Metamodeling of Energy and Operational Carbon in Detached Accessory Dwelling Units

Preston Pape

A thesis submitted in partial
fulfillment of the requirements
for the degree of

Master of Science in Architecture (Design Computing)

University of Washington
2022

Committee:

Rick Mohler, Chair

Tomás Méndez Echenagucia

Teresa Moroseos

Program Authorized to Offer Degree:

Architecture

University of Washington

Abstract

Metamodeling of Energy and Operational Carbon in Detached Accessory Dwelling Units

Preston Pape

Chair of the Supervisory Committee:

Rick Mohler
Department of Architecture

The rapidly escalating cost of housing has created a crisis in the United States that stems from a lack of housing supply, which is exacerbated by single-family zoning. While revising this policy is a needed course of action, doing so is slow and politically arduous. In the interim, many architects and planners propose allowing a detached accessory dwelling unit, or DADU, in the rear yard of single family lots to provide more housing options. This strategy has increased housing supply in west coast cities such as Portland, Seattle, and Los Angeles. Leveraging this trend, this research involves the creation of a web-based application for predicting DADU energy consumption, utility costs, and carbon emissions in the early stages of design through the use of surrogate modeling. This enables both homeowners and designers to quickly estimate and improve energy performance to reduce long term costs and increase the production of DADUs. The dataset was created by simulating the entire examined design space using Rhinoceros, Grasshopper, Ladybug Tools, and EnergyPlus. Simulation data is stored in .CSV files which were preprocessed using Pandas. XGBoost was selected as the machine learning option, with hyperparameter optimization via bayesian search. The web app was then constructed using Streamlit and hosted on Heroku.

Table of Contents

List of Figures

List of Tables

Glossary

(DETACHED) ACCESSORY DWELLING UNIT (ADU/DADU): Separate living space within a house or on the same property as an existing house.

ADIABATIC: Boundary condition of which there is no temperature difference across the surface.

BAYESIAN SEARCH: Method used to search for optimal solution using Bayesian statistics. Commonly used for hyperparameter turning (see below term).

BOUNDARY REPRESENTATION (BREP): Method for representing solid models using limits. Model components consist of faces, edges, and vertices.

COEFFICIENT OF PERFORMANCE (COP): Ratio of useful heating or cooling capacity provided to electrical energy consumed. Used to compare efficiency of HVAC systems.

DATAFRAME: Two-dimensional data structure for containing tabular data. Size-mutable, potentially heterogenous.

DESIGN SPACE: Defined parameters of interest to explore. Typically refers to set of variables in which an optimal combination is of interest.

EMBODIED CARBON: Carbon emissions which occur during construction of a building. Constitutes larger portion of lifetime emissions as building performance increases.

ENERGY COST BURDEN (ECB): Ratio of total household energy cost to median household income by neighborhood.

ENERGY USE INTENSITY (EUI): Benchmarking unit which refers to total annual energy usage divided by gross floor area of building. Defined in SI units as kWh/m^2 and in US customary units as $kBTU/ft^2$.

F-SCORE: Refers here specifically to a metric used in the XGBoost (see term below) Python package to represent feature importance (see also below). Calculated by tallying number of decision trees containing a decision point based on each individual feature.

FEATURE: Individual measurable property of interest. In machine learning, usually numeric, but can be string.

FEATURE IMPORTANCE: Comparative weight of each feature on prediction (See also F-score).

FEATURE SELECTION: Preliminary step used in machine learning methods to remove redundant or confounding features to make dataset more manageable.

FILTERING: Process in which aging residences become increasing affordable. Results in lower-income households being more likely to rent units of lesser quality.

GLOBAL WARMING POTENTIAL (GWP): Primary impact category of a life cycle assessment. Describes embodied carbon in terms of equivalent weight, typically in kg of CO_2

GRASSHOPPER: Visual scripting plugin for Rhinoceros (see term below).

HONEYBEE: Plugin for Grasshopper for energy modeling. Facilitates interfacing between EnergyPlus, OpenStudio, and Grasshopper/Rhinoceros.

HYPERPARAMETER: Parameter whose values are used to control the learning process. Affects speed and quality of learning process. Values set manually or by optimization/search.

HYPERPARAMETER OPTIMIZATION: Method for finding optimal hyperparameter settings. Bayesian search, random search, and grid search are commonly employed.

INFILTRATION RATE: Rate at which air is introduced to building through the envelope. Defined here in units m^3/sec (per m^2 of building facade).

LIFE CYCLE ASSESSMENT: Technique for assessing the environmental aspects associated with a product or building over its entire life cycle.

LOSS: Measure or function of how far from predicted value is from true value. Penalty for bad prediction.

METAMODELING: Engineering method used when outcome of interest cannot be easily measured or computed. Also known as *surrogate modeling*. In this case, use of machine learning to predict key values in place of simulation or real world data.

MINI SPLIT HEAT PUMP (MSHP): Ductless heat pump system consisting of outdoor compressor/condenser and indoor air handling unit. Provides heating and cooling.

OPERATIONAL CARBON: Carbon emissions which occur during use, management, and maintenance of building.

RENT BURDENED: Defined by United States Department of Housing and Urban Development as a household that spends greater than 30% of its gross income on rent.

RHINOCEROS: 3D modeling application commonly used for architectural design developed by McNeel & Associates.

SETBACK: Minimum distance which a building or other structure must be removed from street, lot line, or other structure.

SHAP VALUE: Method based on cooperative game theory used to increase transparency and interpretability of machine learning models. Developed at University of Washington.

SOLAR HEAT GAIN COEFFICIENT (SHGC): Fraction of solar radiation admitted through window, door, skylight, or other transparent/translucent surface (ranges between 0 and 1, commonly between 0.2 and 0.9).

WINDOW TO WALL RATIO (WWR): Proportion of exterior facade that contains glazing (windows). Represented as decimal value.

XGBOOST: Optimized distributed gradient boosting library for machine learning.

Acknowledgments

My research would not have been possible without the support of those around me. I would like to foremost thank my committee chair Prof. Rick Mohler for his extensive knowledge in housing and policy and for his support in helping me situate my thesis within these areas. I would also like to thank my committee members Prof. Tomás Méndez Echenagucia and Teresa Moroseos; Tomás for his help in reinforcing my computational methods and for programming assistance at many points, Teresa for her insights into EnergyPlus model validation.

I would like to credit Prof. Christopher Meek for his input regarding selection of simulation parameters. As part of the 47° North professional mentorship program, Bryan Samuel of Olson Kundig provided feedback on visual representation of the project. Prof. Arthur Acolin directed me towards previous research on housing from a real estate perspective. A very special thanks to Matthew Hutchins and CAST Architecture for their willingness to share DADU plans for analysis. I would also like to express my gratitude to Mills Selkregg for her wisdom and guidance in the realm of data science. At the tail end of development of my web application, Nicolas Welch of the Seattle Office of Planning and Community Development granted in-depth usability feedback from a professional, civic standpoint.

Finally, I am forever grateful for the constant encouragement from my family. They were there for me all the times I vented about deadlines and general stressing. I also need to thank Mia Lorentsen for her unwavering support and for dealing with my late nights and even earlier mornings.

Chapter 1

INTRODUCTION

1.1 Context

Addressing the unprecedented housing and climate crisis in the United States requires a twin focus on equity and sustainability through progressive policy change and innovative design and construction strategies. The predicament facing the housing market is exacerbated by zoning regulation, as it artificially limits the supply of housing in large areas of fast-growing cities. This was historically applied intentionally to segregate cities by race and class beginning in the 1920s, and is referred to as redlining. (Flournoy 2021). While this research does not delve too deeply into this, it is important to understand the current context. Resultantly, rising housing costs stem from an ever-increasing lack of housing supply. For many, housing costs account for a greater share of their income, restricting their ability to afford basic necessities and decreasing overall quality of life. Due to the commodification of the housing market, increasing amounts of residents are experiencing homelessness and unable to find a foothold back into permanent housing normalcy (Routhier 2021). Stabilized property values, as opposed to the current speculation, are socially beneficial to all residents (Brinig and Garnett 2013). As the realities of climate change begin to set in, the method employed to ease the housing crisis must also address sustainability. Further, people of color and those with lower levels of income are disproportionately affected by skyrocketing housing prices, and are also those most at risk of the enlarged rate of natural disasters due to our changing climate. (Thomalla et al. 2006). Low income residents currently encounter asymmetric energy costs when compared to wealthier citizens, widening the disparity in the cost of living (Kontokosta, Reina, and Bonczak 2020).

Traditional means of increasing housing stock is by increasing density and constructing multifamily housing structures. However, since the 1920's, the trend has been larger plots of

land per owner, leading to urban sprawl (Hertz 2015a). In turn, this sprawl has created a dependence on automobiles and one of the most energy-intensive housing typologies, being single-family dwellings. Proposed changes to land use policy are frequently met with pushback from homeowners due to lesser demand equaling a reduced property value (Brinig and Garnett 2013). In an ideal world, single-family residential zones would be converted into multifamily residential or mixed multifamily (residential and commercial). However, this strategy would hurt rather than help, due to the effects of displacement. As this is not feasible, cities around the country have begun to experiment and implement new regulations allowing for the construction of accessory dwelling units (ADUs). Accessory dwelling units are separate residential units on single-family residential lots, typically used as rental housing or for housing an aging loved one close to home.

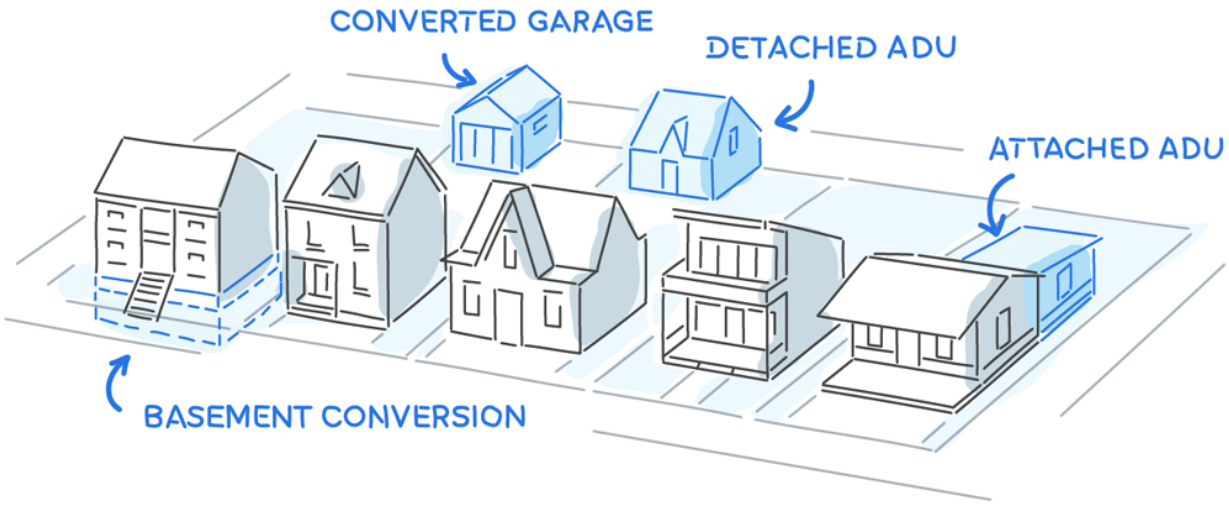


Figure 1.1: Types of backyard cottages which includes ADUs (Housable 2021)

ADUs fall into two sub-categories: attached and detached, with the former being an expansion onto the existing structure, while the latter is separate. This thesis will be focusing solely on detached accessory dwelling units, or DADUs, but may use both terms interchangeably. Due to their limited size, ADUs encourage individuals to downsize the amount of space they occupy, reducing energy use and carbon footprint per capita. ADUs being a comparatively affordable investment, combined with their intrinsic ability to increase property value,

has led the way for new policy to allow for their construction (Chapple et al. 2017). Beyond restrictive zoning and permitting regulations, price contributes to the rate of accessory dwelling unit construction. In fact, recent 2019 survey data collected in Seattle shows that homeowners desire an increased focus on sustainability and reduced cost (City of Seattle 2019b). Developing an effective method for reducing energy use will decrease operational costs and could lead to an increased construction rate for ADUs, benefitting both owners and occupants, while addressing climate change and housing scarcity. It is of utmost importance for architects, policymakers, and residents to recognize that energy use and housing availability are not solely problems of climate or equity, respectively.

