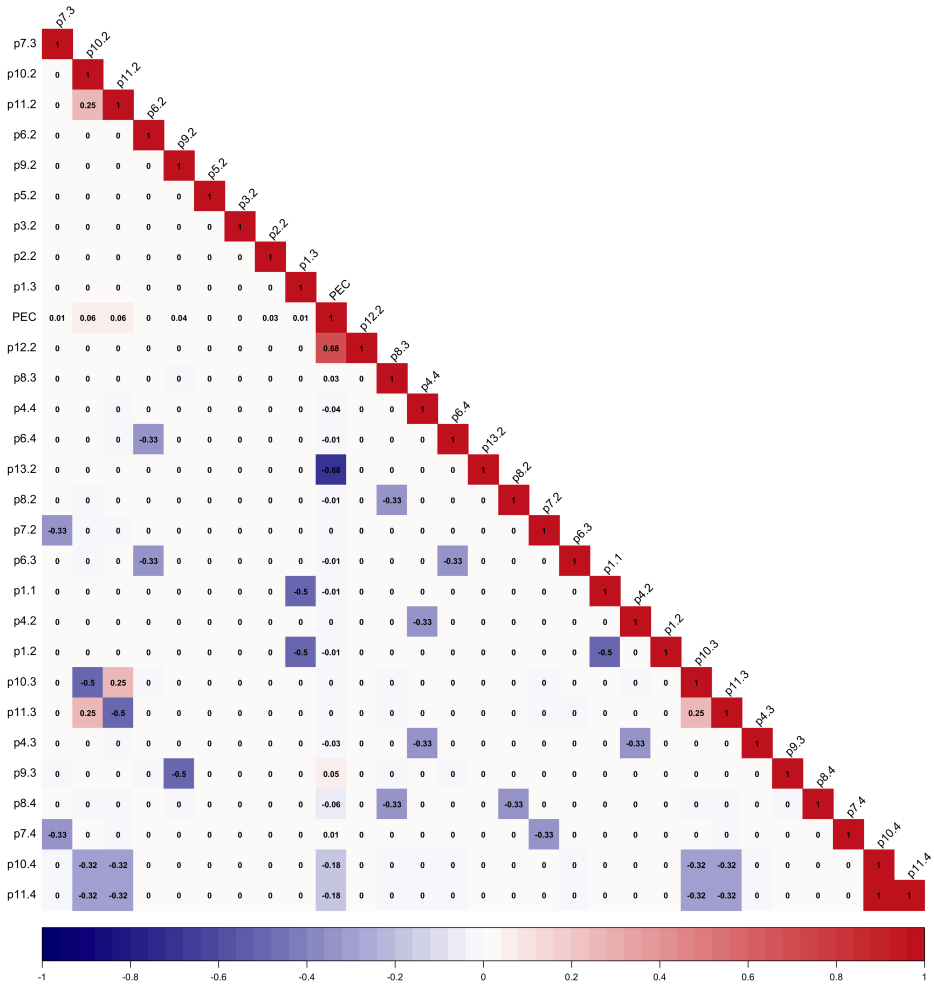

Figure 3.9 Correlation map of building design parameters

Going beyond the values encountered in the literature, this study also differentiated the ratios of training and test sets, intending to observe the direct effect of the training set on model performance. The test set sample size was kept all the same and equal in order to evaluate each training set fairly, as 7,372. To evaluate the performance of the ML algorithms with different sample sizes, datasets with 11,058, 14,744, 22,117, and 36,863 instances were sampled with random split. The training sets have been determined to be 10%, 20%, 40%, and 80% of the entire dataset and are shown in the Table 3.5.

Table 3.5 Dataset splits

Data Set Code	Data Set Size	Train Set Ratio	Train Set	Test Set
DT0.1	11,058	10%	3,686	7,372
DT0.2	14,744	20%	7,372	7,372
DT0.4	22,117	40%	14,745	7,372
DT0.8	36,863	80%	29,491	7,372



Figure 3.10 Design parameter value distributions of DS1



Figure 3.11 Design parameter value distributions of DS2



Figure 3.12 Design parameter value distributions of DS3



Figure 3.13 Design parameter value distributions of DS4

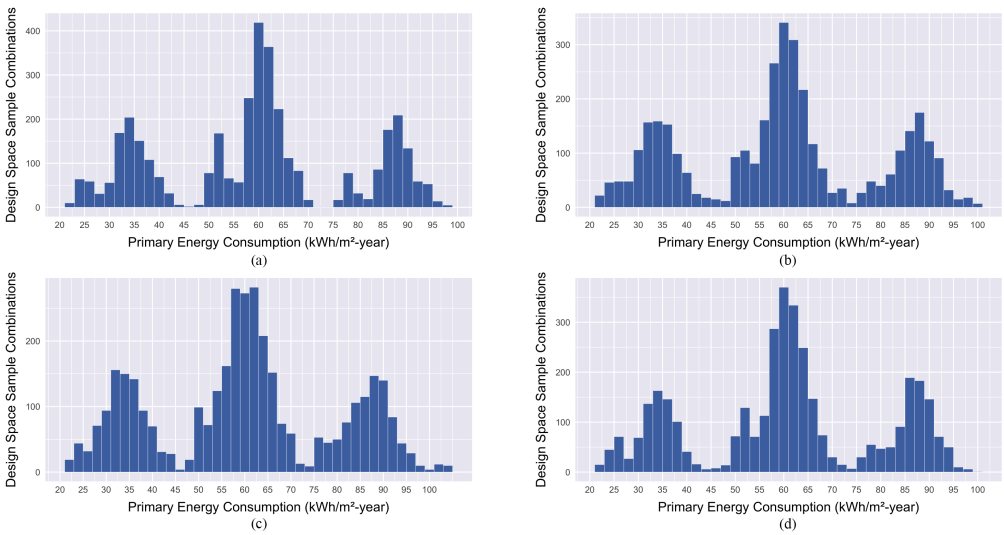

Figure 3.14 Histogram of primary energy consumption for DT.01 downsampled train data sets a)DS1 b)DS2 c)DS3 d)DS4

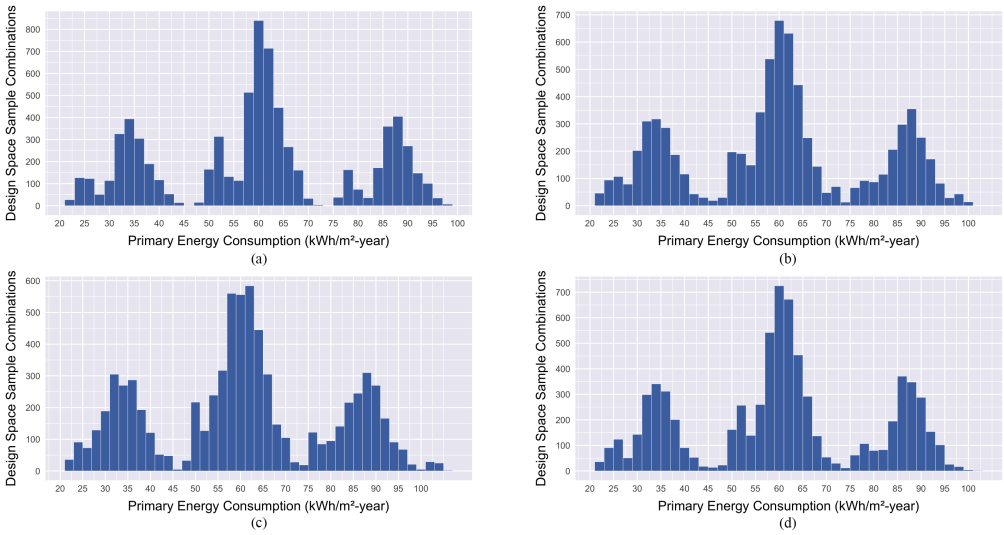

Figure 3.15 Histogram of primary energy consumption for DT.02 downsampled train data sets a)DS1 b)DS2 c)DS3 d)DS4

A histogram of the downsampled DS1, DS2, DS3, and DS4 simulation datasets and test sets were shown in Figures 3.14, 3.15, 3.16, 3.17, and 3.18 .

3.1.5 Machine Learning Models Development

In particular, the primary criterion was to obtain the surrogate model type with the highest possible accuracy. In some cases, it was preferred to balance optimal accuracy and proper interpretation of the model. While each type of surrogate model had benefits and drawbacks concerning specific modeling requirements, several authors recommended using multiple models to identify the most suitable one [11].