Traditional methods for evaluating energy performance of a proposed structure involve what is referred to as the modeling approach. This method requires an accurate 3D model to be constructed with specific materials applied, along with geographic information. The model is then simulated over the course of one calendar year to evaluate energy performance metrics, including energy use intensity (EUI). EUI is defined as the total energy consumed by a building over the course of one year, divided by gross floor area of the building kBTU/ft^2. This metric is helpful to compare energy use between buildings of varying sizes and typologies. EnergyPlus, an open-source building energy simulation program, is a widely-used framework for this method in the United States. While precise, the modeling approach is time consuming on the scale of an individual design project. In contrast, the statistical approach is gaining popularity. These methods for calculating building energy performance rely on statistical inference using specified design inputs. This thesis investigates the use of surrogate modeling as an improved method to derive EUI. Surrogate modeling uses statistics and computation to estimate values to a high degree of accuracy. Using machine learning models to calculate EUI saves time for designers, but requires extra upfront resources to train, typically in the form of a rich database.

1.2 Related work

Efforts to increase the production of ADUs across Seattle have varied in goal and scope. Beginning in 1994, Seattle began to allow ADUs only in single-family zones due to requirements in the Washington Housing Policy Act (Levy and Pennucci 2019). However, the limitation at the time was that only attached accessory dwelling units (AADU) were allowed. Following

up on this, Seattle expanded the ADU rules to allow for DADUs in 2010. As only 50 out of 100,000 eligible properties constructed ADUs since this change, Seattle began exploring policies to make accessory dwelling units more accessible in 2014. In 2019, an initiative to inform potential ADU owners named ADUniverse was created as a joint project between Seattle Office of Planning and Community Development (OPCD) and the Data Science for Social Good Program at the University of Washington (Mohler et al. 2019). ADUniverse's goal was to understand where ADUs have historically been built in Seattle, and to identify potential issues with eligible lots, while attempting to estimate costs associated. Generally, ADUs are equally prevalent across all viable zoning types across the city, as seen in figure 1.2. As part of this project, a survey was conducted later in September of 2019 to begin to understand the design criteria most important to potential owners. This survey received a total of 568 responses, with 85% of respondents identifying as current homeowners, and 35% of all respondents identifying as working as "an architect or similar professional". Results showed that "low cost" was the criteria in first place with 48% of those surveyed responding "very important", followed by 'green building' at 46%. Other more-specific priorities suggested include: longer-term environmental costs, site specific considerations, and predictability in both construction and cost (City of Seattle 2019b). The results of this survey were then used to inform design submissions from firms, with ten of these being selected later by the City of Seattle to be featured on their website as pre-approved plans.

As the ADUniverse survey was administered to potential ADU owners, it lacked data on how existing units are being used. Seattle OPCD and University of Washington are planning a joint survey of owners and occupants on their usage for Summer 2022. However, existing data from both Portland, Oregon, and California (statewide survey) is relevant. In Portland it was found that 77% of ADU owners initially built their units with the intent of renting out through long-term leases, while 35% said the purpose was housing family or close friends (note that these values sum to greater than 100%, as respondents could pick more than one option) (Gebhardt, Gilden, and Kidron 2018). When asked how these units are currently being used, 53% chose currently occupied/someone's primary residence, while 31% said the current use was short-term housing. While a direct delta cannot be drawn as the response criteria differs, it appears there has been somewhat of a shift towards short-term rental usage. Short-term rental is described in the report as services similar to Airbnb. In California on

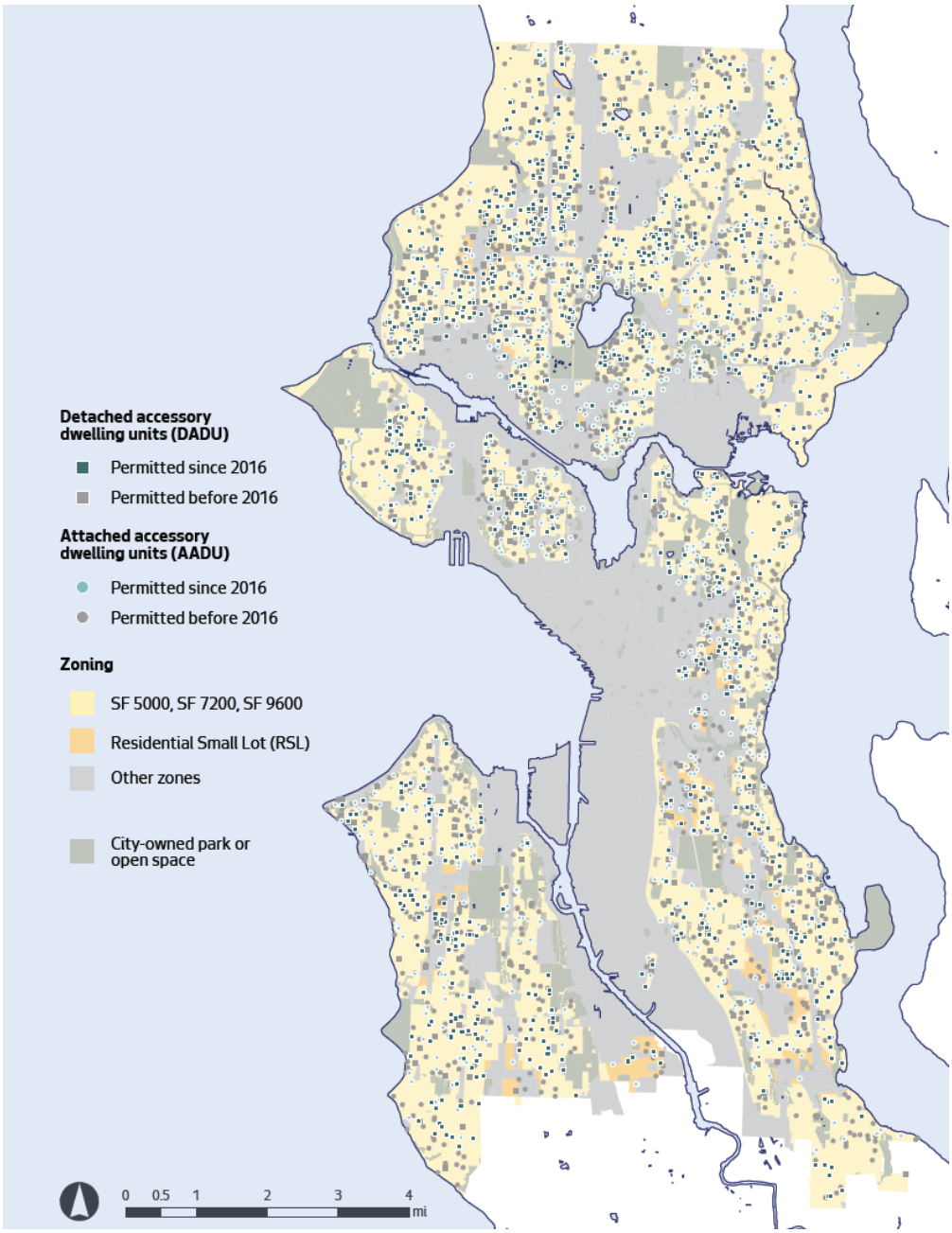


Figure 1.2: Distribution of ADU permits throughout Seattle (Welch et al. 2021)

the other hand, results showed 51% of new units were utilized as income-generating rentals, with 16% used as no-cost housing for a relative, and only 8% of units as short-term rentals (Chapple, Ganetsos, and Lopez 2021).

While these pre-approved designs are from many renowned firms who do indeed focus on sustainability, they will fall short on energy efficiency. Due to the site-agnostic nature of pre-approved plans, unique characteristics of each site will have influences on their effectiveness: shading from surrounding objects and structures, orientation to existing buildings, solar gain, etc. On the other hand, being pre-approved, many headaches regarding permitting and construction can be bypassed. To fully remedy this situation a modular approach would be recommended. In the meantime, the collective improvements to access and modifications to ADU rules has proven successful. Following the new rules which the City of Seattle implemented in 2019 regarding ADUs and the collection of 10 pre-approved DADU designs in 2020 as part of ADUniverse, ADU production has increased immensely. As this comes during a global pandemic, the values would likely be larger under normal circumstances. According to OPCD, in 2020 the number of AADU and DADU permits increased 53% and 112% respectively, over similar numbers from 2019 (Welch et al. 2021). In other words, 237 attached ADUs and 276 detached ADUs were added. With this increase in production comes a need to continue to maintain this boost in production while simultaneously optimizing the energy use of these newly permitted units.

1.3 Research questions

As ADU production ramps up dramatically in Seattle, and across the country, there must be an emphasis on the sustainability of each new construction. The hope is that by providing such a tool, users are empowered to finance and construct higher performance dwelling units.

Primary research questions

- Can a metamodel accurately predict energy consumption, carbon emissions, and utility costs of DADUs located in Seattle, Washington?
- How does the energy performance of sampled DADUs compare to similarly specified DADU designs predicted using the web app?

- Are design results generated by the tool able be utilized to verify city planning policy initiatives? (see below)

Secondary questions

- Which design constraints impact DADU energy consumption the greatest?
- Policy: what effects does the removal of rear yard setbacks in the *proposed* setbacks option have on energy use intensity?
- Policy: if neighbors build DADUs at the lot line sharing walls, is this more energy efficient?

1.3.1 Aims and objectives

The primary goal of my research is to provide designers and homeowners with the ability to see estimates of energy and operational carbon consequences, as well as energy costs in real-time. The hope is that this would greatly increase performance of new DADUs designed and built across the country, but beginning in Seattle, Washington. By leveraging the power of machine learning, DADU energy performance can be predicted accurately in real-time under existing land use and zoning codes. Increasing energy performance can also reduce costs, making the prospect of constructing a DADU a more worthwhile investment for the homeowner while reducing utility costs for potential tenants providing additional value to the community or neighborhood. Further, results from the web tool as is, or modified for more specific study could be utilized to encourage land use policy changes that enhance energy performance while providing more efficient use of land.

Designing a functional web tool for the prediction of impacts (energy, carbon, costs) of DADUs to increase production and performance of new constructions is the primary objective of this research. In designing the web tool, the aim is accessibility for homeowner users, with the option of more advanced utility for designer users. Design variables will be labeled in their technical terms but with simple explanations accompanying them for clarity. Users should be able to define design inputs, a time period for prediction, and be able to visualize both a simple 3D model of the DADU and a plot contrasting between prior prediction results. Finally there should be a means for downloading or saving results from a session.

1.4 Thesis outline

In chapter 2, the background history and literature review of housing, building performance simulation, and machine learning techniques will be discussed. Within the housing component, the impact of energy usage and costs on affordability of housing is covered. Likewise, the question of how accessory dwellings can act as a solution is answered. Chapter 3 covers the methodology utilized in this study. Points such as choice of design variables, machine learning model selection, and software used are discussed. Further, data collection, preprocessing, and analysis is explained. Chapter 4 reviews the results of the study including: surrogate model accuracy, web application usability, and comparisons between predictions and real-world outcomes. In Chapter 5, conclusions are drawn from the study and discussed as well as future work and potential use-cases of the web application and methodology.

Chapter 2

BACKGROUND

Homeownership is the primary means to grow wealth for most households in the United States. For an increasing number of households, this is financially impossible. Those lucky enough to afford a mortgage may be able to use the equity in their home to purchase a second property to rent as landlord. However, many more are caught in the cycle of renting. Renting has become the new culture du jour for many Americans. Between 1960 and 2017, the median house price in Seattle increased 286%, while the median household income increased by only 59%. This disparity has created a challenge in which the average income earner cannot save due to high rents making them unable to afford the down payment or mortgage payments for even a historically modest "starter" home (Tekin 2021).

This challenge has been exacerbated by residential trends and restrictive zoning. By 1912, the first construction boom of bungalow-style homes was in full swing across the United States. These new bungalow dwellings consumed much more land per unit than multifamily apartments and even the single family homes that preceded them. Average lot sizes increased, decreasing the per unit supply of available land for construction and driving up housing prices on average. Along with these changes, the first zoning policies were enacted to protect wealth held in housing by the upper class and as a means of racial segregation. The automobile made urban sprawl possible and led to the creation of the suburbs as a result. In turn, this new paradigm of American dependency on the automobile dramatically raised the energy consumption per household, with more than 50% of households owning at least one vehicle by 1920 (more on this in Section 2.3.1) (Hertz 2015a). Overall cost of living doubled during this period, yet housing prices increased nearly six-fold. Interestingly, bungalows were harshly criticized for "ripping apart the fabric of the city"- echoing sentiments surrounding the multifamily apartment densification of modern cities (Hertz 2015a).

Cities such as Seattle, New York City, and San Francisco are in similar housing predica-

ments and can be described as inelastic housing markets, meaning housing development does not respond normally to an increase in demand (Colburn and Aldern 2022). Seattle and the other mentioned cities distinct lack of elastic recourse to a housing shortage can be attributed to a combination of difficult topography and regulatory constraints. As terrain cannot be modified to a meaningful extent, change to development of housing in Seattle and the United States must be addressed through legislation. However, politicians at the local level have the greatest leverage to affect change. This becomes problematic when we consider that the majority of constituents in city politics tend to be homeowners, wielding power to defend land valuation and to maintain lower property taxes (Brinig and Garnett 2013). William Fischel, coined the term Homevoter Hypothesis to explain this phenomenon in which suburban homeowners tend to support regulation that increases home value above all else. Maintaining current zoning policy has limited the supply of developable land- the root cause of the current housing crisis. In the face of this uphill battle, ADUs are situated as a partial fix among the greater solution.

2.1 State of U.S. housing

In the United States there are 44 million renters, and of these more than 47% are rent burdened (Kontokosta, Reina, and Bonczak 2020). The term 'rent burdened' is defined by the United States Department of Housing and Urban Development (HUD) as a household that spends more than 30% of their gross annual income on rent. The US is also one of few countries in the developed world which does not have any successful form of public housing in effect due to traditionally neoliberal housing policy. Countries such as Austria boast more affordable, and higher-quality public housing offerings than the average rental in the United States, with the public costs coming primarily from taxes upon large corporations (Forrest 2019).

2.1.1 Exclusionary tactics

The behaviors that govern this issue are systemic, but also unconscious for the most part on the side of the homeowners. This same group makes up the bulk of those attending municipal meetings regarding land use and urban expansion to fight for their own interests. This is not a desire to price out others from purchasing their first home, but rather the

desire to increase their home valuation. Historically however, some have actively used this exclusionary strategy to deter black and brown people from integrating into suburbs for example. On the other hand, there are housing activists fighting for a more just and affordable reality for city-dwellers. Driven primarily by the philosophy of new urbanism, this discourse primarily relates to ideas such as rent control, public housing options, expanding multifamily residential zoning, and the creation of more green spaces (Brinig and Garnett 2013).

Redlining, or the increase in interest rates due to a perceived 'risk' to issue mortgages in varying neighborhoods based on ethnic makeup or crime rates has legally been eliminated, but other issues continue to plague the US housing market. The twin concepts of filtering and gentrification, while not systemic, are systems that impact housing affordability. Filtering can be described as the mechanism in which initial wealthy occupants move to newer, more expensive housing as their original dwellings age, allowing those with lower incomes to move in (Hertz 2015b). Gentrification, on the other hand, has the opposite effect. Boutiques, cafes, and other non-essential shops opening in historic, filtered districts spike the housing prices back up or above the original prices, considering inflation. Many of these older dwellings are retrofitted into luxury apartments far out of reach of the existing residents. In turn, this causes many of those who have lived there for decades to be displaced to nearby neighborhoods once they can no longer afford the increased rental prices.

2.1.2 Contemporary dilemmas

While affordable housing is subsidized through various methods, either publicly or privately, homeowners are also subsidized in ways not typically discussed. Homeowners are given mortgage interest tax deductions, leading towards mortgages often being more affordable than renting. However, many current renters are unable to secure mortgage loans due to credit history or down payments being out of reach. In addition to this, in 2021 a new force adding to the impossibility of purchasing a home for millennials and generation Z. Private equity firms such as BlackRock are purchasing massive amounts of starter homes to convert to rentals. In just the first quarter of this year, 15% of all home sales were to private equity like BlackRock, some paying upwards of 20% - 50% over asking price in cash in an effort to outcompete prospective buyers, while paying nearly anything in interest (Botella 2021).

2.2 Impact of energy efficiency on affordability

Extending the complexity of the situation, the architecture, engineering, and construction (AEC) industries account for one third of all carbon emissions. Therefore, housing availability goals must also align with carbon goals (Zhong et al. 2019). Carbon emissions are discussed in two distinct varieties: embodied carbon and operational carbon. Embodied carbon can be defined as carbon emissions stemming from construction of a building (this includes manufacturing and transport of materials). In contrast, operational carbon refers to emissions which occur during the use, management, and maintenance of buildings. In most cases, the operational carbon is directly tied to the cleanliness of the local energy grid. Carbon equivalence is a scalar value used to calculate the mass of carbon emissions per unit of energy used, given most commonly in units of $kgCO_2$. While this grid carbon equivalence varies across the United States, the average is 0.386 $kgCO_2$ (United States Environmental Protection Agency (EPA) 2022). Due to a general reduction in carbon across the energy infrastructure throughout the United States, it is important for architects to begin to focus more closely on embodied carbon. However, this research focuses solely on the operational side of the equation and its related impacts on both owners and occupants.

2.2.1 Energy cost burden

Operational carbon emissions are not only a function of grid carbon equivalence, but also energy consumption. With DADUs proposed as part of the solution to housing equity, it is important to further understand the complications that energy costs have on affordability. Households with the lowest income are disproportionately affected by excessive energy spending. This is exacerbated by the inability to make investments into energy efficiency (Kontokosta, Reina, and Bonczak 2020). In extension of the definition of rent burdened, it can also be helpful to analyze the energy cost burden (ECB) of a city or neighborhood. Defined as the total household energy cost divided by median household income by neighborhood, results from a 2020 study found that those who earn less than 80% of the median area income have a median ECB of 7.2%, with some upwards of 25% (Kontokosta, Reina, and Bonczak 2020). Latino and African American families are inordinately impacted by high energy cost burdens. The above statistic can be explained as another effect of filtering as

older units tend to be much less energy efficient than new construction. Additionally, apartments rarely allow occupants to make these modifications in the first place. Kontokoska et al. discovered that in Seattle, the energy cost in dollars per square feet was around 1.0 across all income bands. Seattle's energy grid is owned and operated by Seattle City Light, a public utility company. The first electric utility to become net zero in 2005, it is also the most affordable at $0.12 per kilowatt hour. Most of the grid is supplied by hydroelectric power, with the emissions of the remaining sources made up through purchased carbon offsets. As such, Seattle energy costs tend to lead to a lower ECB on renters; however, any reduction is beneficial.

2.2.2 High performance systems

In terms of accessory dwelling units, both occupants and owners have a stake in reducing energy consumption and costs. Lowering utility costs makes an ADU unit more attractive or even simply realistic to potential renters. Ideally, ADU owners may even pass on potential energy cost savings onto residents. In response to the comparatively affordable energy costs in Seattle, this plays a smaller role in this context. However, it is essential to examine as the methodology of this research can be applied anywhere ADUs are able to be constructed. For users, both homeowners and designers alike, the ability to quickly visualize anticipated energy expenditures additionally remains valuable. Performance tradeoffs exist between more expensive, up front investments and cheaper, less efficient alternatives. Real-time comparisons between low- and high-performance design options would expectantly influence owners to opt for the higher performance solution.

Generally, accessory dwelling units as a typology intrinsically exhibit high surface-area-to-volume ratios (SVR) on account of compactness. As a greater surface area in relation to a constant volume equates to a greater heat loss through the building facade, ADUs are at a disadvantage when compared to single family dwellings. Overall however, less energy will be consumed as a result of having less floor area. Due to their size, many ADU heating, ventilation, and air conditioning (HVAC) designs opt for ductless heat pumps in place of traditional duct systems. This choice reduces energy consumption due to heating, and is especially effective in a temperate marine climate such as Seattle, Washington. While data on average annual energy savings from this system in ADUs could not be located, a study on

multifamily housing found the value to be about 350 kWh/yr per unit (Logsdon and Larson 2016). This assessment investigated two newly developed multifamily residential buildings consisting of stacked studio and one bedroom apartments. Usage of high-efficiency systems such as heat pumps and methodologies such as the one described within this research can help ensure that these advances in housing availability are not regressive in nature towards climate goals.

2.3 Accessory dwelling units as a solution

To reiterate: accessory dwelling units are not the end-all be-all solution to the housing affordability crisis, but remain a part of a suite of strategies addressing the crisis. Accessory dwellings offer many potential changes for the better by increasing housing density with lesser pushback. Affordable, inclusive housing, independence for the elderly, and supplemental income for owners are all enabled by the increased construction of ADUs (Brinig and Garnett 2013). Additionally, an increase in ADU occupancy leads to a reduction in individual carbon footprint compared to a typical house, due to a reduction in per capita energy consumption and physical footprint.

ADU construction has steadily increased as policy was updated and more homeowners were made aware of the possibility. According to the most recent data from the City of Seattle, the year-over-year increase in production of ADUs (attached and detached total) between 2019 and 2020 was 71.83%. This growth continued with the 64.96% change in production between 2020 and 2021. Data from Seattle is only current through Q2 of 2022, but extrapolating the 309 units accounted for, it is safe to assume similar growth of around 1236 new units this year (53.54% estimated annual increase). Seattle will welcome a predicted additional 1.8 million residents by 2050. Based upon a report from the Puget Sound Regional Council (PSRC), this population influx will demand 830,000 new dwelling units, using their prediction of an average household size of 2.36 persons by 2050 (Fesler 2018). Utilizing ADUs alone will not meet this demand at this rate; therefore, they must be situated as only one part of the solution. In Portland, Oregon for example, ADUs make up only around 0.7% of all housing units throughout the city (Gebhardt, Gilden, and Kidron 2018).