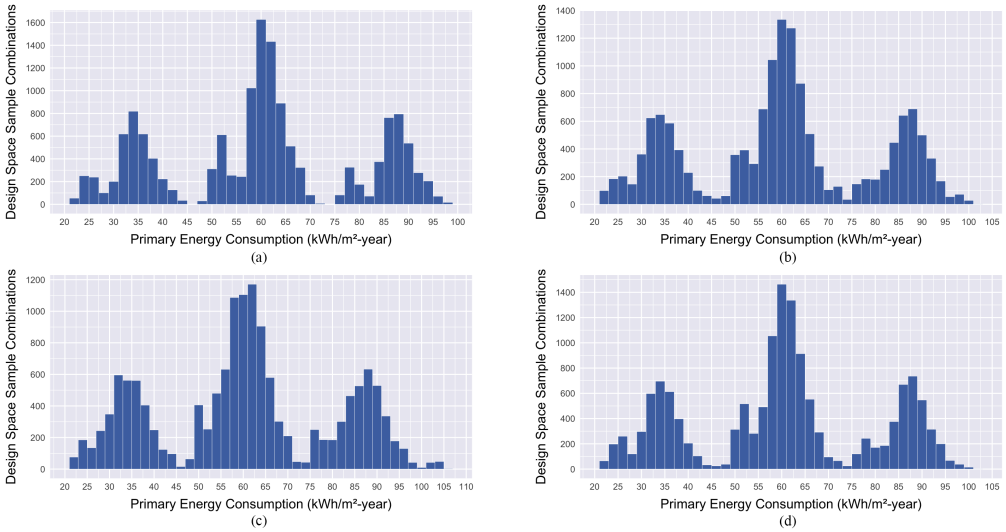

Figure 3.16 Histogram of primary energy consumption for DT.04 downsampled train data sets a)DS1 b)DS2 c)DS3 d)DS4

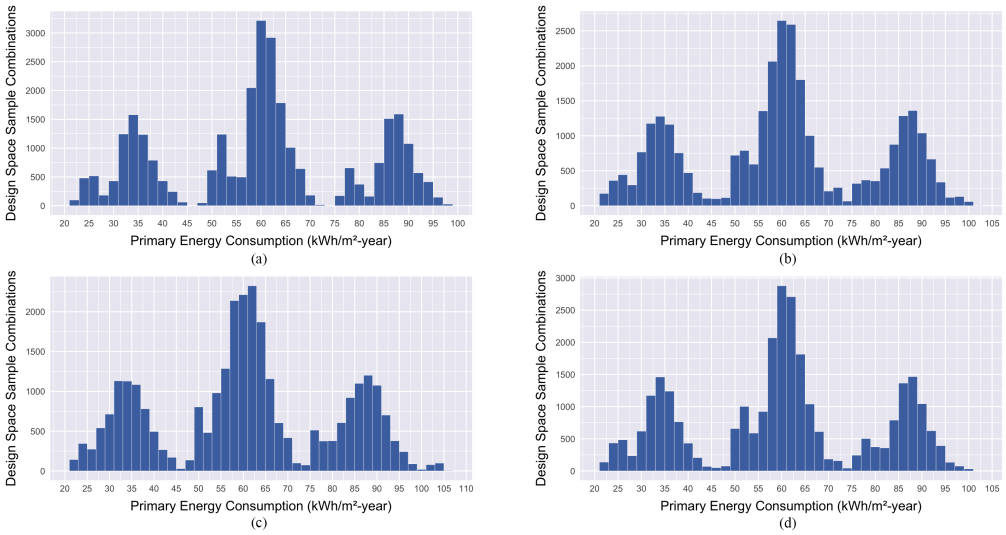

Figure 3.17 Histogram of primary energy consumption for DT.08 downsampled train data sets a)DS1 b)DS2 c)DS3 d)DS4

ANN type of FFNN used in this study, with a single layer, was chosen. The architecture followed a straightforward feedforward path from input to output. Numerous studies have demonstrated effectiveness of this algorithm for predicting building performance [31]. This algorithm was specified by the method = "nnet" in the train function from the caret package. The linout = TRUE function indicated that the output layer should use a linear activation function, which was necessary for regression tasks. Lastly, maxit was set to 200.

RF model created with the randomForest function and inherently provides an out-of-bag error estimate, which is an internal cross-validation method [71].

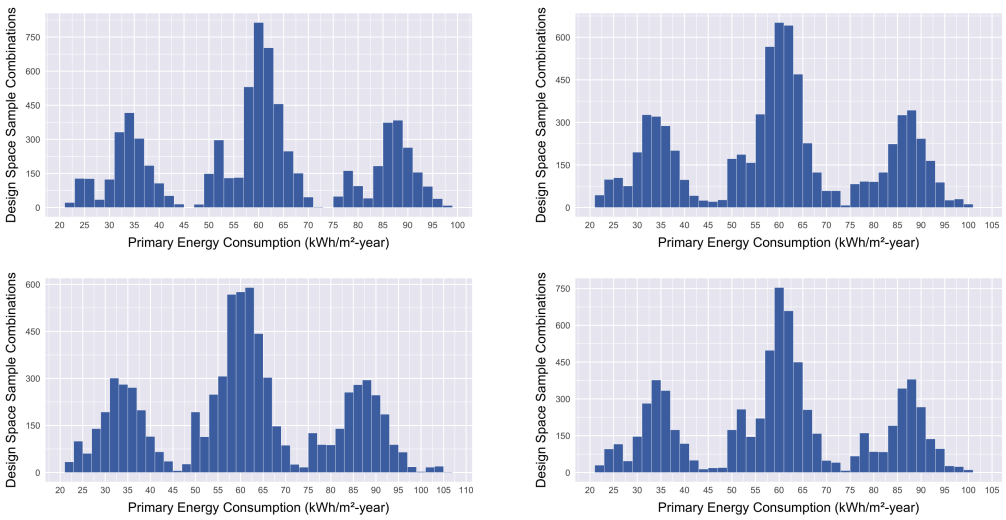

Figure 3.18 Histogram of primary energy consumption for downsampled test data sets a)DS1 b)DS2 c)DS3 d)DS4

SVM model train function from caret package. SVM with an RBF kernel is specified with method = svmRadial function. The data was preprocessed by centering (subtracting the mean) and scaling (dividing by the standard deviation) with preProc = c("center","scale") before training. Also, with the expand.grid function the C=c(1., 10., 100.) and sigma= c(0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1., 2., 5.)) values combinations created.

3.1.6 Hyper-Parameter Tuning

To achieve the best prediction performance, ML models employed different hyperparameters. In this thesis, each set of hyper-parameters was utilized using a 5-fold CV. This method was commonly used and prevented overfitting because an overfitted model performs well on training data but does not generalize well when it comes to unseen data [11].

For ANN, size function that represented the number of neurons (units) within the hidden layer, was set to 9, 18, 27, and 36. The decay function, weight decay (regularization) parameter, was set to values of 0, 0.1, 0.01, 0.001.

For SVM, with the expand.grid function the C=c(1., 10., 100.) and sigma= c(0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1., 2., 5.)) values combinations were created.

RF model may not have been the most precise without searching the optimal mtry because the total number of variables is fairly small (13 input variables in total). It was possible to test all numbers and select the optimal mtry by comparing their corresponding generalization performance. The expression (mtry = c(2:5)) created

a grid of mtry values to be tested. In this case, values from 2 to 5 were included in the grid. These values were processed to experiment with different mtry values, aiming to discover an optimal set of hyperparameters. The model selected the mtry number with the smallest RMSE as the final value. nodesize = c(1, 3, 9) and ntree = c(100, 200, 500) functions values were also included in the model. The RF model was carried out with the train function from the caret package, and the best model was tried to be found with hyper-parameters. The detailed results of tuning models were presented in Chapter B.

3.1.7 Model Performance Analysis

Lastly, surrogate models were evaluated using the remaining samples, referred to as the test dataset. The test dataset was created for the evaluation of how the model performed on unseen samples. In this study, Root mean square error (3.2) , R-squared (3.3), Mean absolute error (3.4), and normalized Mean absolute error (3.5) were chosen, which were determined using the equations provided,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{\text{predict},i} - y_{\text{true},i}\right)^2} \tag{3.2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_{\text{predict},i} - y_{\text{true},i}\right)^2}{\sum_{i=1}^{n}\left(y_{\text{true},i} - \bar{y}_{\text{true}}\right)^2} \tag{3.3}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_{\text{predict},i} - y_{\text{true},i}\right| \tag{3.4}$$

$$nMAE = \frac{MAE}{\frac{1}{N}\sum_{i=1}^{N}\left|y_{\text{true},i}\right|} \tag{3.5}$$

where, $y_{predicted,i}$, y_{true}, and $\bar{y}_{true}$ represented the predicted data, simulated data, mean of primary energy consumption from simulated data respectively. N was the length of the dataset [33].

Østergård [7] arbitrarily explained in one of his work, consider a metamodel with R^2>0.90 as reasonable, R^2>0.95 as accurate, and R^2>0.99 as near perfect, respectively.

The surrogate models' prediction accuracy, time efficiency, and generalization capacity were evaluated. The comparison was made between the results of the surrogate models and the simulation results, including the prediction time.

3.2 Results and Discussion

Three different ML based surrogate models, ANN, RF, and SVM were used for design space exploration. These surrogate models were created based on the methodology discussed in the previous. The tables were presented separately to examine the results of interpolation and extrapolation approaches and included Coefficients of Determination(R^2), RMSE and MAE values, train and test time, and the total time. Surrogate models were developed using Rstudio programming language, with the caret library, and evaluations were executed on a PC, MacBook Air with Apple M2 8GB Memory.

3.2.1 Interpolation Results

The comparison between the building energy performance prediction values and the simulation results for three ML models — ANN, RF, and SVM were given in the Tables 3.6, 3.7, 3.8, and 3.9. This comparison was performed for the DS1 with varying training sizes (10%, 20%, 40%, and 80%), both train and test datasets were belong DS1, and represented the interpolation approach.

Table 3.6 Performance comparison based on DS1 test data for %10 train size

DT0.1	**ANN**	**RF**	**SVM**
R^2	0.9999	0.9978	0.9976
RMSE	0.190	1.034	1.073
MAE	0.152	0.788	0.911
nMAE	%0.253	%1.311	%1.516
Train Time (μs)	43307	4096	8510
Test time (μs)	4	68	5
Total time (μs)	43311	4164	8515

Table 3.7 Performance comparison based on DS1 test data for %20 train size

DT0.2	**ANN**	**RF**	**SVM**
R^2	0.9991	0.9978	0.9979
RMSE	0.556	0.777	0.923
MAE	0.425	0.598	0.746
nMAE	%0.707	%0.995	%1.242
Train Time (μs)	38733	6207	28103
Test time (μs)	3	78	14
Total time (μs)	38736	6285	28117

ANN consistently performed well across all training sizes and demonstrated a good fit with R^2 values nearing 1.0. Notably, ANN achieved low RMSE and MAE across all datasets. Furthermore, the model maintained efficient computation times in training and testing, suggesting its suitability for practical applications where quick predictions were essential.

Table 3.8 Performance comparison based on DS1 test data for %40 train size

DT0.4	**ANN**	**RF**	**SVM**
R^2	0.9999	0.9991	0.9981
RMSE	0.179	0.602	0.871
MAE	0.140	0.456	0.705
nMAE	%0.233	%0.759	%1.173
Train Time (μs)	45216	11079	64081
Test time (μs)	4	92	6
Total time (μs)	45220	11171	64087

Table 3.9 Performance comparison based on DS1 test data for %80 train size

DT0.8	**ANN**	**RF**	**SVM**
R^2	0.9997	0.9995	0.9981
RMSE	0.331	0.474	0.856
MAE	0.259	0.362	0.703
nMAE	%0.431	%0.602	%1.169
Train Time (μs)	45530	20525	180683
Test time (μs)	3	119	7
Total time (μs)	45533	20644	180690

RF also performed well, with high R^2 values indicative of a good model fit. However, the RMSE and MAE values were consistently higher than the ANN models. Additionally, RF's training and testing times increased substantially with larger datasets, reflecting higher computational costs. However, RF could be parallelized to speed up training time, especially when the number of trees (ntree) were large. This is not explicitly handled in this study but could be investigated. Even though the computational cost of training increased faster than that of neural networks, RF remained a viable option due to its balance between performance and interpretability. SVM showed competitive R^2 values but with high RMSE and MAE metrics, indicating that outlier numbers is relatively high. Moreover, SVM exhibited long total computation times, particularly with larger datasets. This increased computational burden might have limited its practicality in real-world scenarios despite its robustness in some contexts. It is worth mentioning that performance of the methods was strongly dependent on the selection of hyperparameters, and better performance metrics could have been obtained with more effort. Normalized error histograms illustrated in Figure 3.19 implied no systematic errors for ANN and RF models since the distributions were normal-like. However, we can see from the graphics, the distribution was asymmetric for SVM, which hinted that different mechanisms could contributed to the errors. The gaps in the DT0.1, DT0.4 for the ANN model, and DT0.8 for both ANN and RF models should be investigated.

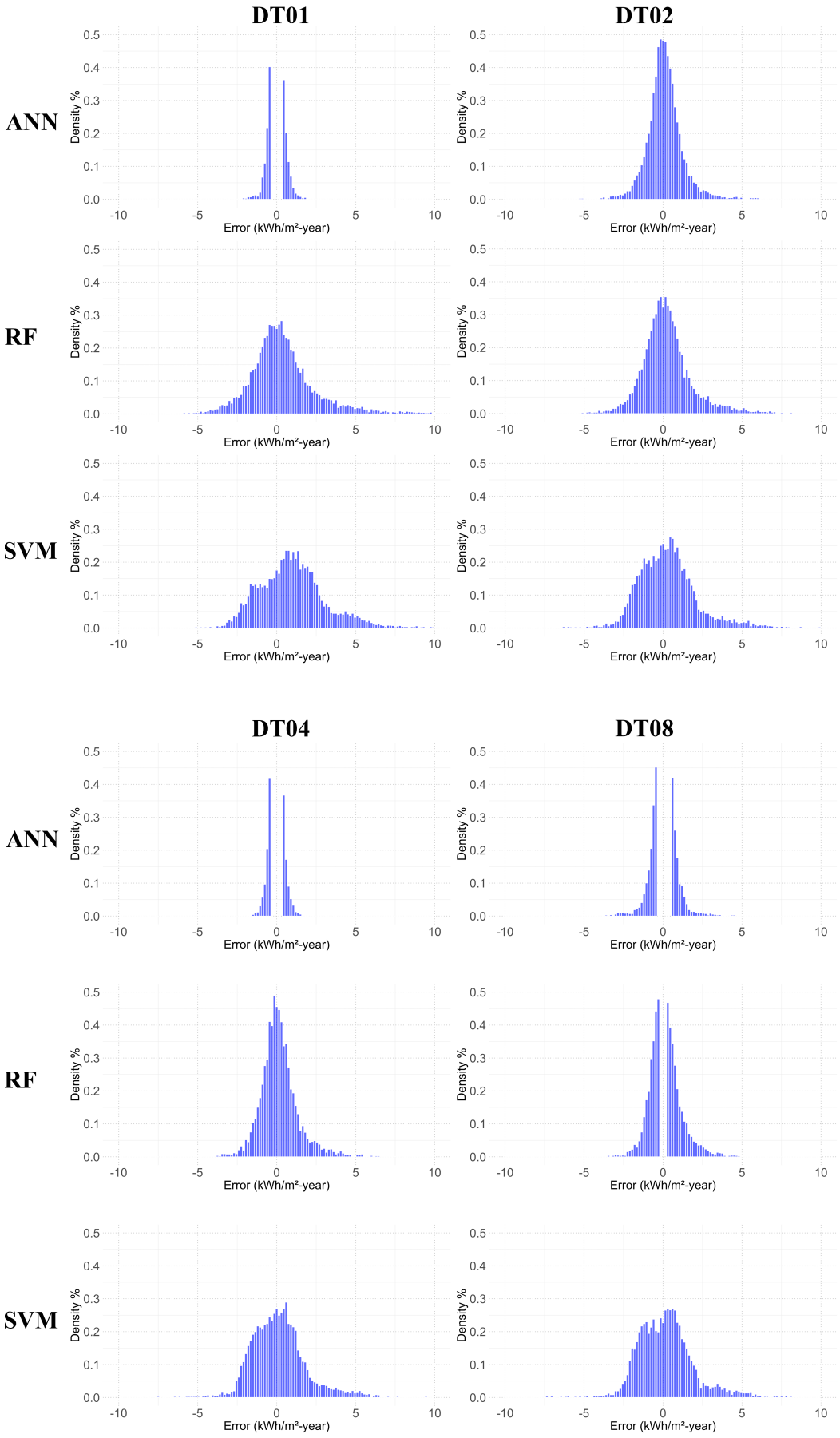

Figure 3.19 Error distribution of interpolation for all the train sizes

3.2.2 Extrapolation Results

In order to measure the extrapolation performance of the methods, we trained them on three different datasets, and results were given in the Tables 3.10, 3.11, 3.12, and 3.13. While two datasets involved samples from DS1 or DS2, the latter was a mixture of these two. Since there was no chance for the models to learn the relationship between PEC and WWR by providing only one WWR value, we used models trained on DS1 and DS2 as a baseline. Models trained with the mixture of two (DS4) should surpassed the baseline models if they learned the underlying relationship between WWR and PEC such that they could extrapolate well.

Table 3.10 Performance comparison based on DS3 test data for %10 train size

DT0.1	**ANN**		**RF**		**SVM**	
	30_40	**40**	**30_40**	**40**	**30_40**	**40**
R^2	0.9928	0.9965	0.9881	0.9926	0.9960	0.9949
RMSE	1.731	1.272	2.341	1.879	2.303	1.482
MAE	1.273	0.946	1.711	1.344	1.835	1.018
nMAE	%2.098	%1.559	%2.819	%2.214	%3.023	%1.677

Table 3.11 Performance comparison based on DS3 test data for %20 train size

DT0.2	**ANN**		**RF**		**SVM**	
	30_40	**40**	**30_40**	**40**	**30_40**	**40**
R^2	0.9873	0.9960	0.9910	0.9942	0.9969	0.9946
RMSE	2.351	1.352	2.039	1.664	3.305	1.573
MAE	1.588	0.999	1.476	1.198	2.595	1.128
nMAE	%2.616	%1.647	%2.432	%1.974	%4.276	%1.859

Table 3.12 Performance comparison based on DS3 test data for 40% train size

DT0.4	**ANN**		**RF**		**SVM**	
	30_40	**40**	**30_40**	**40**	**30_40**	**40**
R^2	0.9837	0.9962	0.9925	0.9949	0.9977	0.9947
RMSE	2.594	1.331	1.850	1.553	3.195	1.580
MAE	1.780	0.981	1.341	1.109	2.494	1.133
nMAE	%2.932	%1.616	%2.209	%1.827	%4.109	%1.862

Table 3.13 Performance comparison based on DS3 test data for 80% train size

DT0.8	**ANN**		**RF**		**SVM**	
	30_40	**40**	**30_40**	**40**	**30_40**	**40**
R^2	0.9310	0.9946	0.9994	0.9955	0.9978	0.9947
RMSE	5.206	1.540	1.653	1.461	3.101	1.563
MAE	2.461	1.172	1.202	1.047	2.418	1.130
nMAE	%4.055	%1.931	%1.981	%1.725	%3.984	%1.862

Investigating the performance of the models, we comprehended that for almost all dataset sizes, the baseline performed better or comparably. Besides, the performance of the models did not improve monotonically for all methods. In particular, the ANN method showed significant fluctuation with the dataset size, implying training instability. The dependence of optimal hyperparameters on dataset size can explain this behaviour.