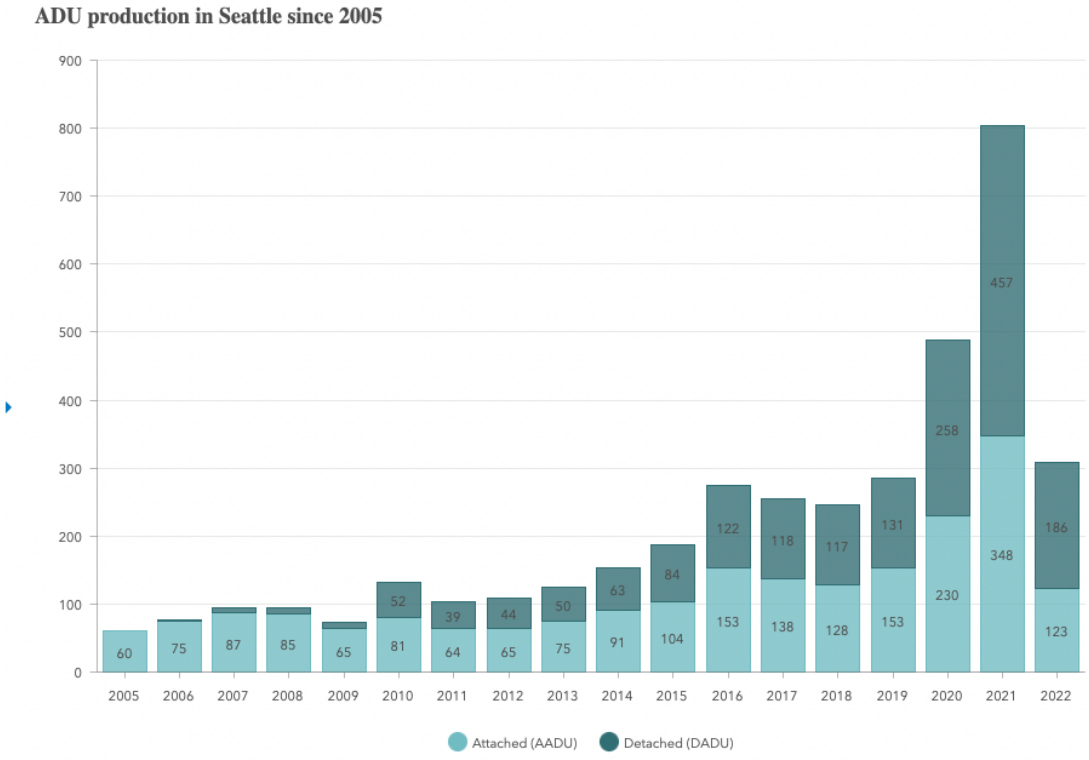


Figure 2.1: Seattle ADU production since 2005 (City of Seattle 2021)

2.3.1 Affordability and inclusivity

Of the major benefits attributed to accessory dwelling units, this research focuses primarily on the climate and social resilience aspects. ADUs enable affordable inclusion of individuals and families formerly restricted by redlining, systemic racism, and exorbitant housing costs. Many ADUs are rented out at prices not only desirable but affordable to the aforementioned groups. Out of 515 responses to a survey out of Portland, Oregon, 44% of ADU owners report renting their unit(s) out below market value (Gebhardt, Gilden, and Kidron 2018). Minority and working class groups are then advantaged to move into neighborhoods which were once considered out of reach (Brinig and Garnett 2013). As a result, families taking this initiative can gain access to schools with greater funding. This is due to funding being tied to property taxes in the United States, meaning lower-income neighborhoods tend to have less adequate public school funding.

Cities across the United States including Seattle have begun aiming for what is referred to as the 15 minute city- wherein all city dwellers have all of their essential needs met within a 15 minute walk from their dwelling (Bicknell Argerious 2020). Another result of ADU adoption is an increased density in single family zoning. This should help to support the lofty goal of becoming a 15 minute city and aid in removing the collective dependency on automobiles. Other benefits of this agenda include reduced urban sprawl and food deserts, neighborhoods without access to fresh food and groceries.

2.3.2 Aging in place

Additionally, accessory dwellings enable older citizens to remain independent by forgoing a nursing home to live on family property. While Seattle data is yet to be collected, over 35% of Portland survey respondents said the original intended purpose of their ADU was as a residence for a family member (Gebhardt, Gilden, and Kidron 2018). This arrangement is referred to as aging in place and is increasingly popular among the aging American population. According to the AARP, the majority of older adults reported they would prefer to age in place, given the chance. (Cobb and Dvorak 2000). Additionally, the adult children of age in place elderly see this system as mutually beneficial. Being able to stay near to their parents, as well as the possibility of the elderly parents to contribute to childcare are

seen as perks. AARP actively advocates for removing certain legal restrictions on new ADU construction and has even published sample ADU legislation (Brinig and Garnett 2013). The caveat being that around 50% of AARP members are wealthy and likely oppose progressive land use policy and/or zoning updates.

2.3.3 Owner benefits

In order for ADUs to be constructed in the first place, the owner must be able to gain benefits significant enough to warrant the financial and time investments necessary. Construction of accessory dwelling units are primarily paid for in cash or through a home equity line of credit by owners (Gebhardt, Gilden, and Kidron 2018; Chapple et al. 2017). Owners in either of these positions have very little to lose and recognize the long-term value generated through supplementary rental income. While this research posits that removing the necessity to own land from individual financial success in the United States, working within this existing framework to bring value to wealthy property owners supports essential housing goals. In order to address allowing less-advantaged homeowners to invest in ADUs, Seattle is exploring an accessible loan program for ADU/DADU construction costs in order to increase equity in this space (Levy and Pennucci 2019). Due to the flexibility inherent to accessory dwellings, owners can opt to switch to using the structure for office space or aging in place housing as they see fit.

2.3.4 Seattle ADU code

Previously mentioned in section 1.2, Seattle has allowed attached ADU construction in single family zoning since 1994 and detached ADU construction since 2010 (Levy and Pennucci 2019). Due to the ADUs existing in single family zones, promotion of this housing strategy does not require a full revision of land use policy, only minor changes around the edges. Updated Seattle ADU policy in July 2019 from then-mayor Jenny Durkin brought about many such changes (Welch et al. 2021). Amongst these, the off-street parking and owner-occupancy requirements were waived. Additionally, viable lots can now contain 2 ADUs, with the only stipulation being that only one may be detached. Minimum lot size was further reduced to 3200 square feet. Other notable rules surrounding ADU code include a maximum (combined, in the case of 2 units) gross floor area of 1000 square feet, a maximum rear yard

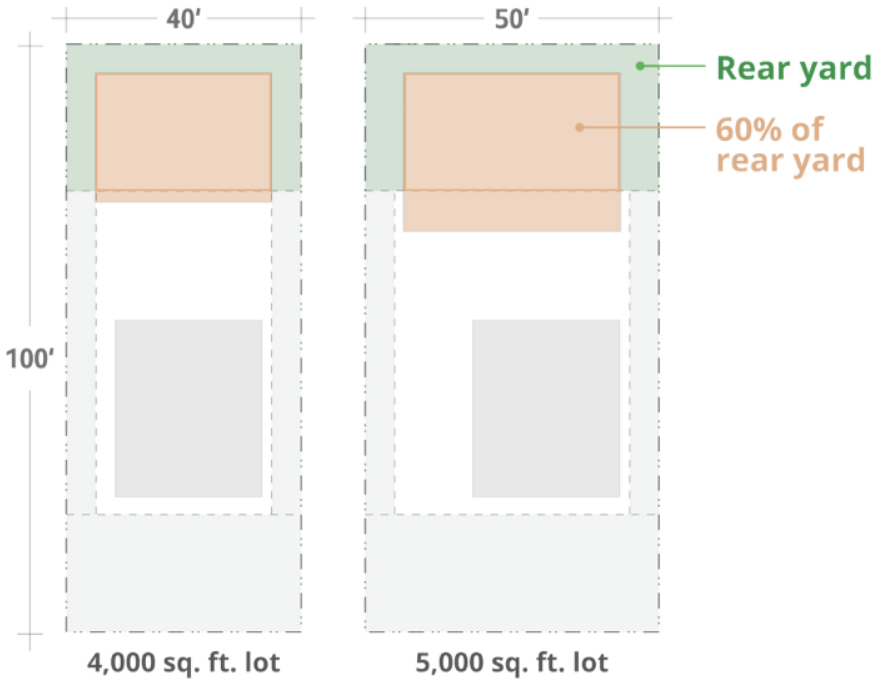


Figure 2.2: ADU lot coverage diagram (City of Seattle 2019a)

coverage of 60% (see Figure 2.2), and units must follow standard property setbacks. Varying height limitations can be seen in Figure 2.3.

2.3.5 ADU code comparison

While Seattle claims to champion the most progressive ADU policy of any municipality in the United States, other states and cities have also made large strides (Quirindongo 2021). Washington state has mandated local governments incorporate provisions allowing accessory dwelling units, though with large flexibility. Similarly, California implemented mandatory ADU-inclusive land regulations in 2002; however, many municipalities delayed or refused to allow ADU construction (Brinig and Garnett 2013). In contrast, New York City, San Francisco, and Chicago have entirely resisted pushes from housing advocates and forbid ADU development entirely. Portland, Oregon and Vancouver, Canada have found success as well, in the face of similarly escalating housing prices as Seattle. Portland abolished the design-review process for ADUs, and has also waived system development charges (SDC)-initial fees charged for sewers, water, etc. (Chapple et al. 2017). SDC charges can oftentimes

Maximum DADU height (in feet)	Lot width (in feet)			
	Less than 30	30 up to 40	40 up to 50	50 or greater
Ⓐ Base structure height limit	14	16	18	18
Ⓑ Additional height allowed for pitched roof	3	7	5	7
Ⓒ Additional height allowed for shed or butterfly roof	3	4	4	4

Figure 2.3: ADU height limitations (City of Seattle 2019a)

account for as much as 30% of the total costs of the construction and permitting process. Vancouver on the other hand has the advantage of deep lot proportions along alleys, giving ADUs near-guaranteed direct access. Furthermore, loan support is prevalent leading to a significant backlog in permitting requests citywide. Overall, ADU policy across North America has been trending in the right direction and picking up speed in recent years in response to the housing crisis. Making sure that this push for ADUs has a focus on carbon and energy reductions is vital. Creating building energy models to simulate ADU energy use is fairly straightforward but can be difficult for laypeople such as homeowners, contractors, and developers to understand. Therefore, crafting an easy-to-use, swift alternative with comparable accuracy was explored.

2.4 Building energy modeling

Building energy modeling describes a large family of techniques used to simulate, predict, and visualize energy consumption of buildings. Methods include statistical, data-driven, and deterministic routes. The use case of energy modeling within the web application targets planned building in early design stages, but it is also used to study metered, post-occupancy energy use compared to anticipated energy use (referred to as measurement and verification) (Granderson et al. 2016). Data-driven energy modeling is also necessary in retrofit projects, as getting existing building stock up to spec is of utmost importance to reach climate goals (Amasyali and El-Gohary 2018). Traditionally, energy modeling was conducted by simulating real world scenarios using thermodynamics equations and physics-based models. However, data collection methods such as advanced metering infrastructure (AMI) are rising in prevalence, allowing researchers to build complex statistical models to predict energy consumption in commercial buildings (Touzani, Granderson, and Fernandes 2018; Amasyali and El-Gohary 2018). Architecture historically has not been associated with 'big data' such as this, but as BIM, parametric modeling, and technology such as AMI increase in popularity, the industry has made progress.

2.4.1 EnergyPlus

One common traditional simulation framework is EnergyPlus, developed by the United States Department of Energy (DOE) in collaboration with the US National Renewable Energy

Laboratory (NREL). EnergyPlus is capable of illuminance and glare calculations, thermal comfort models, and simulation of thermal zones using conduction transfer methods (US Department of Energy 2021a). The EnergyPlus framework is free and open-source and cross platform, with major updates occurring biannually (US Department of Energy 2021b). EnergyPlus operates without a graphical user interface (GUI) via command line by default and performs calculations using input data file format (.IDF) ASCII text files (Big Ladder 2014). OpenStudio is a frequent free and open-source addition to EnergyPlus, extending a GUI implementation for ease of use. Developed by NREL, OpenStudio modifies EnergyPlus simulations to function as object oriented data models (Casini 2022). This organization of energy model criteria simplifies the interfacing process of EnergyPlus with 3D modeling software such as Rhinoceros.

Construction of an EnergyPlus model requires a minimum of two files: an .IDF file and an .EPW file (EnergyPlus Weather File). IDF files contain data that represents building geometry and systems, while the EPW file represents a typical meteorological year (TMY), meaning average annual climate information representative of a 30 year reporting period (US Department of Energy 2021c). EnergyPlus weather data is supplied to the model by pointing to the .EPW filepath, however the .IDF file is typically compiled from many variables using OpenStudio's object oriented data format. Required simulation parameters include: time period, thermal zones, shading geometry, HVAC and mechanical system(s), building loads, schedules, etc. Optionally, users can change the timestep (number of calculations per hour), heat balance algorithm, or surface convection algorithms. Each of these parameters is made up of a larger subset of variables. For example, the time period parameter includes start and stop dates, as well as holiday and/or daylight savings time exclusions.