SVM and RF models showed more stable performance, outperforming the baseline for some metrics, although they could not be said to extrapolate well. However, their performance was still decent compared to ANN, and they were more stable. For SVM, this could be related to having fewer parameters to be tuned since models with fewer parameters are known to generalize better. Although the RF model had a significant number of parameters to be tuned, it was composed of an ensemble of models, which can also improve the method's generalization ability.

Finally, the models showed unsatisfactory performance in extrapolation cases. However, experiments at a broader scale should be conducted in order to get more conclusive results. One of the main reasons for this may be that the sampling grid was very coarse on the WWR ratio axis, only containing two different values. The study can be improved by introducing more variety for WWR ratios. This research required a significant amount of extra simulations and, it was postponed due to computational resource constraints.

Like the interpolation case, normalized error histograms illustrated in Figure 3.20 implied no systematic errors for ANN and RF models since the distributions were normal-like. Similarly, for SVM, we noticed that distribution was asymmetric.

3.2.3 Effect of Increasing Size of Training Data

At their core, surrogate models investigated in this thesis aimed to decrease computational costs. The main questions were to what extent we could reduce the computational costs and how much performance was sacrificed in the process. The computational cost reduction was directly related to training set sizes. For evaluating issues mentioned above, this part investigated the dependency of the performance metrics on the training set size for all models.

We used datasets of different sizes sampled from the same distribution to train three different models. The training set size varied between 3686 and 29491 samples, while the test size remained constant at 7372. We used 5-fold cross-validation on the training set with hyperparameter tuning. Results showed that models performed very well regarding RMSE and other performance metrics for all training set sizes.

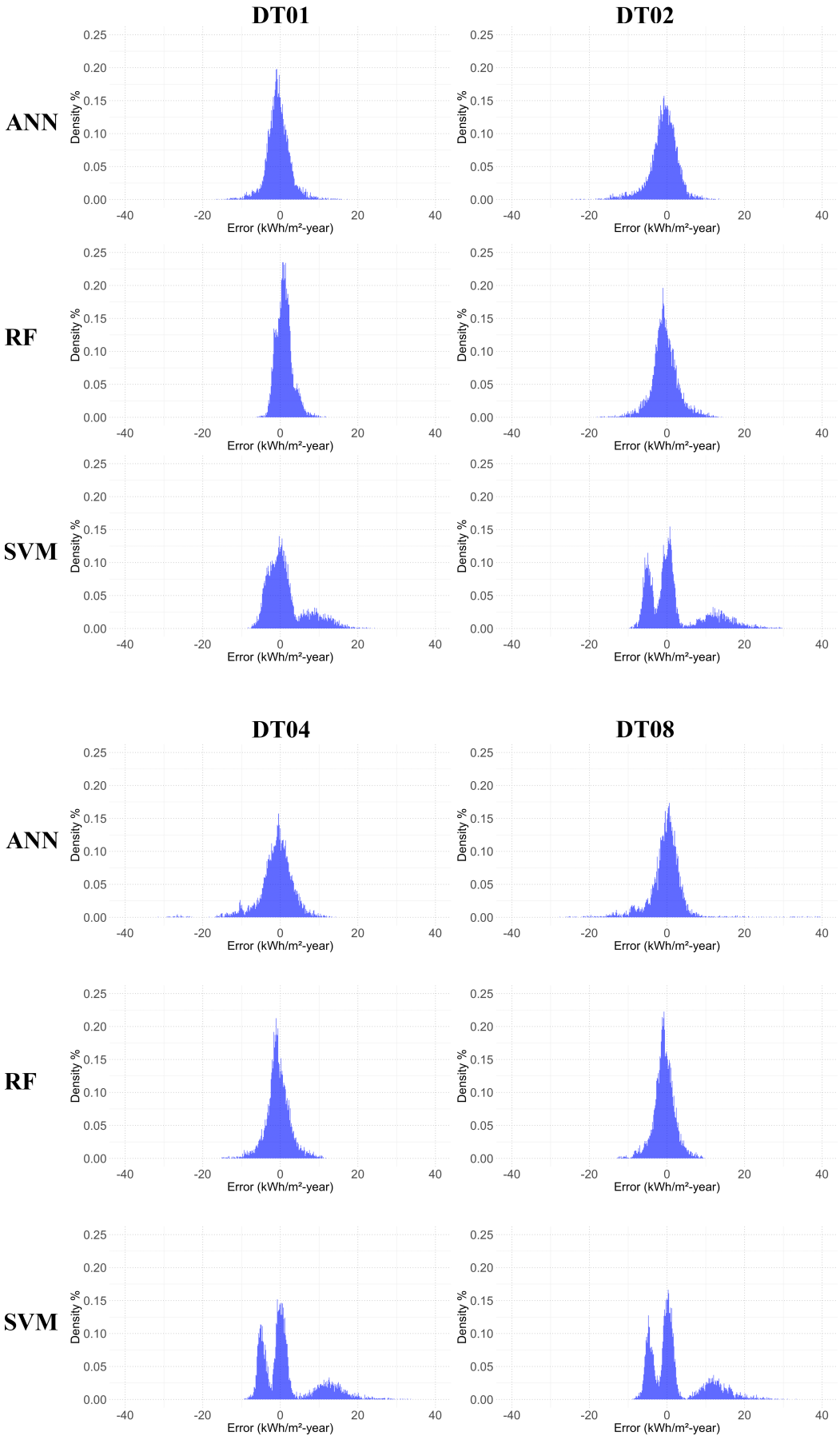

Figure 3.20 Error distribution of extrapolation for all train sizes

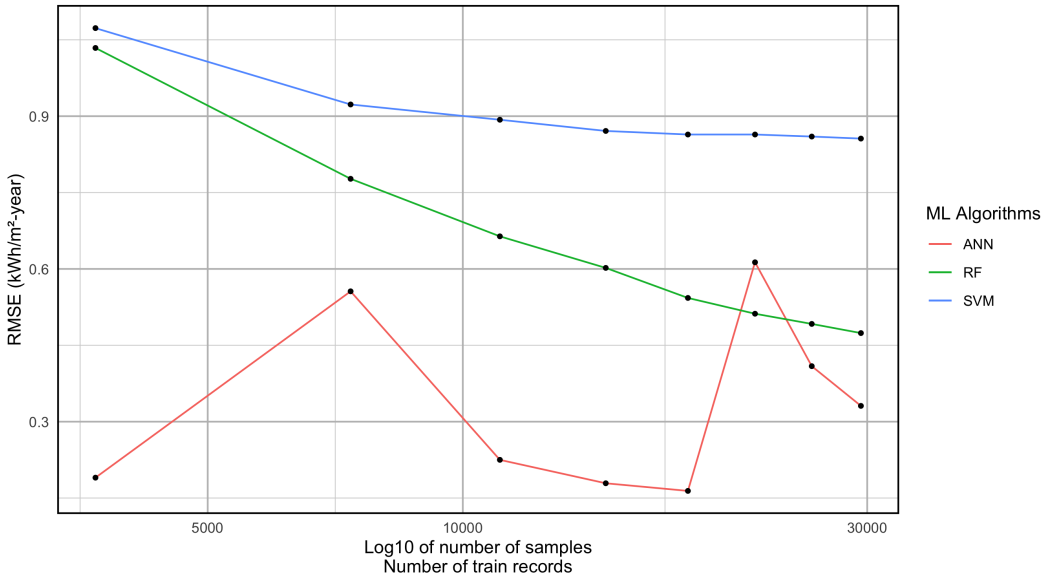

Figure 3.21 RMSE for building energy performance prediction against number of total number of samples used for training

Consequently, if our motivation was to use the models for design space exploration, this would have led to a reduction of computational cost by approximately a factor of 40 with negligible performance degradation (recalling that simulation results deviate from the real-world scenario by up to 13% [38], this is almost negligible). One issue worth mentioning was the ANN model's performance fluctuated due to unexpectedly changing training set size. The reason for this fluctuation could be examined. As expected, other models also performed decently, which improved with the training set size. Figure 3.21 illustrated the relationship between RMSE and the training set size for all three models.

4
CONCLUSION

Energy efficiency and energy savings in buildings are critical focus areas across various fields worldwide. Professionals in each sector strive to minimize energy consumption as much as possible. Achieving efficient and reduced energy use in buildings leads to lower CO_2 emissions and benefits consumers and producers economically. Building energy performance can be calculated through BPS tools with highly accurate predictive feedback. However, developing and creating BPS models comes with several challenges. As in this study, creating all possible combinations based on the relevant parameters with the full factorial sampling method raises the problem of high computational load. Although automated parallel parametric energy simulations were performed during the data set creation, the computational load is still outside of practical limits. Decision-makers require powerful computational resources for simulations that may not be available to all groups. Even if such resources are available, design space exploration can take up to several months, disrupting the architect's creative process. Reducing model fidelity to shorten calculation times is not viable, as it can lead to less reliable results. Due to the uncertainty and variability of building parameter values, millions of high-fidelity energy models must be simulated, creating a high-dimensional design space known as the "curse of dimensionality". The long computation times resulting from this dimensionality make it challenging to explore the design space effectively. Consequently, researchers emphasize the need for simple and fast energy prediction tools to evaluate and compare building design scenarios.