Thermal zones are the core building blocks of EnergyPlus building energy models and are a collection of the surfaces (walls, windows, roofs, etc.) and related information such as ceiling height, floor area, and volume (US Department of Energy 2021a). Each surface has a specific construction applied, consisting of a list of materials ordered from exterior to interior. Construction sets can contain constructions for walls, floors, and windows simultaneously, which when applied to a surface selects the appropriate construction. See Figure 2.4 for greater detail in how these aspects interrelate. EnergyPlus represents three-dimensional geometry as

sets of vertices belonging to zone, surface, or other boundaries. Surfaces separating interior zones from the exterior are assigned a *roof* designation. Floor surfaces separating the interior zone from the outside are either assigned an *exterior floor* or *ground floor* classification. This selection is determined based on boundary condition specified (i.e., slab on grade or crawl space condition).

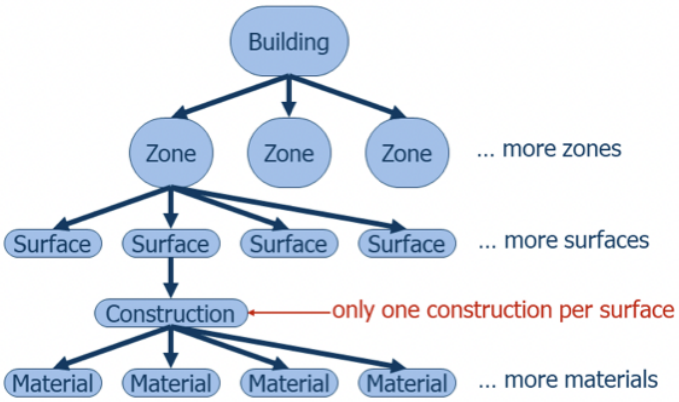


Figure 2.4: EnergyPlus envelope component heirarchy (US Department of Energy 2021a)

Shading geometry is handled by describing each object as either attached to or independent of a zone. Attached or 'Zone' shading typically is used for solar fins, louvers, or overhangs, while the independent or 'Site' shading describes nearby buildings, foliage, or other obstructions. These shading objects primarily impact solar gain by shading windows, in turn affecting energy consumption from electric lighting usage (US Department of Energy 2021a). As HVAC systems are inherently complex, EnergyPlus ships with many templates to represent varying HVAC configurations commonly used. Advanced users can mix and match heating sources, cooling sources, and ventilation methods to more accurately depict the system in question. Schedule objects on the other hand represent everything from occupancy loads to equipment operation schedules to thermostat set points (US Department of Energy 2021c). Schedule parameters can specify hourly occupation or hourly modality of building systems independently for each object, meaning there is a great deal of flexibility. However, for tools adjacent to EnergyPlus such as Honeybee, schedules, loads, and programs can be applied via templates to simplify the process. This notion continues similarly for other aspects of construction of EnergyPlus building energy models when using OpenStudio and Honeybee.

2.4.2 Ladybug Tools

Created for use within Rhinoceros and Grasshopper modeling software, Ladybug tools is a suite of plugins related to quantitative and qualitative analysis of energy, daylighting, climate, and site (Sadeghipour Roudsari and Pak 2013). Plugins include Ladybug (site/climate analysis), Honeybee (EnergPlus-based energy modeling), Dragonfly (urban-scale analysis), and Butterfly (computational fluid dynamics simulations). Of primary interest to this project is Honeybee and its usage of both EnergyPlus and OpenStudio. Ladybug tools was founded by Mostapha Sadeghipour Roudsari and Chris Mackey and is actively developed by a larger team, including the original founders as free and open-source software.

Following in the footsteps of OpenStudio making EnergyPlus more user-friendly by introducing a GUI, Honeybee allows users to create and perform EnergyPlus simulations in the Grasshopper visual scripting environment within Rhinoceros. Beyond the plug and play user interface, Honeybee further holds the advantage of default values for advanced simulation criteria many intermediate users are unfamiliar with, such as setting most building schedules from a building program template. These program template options allow for selection of building vintage and typology to better fit the representation. Honeybee aids in creation of the .IDF files and allows users to run the simulation through an intuitive GUI. As a visual tool, users typically begin by defining thermal zones in Rhinoceros as an approximation of the building form. Thermal zone boundaries are configured from this solid before applying windows, doors, constructions, shading, and schedules to the model. Completed, the Honeybee model is converted to an OpenStudio energy model object before being evaluated by EnergyPlus at the core.

2.5 Metamodeling

Traditionally, the evaluative approach utilizing EnergyPlus or similar has been used to receive quantitative results regarding building energy consumption, but limitations exist. This method is both slow and requires relatively expert knowledge to set up and evaluate. Statistical methods have begun to rise in prevalence within the field due to advances in computation speed, most notably through a process called surrogate modeling. Also known as metamodeling, this engineering methodology involves creating an approximation of a simulation model.

The reasoning behind this is quite simple: repetitive simulation runs are costly from a time and computing perspective, and being able to predict the outcome saves both. Additionally, real-time analysis allows for feedback in early stages of design instead of solely as a final confirmation method (Hanna 2020). This process is an example of supervised machine learning (more on this in Section 2.5.1, given that surrogate models essentially find patterns between inputs and their corresponding outputs (Guo 2020). Also known as learning algorithms, machine learning involves creating statistical models for predicting or making decisions based on patterns discovered in training data.

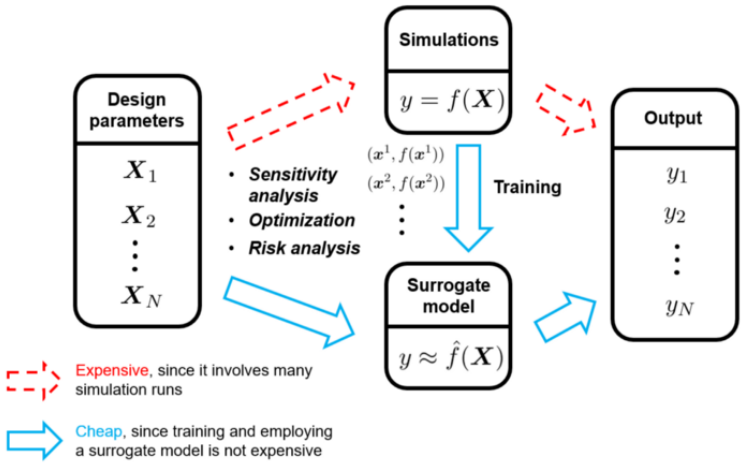


Figure 2.5: Surrogate modeling process (Guo 2020)

Statistical models of this type require large swathes of data in order to accurately predict combinations within. Dataset requirements can be thought of as the upfront cost to creating a metamodel; however, the payoff is the ease of use and relative speed through continued use. Generally, with greater amounts of data to use as the training set comes greater accuracy in predictions. One can think of linear regression as a simple comparison- intricate patterns may exist within gaps of data points. As more data is added to fill in the voids, these previous flawed predictions become more obvious. Another comparable method many computational designers are already familiar with is optimization (Hanna 2020). Multi-objective optimization specifically, is the process in which the most advantageous combination of variables are found. This optimal solution is typically algorithmically located through either the minimization or maximization of an objective function defined by the user to describe design goals. The main difference lies in optimization algorithms being built specifically to

target one design problem, whereas machine learning looks to minimize error over a generalized design space, and predict accurately on new data (James et al. 2013). Usefulness of a surrogate model is dependent on ease of use, however. Apps made without non-academic users in mind fall short due to user frustration (Mueller 2014). In addition, the threshold of accuracy for a successful metamodel is subjective- quick feedback requires less precision, for example.

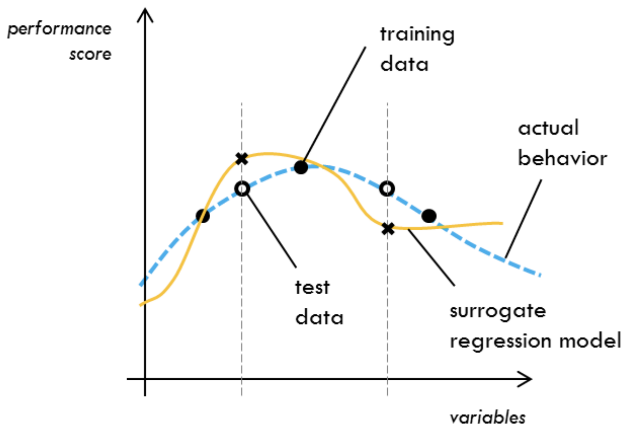


Figure 2.6: Conceptual illustration of surrogate modeling (Mueller 2014)

2.5.1 Machine learning

Machine learning (ML) models are some of the most common techniques used to craft metamodels. The field has varied in terms of popularity over the decades and has existed as a concept since at least the 1950s, when the term was coined by Arthur Samuel of IBM in a study using the game of checkers (Samuel 1959). The process of feeding data to a learning algorithm is referred to as model training. While a massive variety of ML algorithms exist, training generally consists of pattern recognition across variables, or 'features'. Most of these algorithms which are relevant to the field of design are categorized as supervised learning, meaning that the output corresponding to each set of inputs is known to the algorithm. This enables an accurate measure of error to be determined by comparing discrepancies in predicted and actual outputs. Commonly referred to as the 'loss' of the model, and can be calculated many different ways along with different pros and cons. Many ML algorithms are deterministic and will arrive at the same loss, given the same features from the same training

set, and the same hyperparameter values (Sra, Nowozin, and Wright 2012). Hyperparameters are parameters whose values are used to control aspects of the learning process, such as training speed, number of epochs (in neural networks), number of branches (in decision trees), etc. These values are not reflective of the underlying dataset that is being analyzed. Finding the optimal solution of hyperparameters is imperative to ensure model validity and is termed model tuning.

To briefly highlight the machine learning methods relevant to this research, regression models and decision trees will be discussed. Regression can be defined as a predictive modeling technique which determines the relationship between a dependent and independent variable (Pant 2019). The most common category is polynomial regression, as most real world problems are not as simple as some variant of $y = mx + b$. However, pure regression models are prone to overfitting data; that is, when a model overestimates patterns between training data and is unable to generalize predictions towards new data. Another significant branch of machine learning is decision trees. At its simplest, decision trees function by splitting input parameters into logical regions using if-then rulesets (Amasyali and El-Gohary 2018). Each splitting rule is fitted to a constant representing the average output of the region, in the case of regression. Decision trees can also be used to perform regression analysis; however, they tend to be less accurate than other regression techniques. Advances in tree processes have exploited this weakness and turned it into a strength (more on this in Section 2.5.4). Even ignoring this flaw, this type of ML model can be classified as an 'open box model', meaning each decision point in a tree model can be graphically understood (Rokach and Maimon 2014). This simple model interpretation is in sharp contrast to the bulk of machine learning techniques where the decisions made by the model are obfuscated and unable to be directly visualized (James et al. 2013).

2.5.2 Error measures

Error measures are used in machine learning to inform the data scientist as well as the algorithm an estimation of accuracy. Known as loss, it can be defined as the measure or function of how far predicted values are from the true values across the model, with penalties applied for bad predictions. These penalties are of varying degrees depending on the error measure implemented. While options are plentiful, mean absolute error (MAE) (Equation

2.1), root mean error MSE (Equation 2.2), and root mean square error (RMSE) (Equation 2.3) are among the most prevalent across machine learning. For example, MAE finds the sum of the absolute values of the difference between each set of predicted and true values, before normalizing by the total number of data points. In the case of metamodeling, it is important to note that these true values are reflected as simulated values.

Loss functions are primarily picked based on which learning algorithm is selected, but other times are selected simply for their ability to construe model accuracy. Beginning with MAE and RMSE, the error values are relatively straightforward to interpret, having the same units as the dependent variable. In the case of energy consumption prediction, a hypothetical model with MAE or RMSE of 0.23 would mean that on average predictions are off by ± 0.23 kBTU/ft^2. For MSE on the other hand, the formula for MSE creates a value with squared units leading to a more esoteric measure of error. MAE and RMSE are accordingly selected much more frequently than MSE (Chugh 2020). As MAE is non-differentiable, which significantly hampers potential calculations, RMSE is commonly the de facto loss function utilized.

$$\mathrm{MAE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{D}} |\mathrm{y_i} - \hat{\mathrm{y}}| \tag{2.1}$$

$$\mathrm{MSE} = \frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{D}} (\mathrm{y_i} - \hat{\mathrm{y}})^2 \tag{2.2}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{\mathrm{n}} \sum_{\mathrm{i}=1}^{\mathrm{D}} (\mathrm{y_i} - \hat{\mathrm{y}})^2} \tag{2.3}$$

Learning algorithms may calculate only one measure of loss, or many, depending on the algorithm selected and dataset in question. One loss function can be used to show error to the data scientist, while the algorithm interprets a different type of error. Furthermore, novel loss functions can be created as needed when either the dataset is precarious or the practitioner chooses to alter how error is internalized. For example, in a paper detailing the surrogate modeling process of structural design, 4 innovative error measures were introduced (Mueller 2014). Mueller used these indicators solely for validation after training the model,

as they are non-differentiable. Yet, all loss measures outperformed traditional RMSE, MSE, and MAE in this validation stage. The issue is that while such novel loss functions may yield great results in one situation in particular, without being implemented as a Python library they will rarely find use.

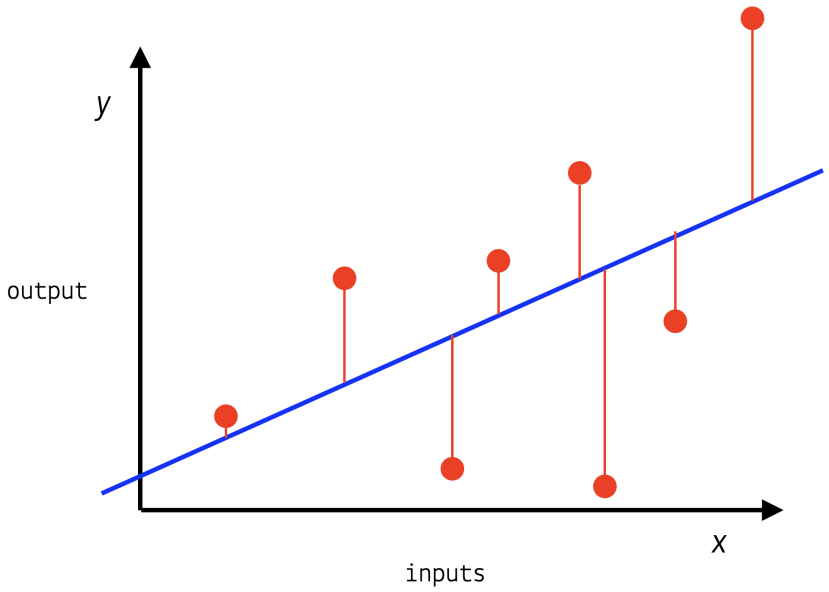


Figure 2.7: Loss computation

In order to reduce model error, a straightforward method of feature selection or feature engineering is often employed. Feature selection is the process of withholding features (independent variables) from the training process in different combinations (Schreck 2018). This can be accomplished either by trial and error or more commonly through the use of statistical methods. To find the optimal set of features guaranteed, a grid search process can even be conducted, evaluating every possible combination set. Conversely, feature engineering is often used where training data lacks substantial dimensionality (Schreck 2018). Feature engineering consists of the synthesis of new features, normally through statistical inference of existing features. Through the use of these methods and similar, data scientists can locate the most accurate model available. Again, loss is useful not only to the data scientist or practitioner, but also as a metric for the model itself to optimize during the training process. While it was mentioned that machine learning as a concept is indeed distinct from optimization, the way to find a model with lowest loss is through the specific process of

hyperparameter optimization (Sra, Nowozin, and Wright 2012).

2.5.3 Hyperparameters and optimization

As explained in Section 2.5.1, hyperparameters are model parameters outside of the dataset which influence model training speed and loss. Some hyperparameters can often be difficult to ascertain and even more challenging to infer a starting point. Instead, optimization or search algorithms are applied to quickly and/or conclusively determine the optimal set of hyperparameters. When utilizing such algorithms, hyperparameters are often specified as a range of values with discrete steps. In combination each hyperparameter and associated range can be tested independently for model loss. Methods such as grid search, random search, and Bayesian search offer tradeoffs between speed and efficiency. Grid search simply iterates through all possible permutations of hyperparameters and returns the set with lowest error. As the name implies, random search progresses through n random samples of hyperparameter combinations before similarly returning the parameters of the lowest error model. In contrast, Bayesian search is actually more akin to optimization methods in that it operates using an objective function. However, unlike traditional optimization algorithms, in Bayesian search, a probabilistic model of the objective function is created and evaluated in place of the unknown objective function (Koehrsen 2018). Essentially, this stand-in model is a surrogate model allowing for cheaper computation and cycles until reaching an imposed time or iteration limit. The use of hyperparameter optimization in tandem with feature selection theoretically leads to the most accurate possible model.

2.5.4 Gradient boosting machines

Focusing in on surrogate modeling techniques to predict building energy, a competition known as the ASHRAE Great Energy Predictor III was held in 2019, challenging teams to predict energy consumption as accurately as possible. Previous incarnations of this competition were held in 1993 and 1994, with winning teams employing *artificial neural networks* (ANN) in each (Miller, Hao, and Fu 2022). In this latest competition however, *gradient boosting machines* (GBM) were the highest performing, according to a post-event survey administered to the top 5% of teams. Nearly 75% of these top performing teams selected a GBM Python library, with the remainder using ANN models. Gradient boosting machines

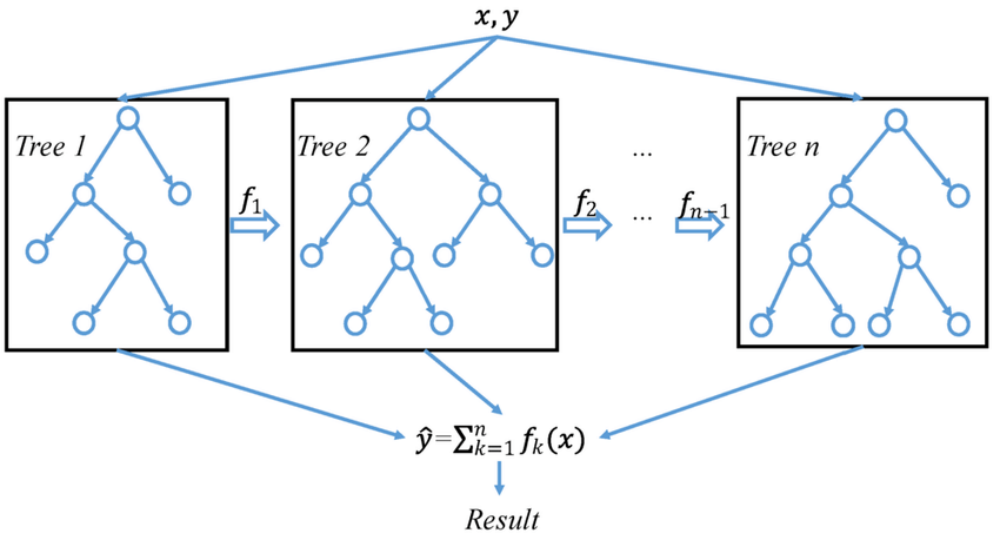


Figure 2.8: Simple illustration of XGBoost process (Wang et al. 2019)

are also sometimes referred to as gradient boosting decision trees, as they are many decision trees in tandem. This is referred to as an ensemble model, and is the adjustment made to reconcile the relatively low accuracy of each individual decision tree (typically referred to as 'weak learners') (Grillone et al. 2020). When combining many trees in this process, the prediction is made by all decision trees voting in mass.

GBMs boast high prediction accuracy due to their ensemble nature, but require hyperparameter optimization and cross validation to avoid overfitting. Overfitting is when a model is focused too tightly on training data, leaving it prone to high error when shown new data outside of its existing range. Countering this trend to overfit, GBMs also make use of two ensemble methods: bagging and boosting. Bagging is also known as bootstrap aggregation and consists of random sampling of the training data, with replacement. This helps to ensure that trees are trained via a representatively wide range of training dataset. In combination, boosting is a method of iteratively training the next sequential tree by using the residual error of the previous (D'Souza 2018).

Chapter 3

METHODOLOGY

Training a machine learning model requires a vast dataset in order to achieve accurate results across all conditions. Datasets are typically organized as dataframes, a programming abstraction representing a spreadsheet. Within these, each case (simulation) is a row and the features (design variables, outputs) make up each column. This arrangement is necessary in order to effectively deduce patterns and trends within the data. However, architecture as a discipline in its current form does not organize its data in modes useful for data science. No datasets exist with the Seattle-specific DADU energy information required for this prediction of DADU energy and carbon impacts, nor any such datasets with the depth of examples needed for valid predictions. Therefore, in order to proceed, a simulated dataset was created. EnergyPlus was selected for the simulation process both due to integrations found in Honeybee and Rhinoceros as well as text-based file handling schema. This methodology is organized in sequential order and will be briefly explained next. Section 3.1 describes the simulation process and variable selection, while Section 3.2 details data preparation. Following this, Section 3.3 describes the method of training the machine learning model, with Section 3.4 diving into the development process of the web application. Lastly, Section 3.5 details the methods used to perform a comparative analysis between the web application and existing DADU designs.

3.1 Dataset simulation

Using Rhinoceros and Grasshopper, a 3D modeling application and visual scripting plugin for Rhinoceros, respectively, were used to model the design space. Honeybee, a plugin for Grasshopper for daylighting and energy modeling was added to the workflow to carry out simulations of each unique design. Honeybee utilizes EnergyPlus, a validated building energy simulation software developed by the DOE, to perform the energy modeling. The pipeline of

these three software tools was selected as a result of the lack of an existing dataset reflective of DADUs located in Seattle, with the variables of interest. In addition, the author was most familiar with this route from previous work. Results and their affiliated inputs were captured and written to .CSV files, allowing for a lightweight solution of data storage, as well as being the most effective way to interface with the data science tools used later on.

3.1.1 Simulation variables

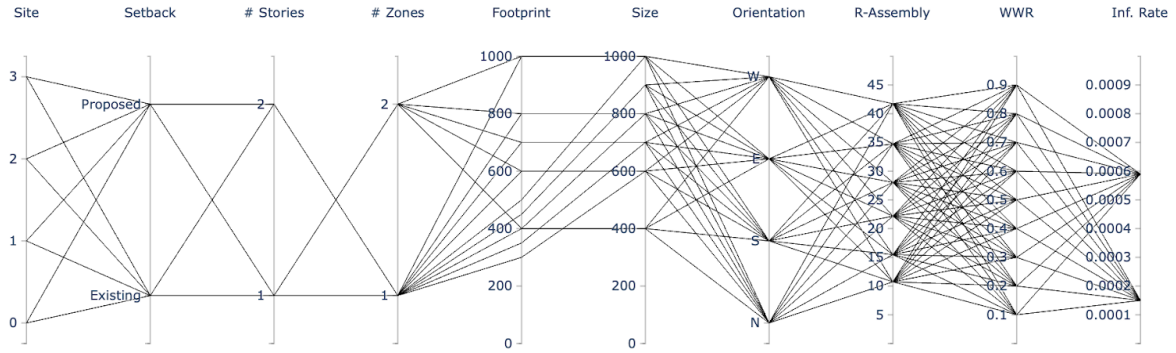


Figure 3.1: Parallel coordinate plot of design space

Design space is a term that refers to the limits of all designs possible within the given constraints and can be represented geometrically. Variables chosen to encompass the simulated DADU design space are: site typology, number of units and stories, floor area, infiltration rate, orientation, window-to-wall ratio (WWR), wall assembly, and setbacks. Reasoning and criteria behind decisions of which variables to consider relevant are covered in detail in Section 3.1.2. Additionally, variables such as height or the footprint of the structure were parametrically defined and will be explained later in Section 3.1.3. The design space was visualized using an interactive parallel coordinate chart generated using Plotly, a data visualization library for Python, among other languages (see Figure 3.1.1). Additionally, refer to Table 3.1 for a detailed breakdown of values.