Therefore, this study aims to employ fast and accurate surrogate models based on supervised ML to develop improved strategies for reducing energy usage by conducting detailed data analyses and predicting building energy performance. In this research, a typical multi-family residential building is taken as a case study with consisting of different design alternatives. Data quality is crucial for developing an effective machine-learning model. The data preprocessing phase involves several key steps: handling missing values, removing duplicates, adjusting data types, detecting outliers, and integrating data. Once the raw dataset is transformed into a high-quality one, building energy performance can be analyzed. This study uses three common ML methods: ANN, SVM and RF, because numerous studies have demonstrated their effectiveness in predicting building performance. According to the results, models can quickly and accurately predict the performance indicator

(primary energy consumption (kWh/m^2-year)) for different design combinations using a data set based on detailed dynamic building energy simulation results and make it possible to explore multidimensional design spaces in a shorter time and with the same efficiency (R^2>0.99).

4.1 Addressing Research Questions

RQ1: Is it possible to use BPS for feasible design space exploration with the help of ML methods? How much performance gain can be obtained using surrogate models, and what is the compromise?

Our experiments showed that the significant computational cost of creating surrogate models required some samples for training. Computational costs per sample were negligible compared to the computational cost per simulation. Besides, prediction times were in the order of microseconds, millions of times faster than running simulations. Thus, we could say that computational gain was $\frac{number of samples}{number of training samples}$, which was arbitrarily high. We also observed that the performance metrics were quite satisfactory (nMAE < 1%) considering that simulations were accurate to around 13% compared to reality.

RQ2: How does the distribution of training and test datasets affect the performance of ML-based surrogate models, particularly under interpolation and extrapolation scenarios?

Surrogate models, when their hyperparameters were tuned accurately, performed very well in interpolation scenarios, especially when the training and test datasets were identically distributed. However, in extrapolation scenarios, our results showed that they could perform poorly. Nevertheless, this study should not be considered conclusive since the sampling grid resolution was low along the extrapolated axis.

RQ3: How does varying training sets' size impact the ML model's performance in estimating the energy performance of buildings?

Figure 3.21 showed the relationship between RMSE values for all models training sizes. We found that the performance of SVM and RF improved asymptotically with increasing training set size. Meanwhile, ANN did not show a stable trend in performance metrics. This was considered to be related to hyperparameter selection, especially the maximum iteration parameter, which had been planned to be fixed. Besides, smaller training set sizes can also be investigated to see if the asymptotic trend continues.

RQ4: Which ML techniques are more suitable for creating surrogate models?

When proper hyperparameter tuning was done, the ANN model performed well in interpolation. Additionally, the time complexity of the training was linear concerning the number of training samples, which became very important for more extensive training sets when more accuracy was favorable. However, hyperparameter selection could be tricky due to the complex structure of the model. On the other hand, tuning SVM and RF models was more straightforward, and their performance dependency on hyperparameter selection was observed to be lower. Consequently, they were suitable for the early stages of design space discovery due to their robustness and explainability, while ANN should have been used at later stages due to its performance.

4.2 Research Limitations

One of the main limitations of ML models are the extrapolation approach, in this study, we took a step to overcome this limitation and tried to predict unseen model outputs. Afterwards, the accuracy of the models is never certain; it can be assumed that models are accurate in order to existing data set limitations. The existing data set which is simulation results also has some limitations. Physics-based modeling is highly complex, demanding advanced expertise and detailed information about the building being modeled [49]. Since the simulations are adjusted using only Istanbul's weather data, they are only valid for this location, so the findings of this study do not extend to other cities. The simulation model was also created considering a residential building; it is not valid for different building types, additional study and training on a extensive also various data set will needed. Another limitation was that the 147,456 data sets created with the full factorial method were reduced %25 due to computational power limitations. However, the dataset was still representative of the design space. Finally, the working principle of ML algorithms is a black box. Unlike physics-based models, known as white-box models, they lack transparency in illustrating the relationship between inputs and the predicted building energy consumption. However, both approaches yield good results despite their notable limitations and shortcomings. Therefore, comparing these methods and examining their suitability for building performance studies is crucial.

4.3 Contributions and Future Studies

Future directions involve applying the ML methods on different application areas. The dataset used to train surrogate models has a wide design space, which plays a critical role in building performance. However, the data needs to holistically

address design parameters such as settlement, building, and building elements at different scales. Also, integrating microclimate conditions into the building energy model within the settlement patterns developed depending on the different configurations of urban elements (such as streets, green areas, and transportation) and buildings does not represent the data set. However, the data set needs to be more effective in responding to many complex challenges, such as evaluating the potential of both settlement patterns and residential buildings regarding energy, economic, and indoor/outdoor comfort conditions. Therefore, it is planned to continuously improve the approach to solve these complex challenges with future studies. It is aimed at encouraging the implementation of this approach to contribute to producing sustainable built environments. Future research could investigate the combined effects of building design and occupant behavior on energy consumption in buildings. This study specifically focused on surrogate models in temperate humid climate regions; therefore, future research should assess the models' applicability to other climatic conditions. Some of the results obtained in this study can be examined. For instance, ANN model performance fluctuates unexpectedly with changing training set size. The reason for this fluctuation can be examined. Also, the gaps in the error plots should be investigated in the future works.

REFERENCES

[1] D. o. E. United Nations, P. D. Social Affairs, "World population prospects 2022: Data sources.(un desa/pop/2022/dc/no. 9)," 2022.

[2] O. IEA, "Tracking clean energy progress 2023," IEA Paris, France, 2023.

[3] S. Fathi, R. Srinivasan, A. Fenner, S. Fathi, "Machine learning applications in urban building energy performance forecasting: A systematic review," *Renewable and Sustainable Energy Reviews*, vol. 133, p. 110 287, 2020.

[4] EnergyPlus Development Team, *Energyplus (version 9.5.0) [computer software]*, https://energyplus.net/.

[5] eQuest Development Team, *Equest (version 3.65) [computer software]*, http://www.doe2.com/equest/.

[6] TRNSYS Development Team, *Trnsys (version 18.1) [computer software]*, http://sel.me.wisc.edu/trnsys/.

[7] T. Østergård, R. L. Jensen, S. E. Maagaard, "A comparison of six metamodeling techniques applied to building performance simulations," *Applied Energy*, vol. 211, pp. 89–103, 2018.

[8] A. Forrester, A. Sobester, A. Keane, *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.

[9] R. E. Edwards, J. New, L. E. Parker, B. Cui, J. Dong, "Constructing large scale surrogate models from big data and artificial intelligence," *Applied energy*, vol. 202, pp. 685–699, 2017.

[10] A. Ali, R. Jayaraman, A. Mayyas, B. Alaifan, E. Azar, "Machine learning as a surrogate to building performance simulation: Predicting energy consumption under different operational settings," *Energy and Buildings*, vol. 286, p. 112 940, 2023.

[11] P. Westermann, R. Evins, "Surrogate modelling for sustainable building design–a review," *Energy and Buildings*, vol. 198, pp. 170–186, 2019.

[12] A. Tsanas, A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools," *Energy and buildings*, vol. 49, pp. 560–567, 2012.

[13] A. Singh, B. LastName, "Extrapolation with machine learning based early-stage energy prediction models," *Journal of Energy Prediction*, vol. 10, no. 2, pp. 100–120, 2023.

[14] W. Zhou, J. Zhang, J. Sun, G. Sun, "Using small-scale history data to predict large-scale performance of hpc application," in *2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, 2020, pp. 787–795.

[15] T. Wortmann, A. Costa, G. Nannicini, T. Schroepfer, "Advantages of surrogate models for architectural design optimization," *AI EDAM*, vol. 29, no. 4, pp. 471–481, 2015.

[16] R. F. Woodbury, A. L. Burrow, "Whither design space?" *Ai Edam*, vol. 20, no. 2, pp. 63–82, 2006.

[17] The MathWorks, Inc., *MATLAB*, The MathWorks, Inc., Natick, Massachusetts. [Online]. Available: https : / / www . mathworks . com / products/matlab.html.

[18] G. van Rossum, T. P. D. Team, *Python programming language*, Python Software Foundation. [Online]. Available: https://www.python.org.

[19] R. Team, *Rstudio: Integrated development environment for r*, RStudio, PBC. [Online]. Available: https://www.rstudio.com/.

[20] D. Yılmaz, A. M. Tanyer, İ. D. Toker, "A data-driven energy performance gap prediction model using machine learning," *Renewable and Sustainable Energy Reviews*, vol. 181, p. 113 318, 2023.

[21] S. Petersen, S. Svendsen, "Method and simulation program informed decisions in the early stages of building design," *Energy and buildings*, vol. 42, no. 7, pp. 1113–1119, 2010.

[22] S. Seyedzadeh, F. P. Rahimian, I. Glesk, M. Roper, "Machine learning for estimation of building energy consumption and performance: A review," *Visualization in Engineering*, vol. 6, pp. 1–20, 2018.

[23] Y. Chen, M. Guo, Z. Chen, Z. Chen, Y. Ji, "Physical energy and data-driven models in building energy prediction: A review," *Energy Reports*, vol. 8, pp. 2656–2671, 2022.

[24] Y. Sun, F. Haghighat, B. C. Fung, "A review of the-state-of-the-art in data-driven approaches for building energy prediction," *Energy and Buildings*, vol. 221, p. 110 022, 2020.