Feature [# options]	Values
Site [4]	corner-alley, corner-no alley, infill-alley, infill-no alley
Typology [3]	1-story 1-zone, 1-story 2-zone, 2-story, 2-zone
Size (ft^2) [6]	400, 600, 700, 800, 900, 1000
Infiltration rate ($\frac{m^3}{sec}$)* [2]	0.00059, 0.00015
WWR [9]	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
R-value ($\frac{ft^2 \cdot °F \cdot hr}{BTU}$) [6]	10.7, 15.5, 22.2, 28.1, 34.7, 41.7
Setbacks [2]	existing, proposed
Orientation [4]	north, south, east, west

Table 3.1: Input design space (*per m^2 of building facade)

3.1.2 Design space generation

A goal for this tool was to effectively simulate the variety of single-family residential (SFR) plots, and we achieved this by creating two variables: lot type and lot access. Lot type differentiates between whether a lot is situated between two neighboring lots (infill lot) and a lot located at an intersection (corner lot). Access refers to whether the rear of the site abuts an alleyway or another property directly. Together, these two variables create the four distinct combinations used in this study: corner-alley, corner-no alley, infill-alley, and infill-no alley. While no comprehensive analysis was conducted during the site selection, advice from advisors and a desire to search for lots from varying neighborhoods were the primary drivers. Simple selection criteria included apparent averages in tree cover, yard size, and existing dwelling size. All neighborhoods in which these sites are located are neighborhood residential (formerly single-family residential) zoning and located within Seattle, Washington. Each site is eligible for construction of a DADU under current Seattle land use policy. In order to utilize these sites in the simulations, lots were 3D modeled using Rhinoceros from 2D site data via City of Seattle GIS. Surrounding structures within 100 feet were included and factored into shading calculations, including the existing primary dwelling on each lot. Initial testing included modeled trees in the simulations yet resulted in a negligible difference in energy performance and greatly increased simulation time; therefore, they were not included.

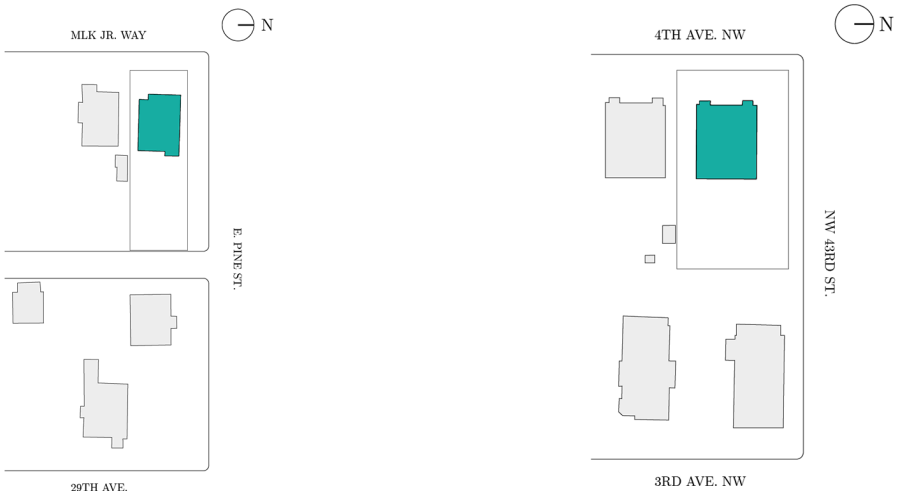


Figure 3.2: Corner lot plans (left: alley (Madrona), right: no alley (Fremont)

Figure 3.3: Infill lot plans (left: alley (Meridian), right: no alley (Central District)

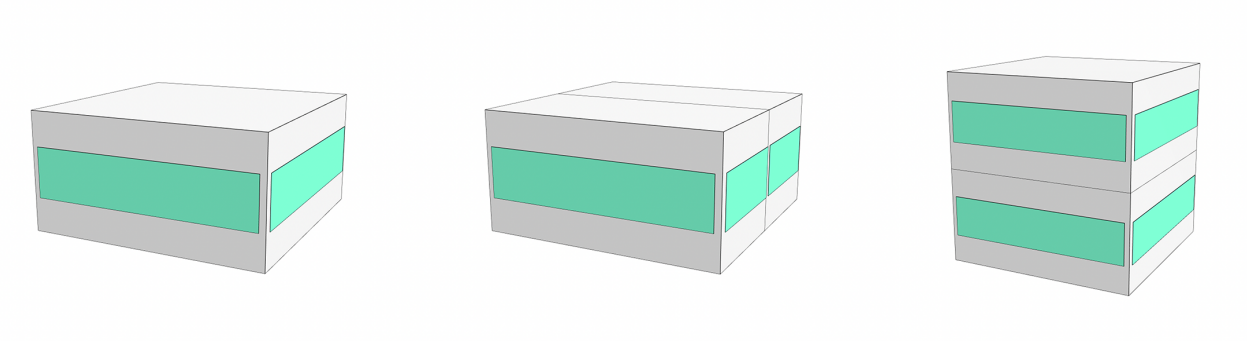


Figure 3.4: Visualization of DADU story and zone combinations (typologies) (from left to right: 1-story 1-zone, 1-story 2-zone, and 2-story 2-zone)

Dwelling units of various typologies were studied including one story-one zone, one story-two zone, and two story-two zone. Throughout this section and following, the terms zone and unit will be used interchangeably when referring to the model simulation process. One story-two zone DADU buildings were modeled as two horizontal adiabatically connected zones. The term adiabatic refers to a boundary condition (walls or floors here), of which there is no temperature difference across the surface. By doing so, this allows conclusions regarding the sharing of walls between DADUs to be drawn. Two story-two zone dwellings were modeled as two vertical, adiabatically connected zones. Figure 3.1.2 demonstrates that one-story one-zone dwellings were represented as a single standalone zone.

Assembly	Framing	Polyiso Insulation	Cellulose Insulation	R-wall ($\frac{ft^2 \cdot ^\circ F \cdot hr}{BTU}$)
0	2" x 4" wood framing	1"		10.7
1	2" x 4" wood framing		3.5"	15.5
2	2" x 6" wood framing		5.5"	22.2
3	2" x 8" wood framing		7.25"	28.1
4	2" x 10" wood framing		9.25"	34.7
5	2" x 10" wood framing	1"	9.25"	41.7

Table 3.2: Wall assemblies

Square footage was varied independent of typology between 400 and 1000 square feet, with the upper bound being the limit in accordance with land use policy. The lower bound of 400 square feet was selected as a rough expectation; however, this minimum could have been set lower in hindsight to add more breadth to the training data. Simulations were assigned floor area in discrete steps: 400, 600, 700, 800, 900, and 1000 square feet. Two infiltration rates were introduced to represent both typical and passive house conditions, with the former being 0.00059 $\frac{\mathrm{m}^3}{\mathrm{sec}\cdot\mathrm{m}^2\mathrm{facade}}$ and the latter a value of 0.00015 $\frac{\mathrm{m}^3}{\mathrm{sec}\cdot\mathrm{m}^2\mathrm{facade}}$. These values are based on previous simulation research and recommendation from Prof. Christopher Meek of the University of Washington. Window-to-wall ratio is a measure of glazing surface area to total wall surface area and was given a range between 0.10 and 0.90 (see Figure 3.1.2. Although values near the upper and lower limits of this range are quite atypical, they were included for added dataset breadth.

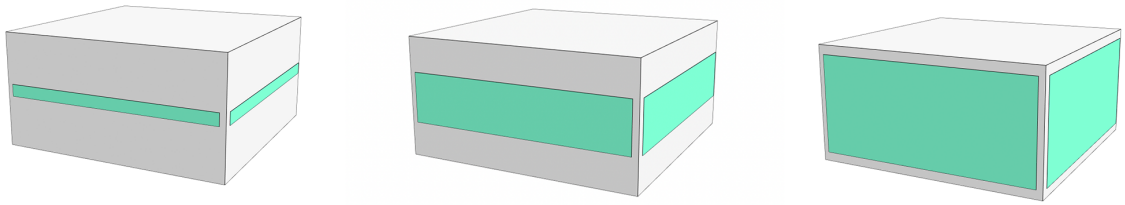


Figure 3.5: Visualization of application of window-to-wall ratio to DADUs (from left to right: 0.1 WWR, 0.4 WWR, and 0.9 WWR)

Likewise, the wall assemblies used reflect the pragmatic wood frame construction type commonly found in Seattle and the greater United States. Wood framing was specified for structure of varying nominal lumber. Blown cellulose and rigid polyisocyanurate (polyiso) insulation were included in varying amounts, as seen in Table 3.2. Floor surfaces were modeled as 0.50 inches of gypsum board and carpet padding, with a negligible air gap between. Foundation used across the board consisted of slab on grade. Due to EnergyPlus requirements, ceilings were defined as an assembly identical to the floor, but in reverse order of layers. Window specifications were kept constant across the design space with U-factor of

$0.35\frac{\text{Btu}}{\text{hr}\cdot\text{ft}^2\cdot^\circ\text{F}}$, solar heat gain coefficient (SHGC) of 0.25, and $T_{vis} = 0.63$. These values were chosen following guidance from Prof Tomás Méndez Echenagucia and Prof Christopher Meek to represent a typical window chosen for residential structures in Seattle.

Figure 3.6: Diagrams showing setback options (left to right: existing corner lot, existing infill lot, proposed corner/infill lot

Side and rear yard setback parameters were set in two stages. First, each simulation was assigned either an *existing* or *proposed* option. Based on the site typology assigned, individual rear, side, and structure setbacks were defined, taking into account both corner versus infill lots and whether or not an alley abutted the rear yard. For example, the *existing* option referred to current policy across the board for all site types. As for simulations given the *proposed* option, the rear yard setback and side yard setbacks (within the rear yard) were waived entirely, allowing for construction of a DADU up to the lot line. The implementation of this variable is to examine whether allowing homeowners to utilize more of their yard will lead to generally less energy consumption.

3.1.3 Workflow and scripts

Using a nested Python loop, a list containing every combination within this design space was created. Consequently, this created a list of lists, with each individual nested value coding for a different design parameter. Next, a string containing the simulation number and an identifier for each feature was generated(i.e. *50033_s_90_4_0_2_1_2_1.csv*) and used as .CSV

Lot type	Existing setbacks	Proposed setbacks
infill-alley	rear: 0', side: 5'	rear: 0', side: 5'
infill-no alley	rear: 5', side: 5'	rear: 0', side: 5'
corner-alley	rear: 0', side: 5'	rear: 0', side: 5'
corner-no alley	rear: 5', side: 5'	rear: 0', side: 5'

Table 3.3: Existing and proposed setbacks

filenames. In total these combinations number 51,840 total unique DADUs simulated. Next, another Python component was created in Grasshopper to decipher the currently selected value from this list. Extracting the variables, the values are sent to the related components along the pipeline as needed. Typically, one would have to toggle a button component within the Grasshopper workflow after setting all simulation parameters- for each simulation. However, through a workaround using the *animate* option of a Grasshopper slider, the slider can be automated to scrub through up to 10,000 values at a time. This in turn allowed for only six simulation batches to be completed in order to generate the entire dataset. On average each simulation required around 16 seconds, including the writing of the .CSV file. Time was the major inhibitor of generating a dataset any larger within the scope of this thesis. However, as detailed in the coming sections, there is inherent flexibility within this workflow, allowing for new variables to be simulated and new models to be trained.

Setbacks were drawn parametrically from the property lines using Grasshopper in order to compare resultant DADUs with both *existing* setbacks and *proposed*. From these setbacks, a buildable area was constructed, which in turn dictated the proportions of the DADU. Given the total square footage for the design combination, the script checks whether or not to create a single or two story DADU. For single story variants, the footprint of the building is equivalent to the square footage, while two story variants use half of the square footage as the footprint. Using all of the above information, the script then creates and centers a rectangle with area equal to the footprint, of equivalent proportions to the buildable area. From here, the outline of the DADU is extruded 10 feet. For two story dwellings, this boundary representation is duplicated above the original for a total height of 20 feet.