[25] H.-X. Zhao, F. Magoulès, "Feature selection for predicting building energy consumption based on statistical learning method," *Journal of Algorithms & Computational Technology*, vol. 6, no. 1, pp. 59–77, 2012.

[26] T. Ekström, *Predicting the Energy Performance of Buildings: A Method using Probabilistic Risk Analysis for Data-driven Decision-support*. Lund University, 2021.

[27] M. Zorn, N. Catunda, L. Claus, N. Kobylinska, M. Frey, T. Wortmann, "Replacing energy simulations with surrogate models for design space exploration," *Bauphysik*, vol. 44, no. 6, pp. 311–316, 2022.

[28] L. Van Gelder, P. Das, H. Janssen, S. Roels, "Comparative study of metamodelling techniques in building energy simulation: Guidelines for practitioners," *Simulation Modelling Practice and Theory*, vol. 49, pp. 245–257, 2014.

[29] C. J. Hopfe, J. L. Hensen, "Uncertainty analysis in building performance simulation for design support," *Energy and Buildings*, vol. 43, no. 10, pp. 2798–2805, 2011.

[30] Z. Qian, C. C. Seepersad, V. R. Joseph, J. K. Allen, C. Jeff Wu, "Building surrogate models based on detailed and approximate simulations," 2006.

[31] H.-x. Zhao, F. Magoulès, "A review on the prediction of building energy consumption," *Renewable and Sustainable Energy Reviews*, vol. 16, no. 6, pp. 3586–3592, 2012.

[32] K. Amasyali, N. M. El-Gohary, "A review of data-driven building energy consumption prediction studies," *Renewable and Sustainable Energy Reviews*, vol. 81, pp. 1192–1205, 2018.

[33] K. Amasyali, N. El-Gohary, "Machine learning for occupant-behavior-sensitive cooling energy consumption prediction in office buildings," *Renewable and Sustainable Energy Reviews*, vol. 142, p. 110 714, 2021.

[34] S. Singaravel, J. Suykens, P. Geyer, "Deep-learning neural-network architectures and methods: Using component-based models in building-design energy prediction," *Advanced Engineering Informatics*, vol. 38, pp. 81–90, 2018.

[35] C. Robinson *et al.*, "Machine learning approaches for estimating commercial building energy consumption," *Applied energy*, vol. 208, pp. 889–904, 2017.

[36] M. M. Singh, S. Singaravel, P. Geyer, "Machine learning for early stage building energy prediction: Increment and enrichment," *Applied Energy*, vol. 304, p. 117 787, 2021.

[37] T. Østergård, R. L. Jensen, S. E. Maagaard, "Building simulations supporting decision making in early design–a review," *Renewable and Sustainable Energy Reviews*, vol. 61, pp. 187–201, 2016.

[38] A. H. Neto, F. A. S. Fiorelli, "Comparison between detailed model simulation and artificial neural network for forecasting building energy consumption," *Energy and buildings*, vol. 40, no. 12, pp. 2169–2176, 2008.

[39] R. Olu-Ajayi, H. Alaka, I. Sulaimon, F. Sunmola, S. Ajayi, "Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques," *Journal of Building Engineering*, vol. 45, p. 103 406, 2022.

[40] A. Ali, R. Jayaraman, E. Azar, M. Maalouf, "A comparative analysis of machine learning and statistical methods for evaluating building performance: A systematic review and future benchmarking framework," *Building and Environment*, p. 111 268, 2024.

[41] M. Zekić-Sušac, A. Has, M. Knežević, "Predicting energy cost of public buildings by artificial neural networks, cart, and random forest," *Neurocomputing*, vol. 439, pp. 223–233, 2021.

[42] M. W. Ahmad, M. Mourshed, Y. Rezgui, "Trees vs neurons: Comparison between random forest and ann for high-resolution prediction of building energy consumption," *Energy and buildings*, vol. 147, pp. 77–89, 2017.

[43] R. A. P. E. Danhaive, "Structural design synthesis using machine learning," Ph.D. dissertation, Massachusetts Institute of Technology, 2020.

[44] T. Østergård, R. L. Jensen, S. Maagaard, "Interactive building design space exploration using regionalized sensitivity analysis," in *The 15th International Conference of IBPSA*, International Building Performance Simulation Association, 2017, pp. 726–735.

[45] J. Gareth, W. Daniela, H. Trevor, T. Robert, *An introduction to statistical learning: with applications in R*. Spinger, 2013.

[46] S. Seyedzadeh, F. P. Rahimian, P. Rastogi, I. Glesk, "Tuning machine learning models for prediction of building energy loads," *Sustainable Cities and Society*, vol. 47, p. 101 484, 2019.

[47] T. Østergård, R. L. Jensen, F. S. Mikkelsen, "The best way to perform building simulations? one-at-a-time optimization vs. monte carlo sampling," *Energy and Buildings*, vol. 208, p. 109 628, 2020.

[48] S. Seyedzadeh, F. P. Rahimian, S. Oliver, S. Rodriguez, I. Glesk, "Machine learning modelling for predicting non-domestic buildings energy performance: A model to support deep energy retrofit decision-making," *Applied Energy*, vol. 279, p. 115 908, 2020.

[49] M. Bourdeau, X. qiang Zhai, E. Nefzaoui, X. Guo, P. Chatellier, "Modeling and forecasting building energy consumption: A review of data-driven techniques," *Sustainable Cities and Society*, vol. 48, p. 101 533, 2019.

[50] Z. Wang, Y. Wang, R. Zeng, R. S. Srinivasan, S. Ahrentzen, "Random forest based hourly building energy prediction," *Energy and Buildings*, vol. 171, pp. 11–25, 2018.

[51] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, S.-H. Deng, "Hyperparameter optimization for machine learning models based on bayesian optimization," *Journal of Electronic Science and Technology*, vol. 17, no. 1, pp. 26–40, 2019.

[52] M. Claesen, B. De Moor, "Hyperparameter search in machine learning," *arXiv preprint arXiv:1502.02127*, 2015.

[53] J. Bergstra, Y. Bengio, "Random search for hyper-parameter optimization.," *Journal of machine learning research*, vol. 13, no. 2, 2012.

[54] J. J. Allaire *et al.*, *Keras: R interface to 'keras'*, R package version 2.9.0. [Online]. Available: https://CRAN.R-project.org/package=keras.

[55] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, F. Leisch, *E1071: Misc functions of the department of statistics, probability theory group (formerly: E1071), tu wien*, R package version 1.7-13, 2023. [Online]. Available: https://CRAN.R-project.org/package=e1071.

[56] A. Liaw, M. Wiener, *Randomforest: Breiman and cutler's random forests for classification and regression*, R package version 4.7-1.1. [Online]. Available: https://CRAN.R-project.org/package=randomForest.

[57] S. Papadopoulos, E. Azar, W.-L. Woon, C. E. Kontokosta, "Evaluation of tree-based ensemble learning algorithms for building energy performance estimation," *Journal of Building Performance Simulation*, vol. 11, no. 3, pp. 322–332, 2018.

[58] T. Catalina, J. Virgone, E. Blanco, "Development and validation of regression models to predict monthly heating demand for residential buildings," *Energy and buildings*, vol. 40, no. 10, pp. 1825–1832, 2008.

[59] A. S. Ahmad *et al.*, "A review on applications of ann and svm for building electrical energy consumption forecasting," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 102–109, 2014.

[60] B. Dong, C. Cao, S. E. Lee, "Applying support vector machines to predict building energy consumption in tropical region," *Energy and Buildings*, vol. 37, no. 5, pp. 545–553, 2005.

[61] Q. Li, Q. Meng, J. Cai, H. Yoshino, A. Mochida, "Applying support vector machine to predict hourly cooling load in the building," *Applied Energy*, vol. 86, no. 10, pp. 2249–2256, 2009.

[62] S. MANGAN, "Development of a web-based decision support tool for sustainable residential building design and retrofit," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 36, no. 4, 2021.

[63] S. D. Mangan, G. K. Oral, I. E. Kocagil, I. Sozen, "The impact of urban form on building energy and cost efficiency in temperate-humid zones," *Journal of Building Engineering*, vol. 33, p. 101 626, 2021.

[64] Turkish Statistical Institute, *Population and housing census, 2011*, 2013.

[65] Ministry of Family and Social Policies, "Research on family structure in türkiye," Ministry of Family and Social Policies, 2011.

[66] Turkish Standards Institution, *Ts 825 thermal insulation requirements for buildings*, Ankara, Turkey, 2013.

[67] DesignBuilder Software Ltd., *Designbuilder (version 6.1)*, Available from https://www.designbuilder.co.uk/.

[68] S. D. Mangan, "A performance-based decision support workflow for retrofitting residential buildings," *Sustainability*, vol. 15, no. 3, p. 2567, 2023.

[69] B. Lee, M. Trcka, J. L. Hensen, "Building energy simulation and optimization: A case study of industrial halls with varying process loads and occupancy patterns," in *Building simulation*, Springer, vol. 7, 2014, pp. 229–236.

[70] B. Lee, N. Pourmousavian, J. L. Hensen, "Full-factorial design space exploration approach for multi-criteria decision making of the design of industrial halls," *Energy and Buildings*, vol. 117, pp. 352–361, 2016.

[71] A. Liaw, M. Wiener, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

A PERFORMANCE DETAILS OF SURROGATE MODELS

Here's a more detailed explanation of the architecture and additional details of the models shown in Tables A.1, A.2, and A.3.

The ANN model's input layer consisted of 19 neurons corresponding to the 19 input features. Each neuron in this layer took one input feature and passed it to the hidden layer. The hidden layer contained 36 neurons. Each neuron in this layer received a weighted sum of inputs from all neurons in the input layer, applied an activation function, and passed the result to the output layer. The output layer consisted of a single neuron, which produced the final output of the network. As an optimization algorithm, a variant of the backpropagation algorithm was typically used to optimize the weights. Also, the model was trained for a maximum of 200 iterations.

Table A.1 Detailed neural network architecture

Property	**Description**
Input Layer	19 Neurons (corresponding to 19 input features)
Hidden Layer Activation Function	36 Neurons Sigmoid
Output Layer Activation Function	1 Neuron Linear (for regression)
Total Units	57
Weight Decay	0.01 (Regularization technique to prevent overfitting)
Training Details	200 Iterations
Optimization Algorithm	Backpropagation

In this study, the RF model is designed to predict continuous values rather than categorical labels. Regression was suitable for tasks where the output variable is numeric. At each node in the decision trees, 4 randomly selected variables were considered for splitting. This key parameter (mtry) in RF helps reduce overfitting and improve model generalization. The Mean of Squared Residuals is a metric that quantifies the average squared discrepancy between the simulation and predicted values. Reduced values indicated good model performance. Lastly, the model explained 99.7% of the variability in the response variable. A high percentage suggested that the model fits the data very well.

The RF model built 500 individual decision trees. Increasing the number of trees generally improves model performance and stability but also increases computational cost Figure A.1.

SVM architecture was set up for regression using the epsilon-SVR model type.

Table A.2 Random forest model architecture

Property	**Description**
Type of Random Forest	Regression
Number of Trees	500
No. of Variables Tried at Each Split	4
Mean of Squared Residuals	1.142251
% Variance Explained	99.7

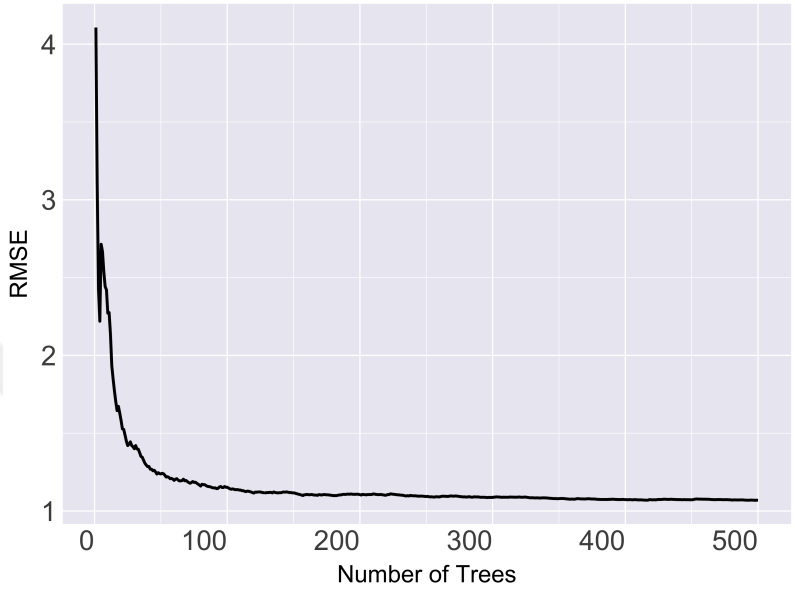

Figure A.1 Relationship between number of trees and RMSE

Model employed a Gaussian RBF kernel with a hyperparameter sigma set to 0.01. The model's epsilon parameter was 0.1, and the cost parameter was 10. The model used 142 support vectors. The parameters, including epsilon and cost, were configured to optimize the model's performance for regression tasks. This setup suggested a balance between model complexity and regularization to prevent overfitting while effectively capturing the underlying data patterns.

Table A.3 Support vector machine model architecture

Property	**Description**
Model Type	eps-svr (regression)
Kernel Type	Gaussian Radial Basis Function (rbfkernel)
Kernel Hyperparameter	Sigma: 0.01
Epsilon	0.1
Cost (C)	10
Number of Support Vectors	142

B
DETAILS OF TUNED MACHINE LEARNING MODELS

The details of tuned ML models were discussed in the Chapter 3 and shown in the Tables B.1, B.2, and B.3. In the tables, the parameters of the model that best predicts building energy performance were highlighted in red.

Table B.1 Relation between the hyperparameters of ANN and RMSE

		RMSE			
size	**decay**	**DT0.1**	**DT0.2**	**DT0.4**	**DT0.8**
9	0.000	2.818	2.838	2.716	5.215
9	0.001	1.086	0.841	0.969	1.040
9	0.010	1.397	2.301	0.649	1.775
9	0.100	1.289	1.016	0.814	1.341
18	0.000	0.662	0.310	1.477	0.505
18	0.001	0.637	0.503	1.644	0.643
18	0.010	0.878	0.504	4.817	0.416
18	0.100	0.848	0.540	0.843	0.579
27	0.000	0.288	0.871	0.726	0.350
27	0.001	0.258	1.605	0.632	0.269
27	0.010	0.228	0.279	0.584	0.353
27	0.100	0.315	0.583	0.454	0.476
36	0.000	0.357	1.023	0.377	0.778
36	0.001	0.234	0.479	0.536	0.813
36	0.010	0.208	1.263	0.692	0.372
36	0.100	0.255	0.526	0.409	0.814

Table B.2 Relation between the hyperparameters of RF and RMSE

Resample	mtry	nodesize	ntree	DT.01	DT.02	DT.04	DT.08
Fold3	2	1	200	8.547	8.086	8.320	7.604
Fold2	2	3	100	8.161	7.962	8.265	7.687
Fold1	2	9	500	8.762	8.399	8.013	7.787
Fold5	3	1	100	4.870	4.379	4.248	3.810
Fold5	3	3	200	4.546	4.239	4.156	3.627
Fold4	3	9	100	5.319	4.680	4.218	3.894
Fold4	4	1	200	2.620	2.277	2.101	1.773
Fold5	4	3	200	2.585	2.255	2.100	1.725
Fold4	4	9	200	2.880	2.614	2.216	1.834
Fold1	5	1	200	1.624	1.399	1.130	0.902
Fold2	5	3	200	1.602	1.305	1.073	0.933
Fold3	5	9	200	1.928	1.556	1.299	1.000

Table B.3 Relation between the hyperparameters of SVM and RMSE

		RMSE			
C	sigma	DT0.1	DT0.2	DT0.4	DT0.8
1	0.01	1.107	1.041	0.984	0.939
1	0.02	1.081	0.949	0.885	0.869
1	0.05	1.396	1.133	0.977	0.887
1	0.10	2.254	1.724	1.346	1.105
1	0.20	4.938	3.587	2.577	1.854
1	0.50	15.191	12.339	9.084	6.187
1	1.00	18.465	17.438	15.546	12.695
1	2.00	19.169	18.830	18.176	17.010
1	5.00	19.417	19.369	19.330	19.227
10	0.01	1.069	0.998	0.948	0.913
10	0.02	1.070	0.943	0.882	0.868
10	0.05	1.392	1.126	0.974	0.886
10	0.10	2.253	1.721	1.345	1.105
10	0.20	4.937	3.585	2.577	1.854
10	0.50	14.773	12.088	8.999	6.169
10	1.00	18.225	17.077	15.087	12.274
10	2.00	19.077	18.681	17.901	16.595
10	5.00	19.389	19.349	19.296	19.169
100	0.01	1.070	0.998	0.947	0.913
100	0.02	1.070	0.943	0.882	0.868
100	0.05	1.392	1.126	0.974	0.886
100	0.10	2.253	1.721	1.345	1.105
100	0.20	4.937	3.585	2.577	1.854
100	0.50	14.773	12.088	8.999	6.169
100	1.00	18.225	17.077	15.087	12.274
100	2.00	19.077	18.681	17.901	16.595
100	5.00	19.389	19.349	19.296	19.169

C
RELATION BETWEEN PREDICTED AND SIMULATION VALUES

Scatter plots illustrating a correlation between simulation and predicted values for interpolation are shown in Figures C.1 and C.2 . Tables indicated that there are no outliers.

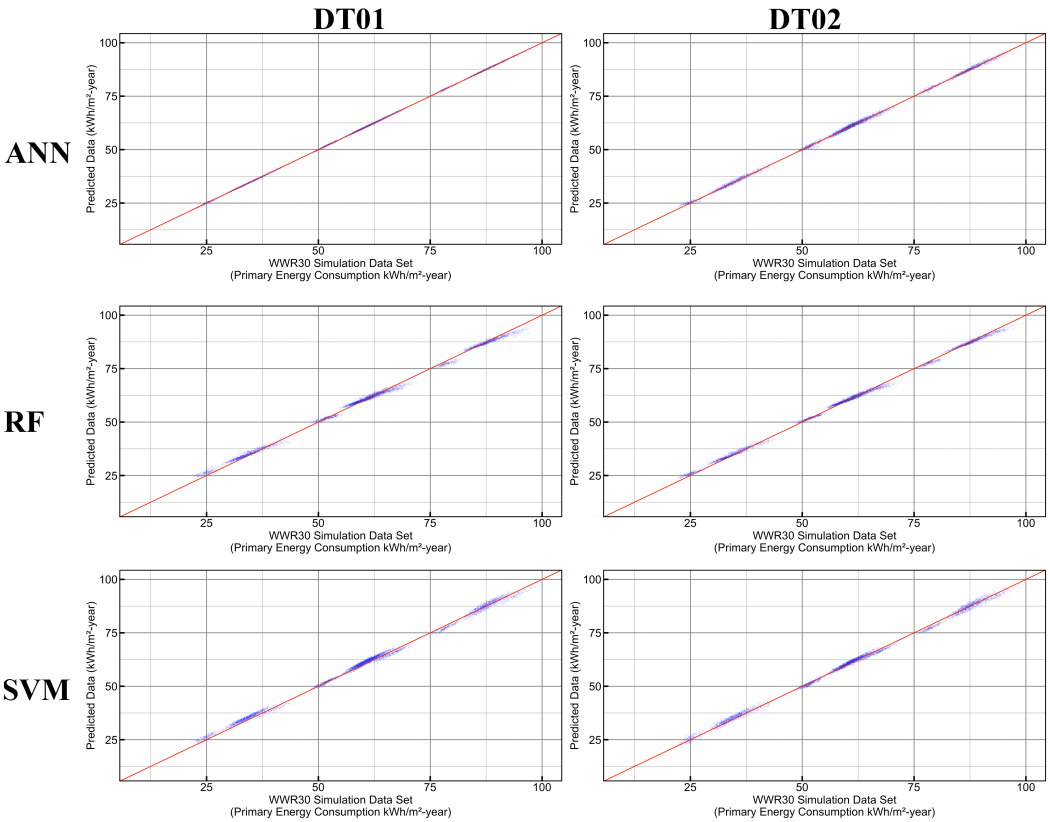

Figure C.1 Scatter plots of simulation and predicted values of interpolation for 10% and 20% train sizes

On the other hand, scatterplots in the Figure C.3 showed more tendency to have outliers for the extrapolation case. Results are not surprising due to the very nature of extrapolation itself. In addition, it was already reported that the performance of models was poor for the extrapolation case.

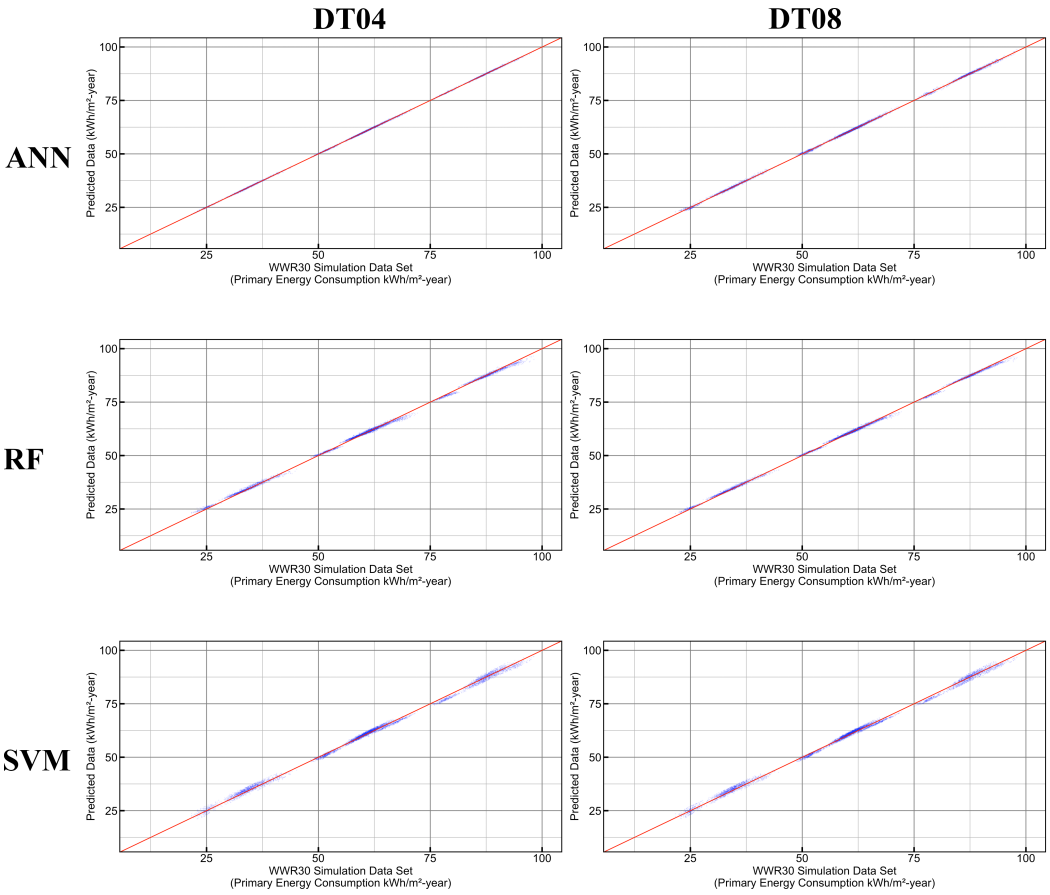

Figure C.2 Scatter plots of simulation and predicted values of interpolation for 40% and 80% train sizes

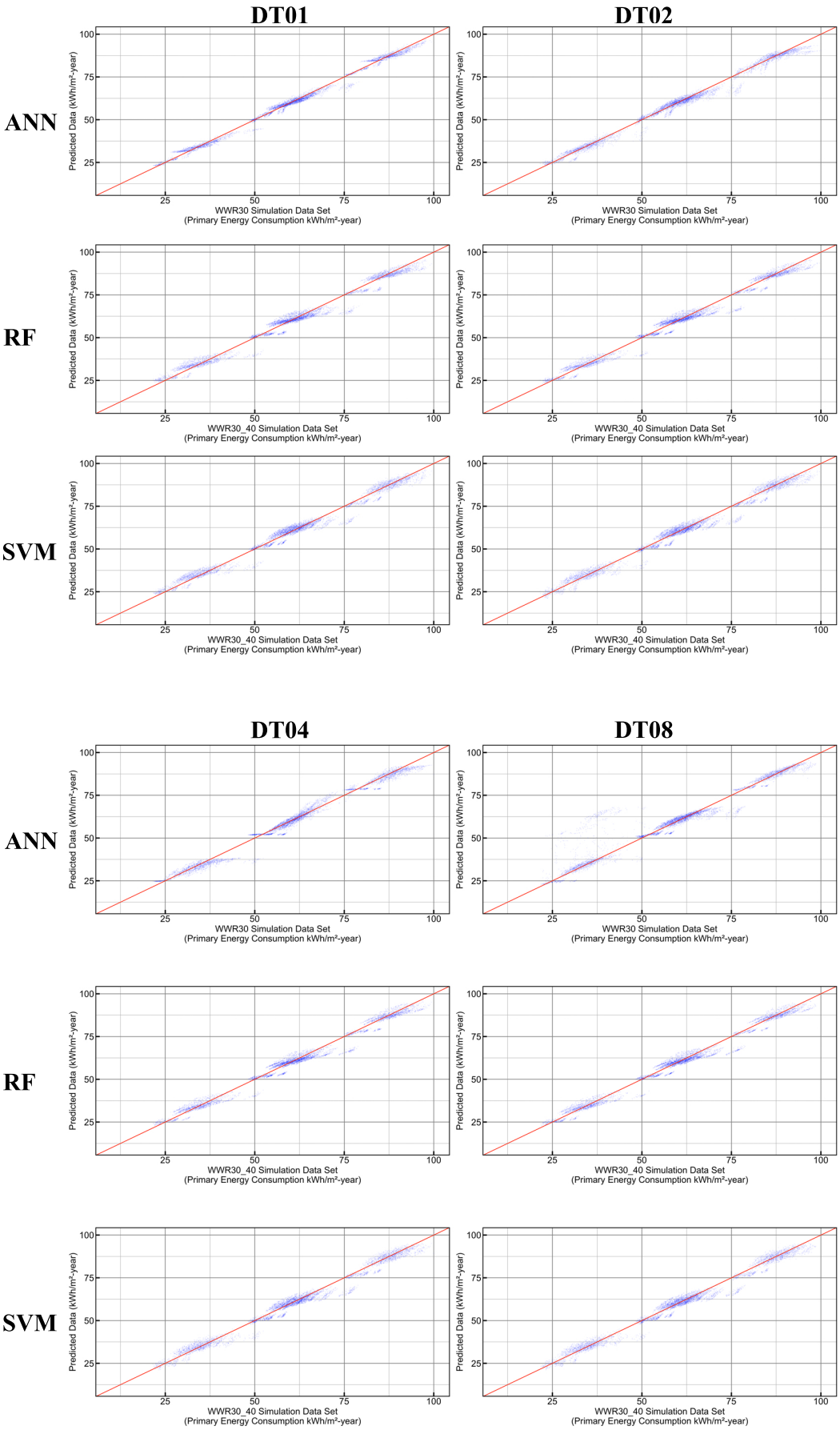

Figure C.3 Scatter plots of simulation and predicted values of interpolation for all the train sizes

PUBLICATIONS FROM THE THESIS

Conference Papers

1. A.R. Efe, S.D. Mangan, "Tasarım alanı keşfinde üç farklı vekil modelin etkinlik düzeyinin irdelenmesi," *4.Ulusal Yapı Fiziği ve Çevre Kontrolü Kongresi*, vol. 4, pp. 196-204, 2024.