Next, the model was given to Honeybee to interpolate the 3D geometry into a format readable by OpenStudio and EnergyPlus. Honeybee operates using what it calls *rooms*, or as they are referred to throughout building simulation literature, as zones. These rooms are essentially an instance of a *class* in computer science terms. Within the Grasshopper interface, they are carried along between each step, gathering more data along the way. For two zone variants, Honeybee's *intersect solids* component was run to locate matching surfaces between each of the two boundary representations. Honeybee rooms are then created from the resulting geometry and the construction set consisting of exterior wall assembly, floor assembly, roof assembly, and slab on grade foundation conditions is applied. Similar to the step prior to last, the adjacencies between the two zones were solved, meaning coincident interior walls and floors were changed to adiabatic type. Subsequently, glazing is added to the room objects on all four facades using the supplied WWR value, along with the window assembly mentioned previously.

In a similar manner, infiltration rate and residential building program were given to the rooms next. No specific heating system is specified, leaving the model to simulate ideal air loads (this is corrected and explained in Section 3.4.1. The model itself is created using the room(s) plus nearby structures as shading objects. Lastly, orientation is applied by rotating the entire model in 90° increments until the desired relationship between DADU and existing dwelling is achieved. An orientation of *North* indicates that the principal dwelling lies to the North of the DADU. The aforementioned rotation is done within Honeybee and does not affect the 3D space proper. Orientation changes as modeled can be thought of as a means of adding more variety to the site typology by representing different shading directions from the principal dwelling. Honeybee then directs the model to EnergyPlus via OpenStudio for simulation and loads the location and climate data from an EnergyPlus weather file (.EPW). These files are created by capturing relevant weather data over the course of many years, primarily at airports and can be acquired for free online. The .EPW specified for this research was *USA_WA_Seattle-Tacoma.Intl.AP.727930_TMY3.epw*, collected at the Seattle-Tacoma International Airport and can be considered representative of the metropolitan area. In the case of adapting this methodology to another location, a different .EPW file would simply have to be retrieved.

Similarly to the method used by Honeybee to capture relevant data within the *room* objects, a class for each DADU simulation was written. This *dwelling* class begins by capturing input data from the loop which generated the list of combinations and adding information such as surface area or number of adiabatic walls as it moves through the pipeline. Using a Python class in this way simplified the simulation process, but was a task in itself. Grasshopper uses a variant of Python called *IronPython*, a .NET framework implementation of Python written in C#. Because of this, many popular Python libraries are unavailable and adding extra complexity. Upon completion of each simulation, the annual energy use results are calculated and broken down into cooling, heating, lighting, equipment, and domestic hot water (DHW) EUI values. These in turn are are then applied as attributes to the *dwelling* class instance and summed to compute the total site EUI in both kWh/m^2 and $kBTU/ft^2$. From this, the annual operational carbon was calculated and stored using the CO_2 equivalent specific to the Washington state electric grid of .096116 $kgCO_2$ (United States Environmental Protection Agency (EPA) 2022). Although the energy grid of Seattle proper is net-zero carbon, Washington state grid statistics were used instead to illustrate the methodology for other regions.

The *dwelling* class method *.write_csv* is then directed to create a single-row .CSV file containing all unique inputs of the simulation and the energy use intensity in a format identical to the last. Additional data captured by the Grasshopper workflow and written to the resultant .CSV files included: number of adiabatic surfaces, R_{wall}, volume, surface $area_{tot}$, surface $area_{glaz}$, surface $area_{opaq}$, surface-area-to-volume ratio, and buildable area. Finally, using the animation option to automate the sliders, the process could operate without user input for several hours at a time. However each simulation creates a screenshot in a folder specified, slowing down the flow and creating unnecessary clutter. To solve this, a Python script was added such that after each simulation, the entirety of the special folder containing the images was erased. This process was then repeated in sets of 10,000 simulations until the dataset was completed.

3.2 Data preparation

Following the dataset simulation process, 51,840 unique runs were saved in .CSV format. Using a Python script, all .CSV files within the folder directory were merged into one main

file. From here, the data was cleaned and prepared using Pandas, a data science library for Python. Pandas has many functions which can be useful for verifying a dataset, including one necessary for the first step in the process, *Pandas.read_csv()* (McKinney 2010). This function was then provided with the filepath of the combined .CSV file created earlier in order to convert the data into a dataframe object. The function *dataframe.shape* allowed for confirmation of the number of rows and columns present in the dataframe post-conversion. Similarly, *dataframe.head()* returns the initial five rows of data as another clarity check. Lastly, *dataframe.isnull()* was run in order to locate missing values and returned a dataframe of equal size as the input. Results of this dataframe read *False* where valid entries exist and *True* where null values are found. This allows for simulation samples with missing data to be rerun and calculations for operational carbon reverified. Runs needing to be performed once more were added to a list and simulated consecutively. All missing data was manually replaced with the results from each rerun for simplicity. In order to be confident, each of the previously-used Pandas functions were executed once more, with a conclusive outcome.

3.2.1 Feature selection

Feature selection is the final step before evaluating learning models, and is important to train models faster due to reduced complexity and number of features. Again, feature selection is the process of finding the optimal, lean set of features (independent variables) before the training process (Schreck 2018). This process was conducted in two portions, likely against proper machine learning methods. First, during the traditional process in this section preceding model training, and also after training utilizing a few techniques to inspect feature impacts. Using polynomial regression techniques, relative weights of each feature can be acquired. Weights equal to or approaching zero can be safely removed, while features whose weights are non-zero should be retained. Following this, the dataframe is then split randomly into separate *test* and *train* sets. This is to ensure that the model continues to perform well when shown new data. Using a random split. training data was set to be 80% of the full dataset, while testing data accounted for the remaining 20%. Without the *train-test split*, the accuracy of the model against never before seen data is difficult to estimate, but the method is also taxing on the quantity of data available for training. After the split, 41,472 simulation samples remained for training the machine learning model.

3.3 Model training

With the created dataframe completed and well-formatted using Pandas, the next step was to train the machine learning model. For this, a gradient boosting machine was selected, known as XGBoost (e**X**treme **G**radient **Boost**ing library) in part due to its prevalence as a building energy predictor (Miller, Hao, and Fu 2022). Being part of the GBM family, XGBoost offers high computational speed and model accuracy due to being an ensemble algorithm (D'Souza 2018). Additionally, XGBoost is difficult to overfit and flexible (Chen and Guestrin 2016). The primary downside to its use lies in the sheer number of hyperparameters involved, especially for those with less experience. However, this is remedied by freely-available information offering definitions of each, as well as general best usage scenarios. In order to begin training the XGBoost model using the simulated dataset, a few more preparatory steps need to be undertaken. XGBoost is particular in how the library interacts with data, requiring the use of a specialized data structure known as a *DMatrix*. These are optimized for memory efficiency and training speed according to XGBoost reference materials (Chen and Guestrin 2016). Luckily, passing the dataset from Pandas dataframe into a *DMatrix* is supported by default.

Next, parameters determining the number of folds used for cross validation and the number of boosting rounds were specified. K-folds cross validation is an additional measure to ensure the training data is varied and novel, while reducing model bias (Brownlee 2018). Set to a value of three, the variable k defines the number of *folds* to utilize. Essentially, the training dataset is randomly split into k groups, while each group takes turns being the test set while the others are used for training. In this case three models are trained, and the test scores of each averaged together. Typically, a value of 10 can generally be seen as the average number of folds selected (James et al. 2013). Importantly, as this data splitting occurs as part of the in-built *xgb.cv* (cross validation) method in this workflow, a value of three sufficed. This is because of the impact of boosting rounds in which each successive tree is updated using residual error of the last. The number of boosting rounds (*num_boost_round*) set for this stage was 100, ensuring significant depth without impacting training speed. This variable *num_boost_round* can be considered one of the hyperparameters required for training the XGBoost model, with more on this in the following section.

3.3.1 Hyperparameter optimization

Not all hyperparameters are modified in the process of optimization and these will be referred to as *fixed* hyperparameters. The previously mentioned *num_boost_round* is an example of one such fixed hyperparameter value. Hyperparameters such as these are manually set and are not optimized along with the rest. Fixed hyperparameters can still be adjusted in the case that they are severely impacting the model results. The remaining hyperparameters in this fixed category are *eval_metric*, *subsample*, and *eta*. Selection of the loss function used to evaluate the model during training is set via *eval_metric*, in this case using RMSE (XGBoost Developers 2021). Subsampling is the percentage of training data to sample from before growing the next round of trees, helping to reduce the chance of overfitting, and is defined by the *subsample* hyperparameter. Subsampling was set to 0.8, or 80% for overfitting protection (the default of 1.0 disables feature). Lastly, *eta* determines what is called step size shrinkage. Based on this value, after each round of boosting the associated weights of each feature are shrunk in order to keep the model more conservative as another means of combatting overfitting. Another alias for this hyperparameter is the learning rate in the context of most other machine learning models. Compared to the default of 0.3, this was designated as 0.1 resulting in slower model training and a decreased risk of overfitting.

On the other side, 4 values were allowed to vary when conducting hyperparameter optimization: *max_depth* (maximum depth), *gamma* , *lambda*, and *colsample_bytree* (subsampling rate for tree columns). Maximum depth constrains the depth of each tree trained, increasing complexity with deeper trees and has an allowed range of [0, ∞) (XGBoost Developers 2021). Supplied range for this variable was $[[3, 50]]$. *Gamma* controls what is called minimum loss reduction and dictates whether or not to create a partition at a leaf node, allowing values in range [0, ∞). Range specified for *gamma* was $[[0, 20]]$. Next, *lambda* specifies the level of L_2 (Ridge) regression applied to feature weights. Lower *lambda* leads to overfitting, but a value too large can result in an underfit model which generalizes too greatly. This parameter can technically be any value in the range [0, ∞), but in nearly all cases will fall between a range $[0, 1]$ (James et al. 2013). In this case the latter range was selected. Finally, *colsample_bytree* determines the subsampling rate for tree columns meaning the portion of tree columns to sample from when creating the next boosted tree. In this case, a suffic of *..._bytree* denotes

that this occurs only once per tree grown (XGBoost Developers 2021). This value can range between $(0, 1]$, with a default value of 1. Here, the range was defined as $[0.3, 0.9]$, with a step of 0.1 increments.

Three distinct methods of hyperparameter optimization were tested and ultimately included within the Python script responsible for model training (with a toggle). These include random search, grid search, and Bayesian search. Results from both random and grid search methods lacked accuracy and speed, respectively. By definition, random search does what its name implies. After supplying a number of random targets to search, random search iterates through keeping track of the permutation of hyperparameters which returns the model with lowest loss. In nearly opposite fashion, grid search tests every combination of hyperparameter values and returns the combination with the lowest model loss. In practice, random search returned a model with between double to triple error value than the ideal solution found through grid search. However, random search completed in under five seconds whereas grid search took upwards of an hour to compute.

In the end, an approach taking advantage of Bayesian search was employed, using the *bayes_opt* Python library. For greater detail into how Bayesian search and optimization operates, see Section 2.5.3. Bayesian optimization completed in the time it takes random search to finish, with 99% veracity when compared to the grid search method. Additionally, when using *bayes_opt*, the hyperparameter*lambda* was not included, reducing complexity. Upon retrieving the hyperparameter values resulting in the model with the lowest loss, EUI was predicted for all values within the test dataset. Final RMSE loss was calculated by comparing the validity of these predictions to the true (simulated) outcome, which was kept hidden from the trained model.

3.3.2 Model deployment

Through the use of a few simple charts, impacts of each feature on the EUI was visualized (more on this in Section 4.2). It was at this stage that the second round of feature selection was conducted after noticing discrepancies in these impacts. For example, after initial feature selection, features *surf_glaz* and *wwr* both remained for training. Total surface area of glazing (surf_glaz) and window-to-wall ratio (WWR) can be thought of as two competing methods of

describing window coverage. As WWR is a primary input and total surface area of glazing is derived from the design, WWR was selected to remain. Removing similar confounding pairs of features increased model training speed, reduced model loss, and revealed clearer patterns in feature impacts on EUI. Finally, once a suitable loss was achieved during training, the model was serialized into a binary file using *pickle*, a native Python input/output module. Exporting the trained machine learning model in this universal Python-readable format allows for a straightforward update and deployment process within the web application.

3.4 Webtool development

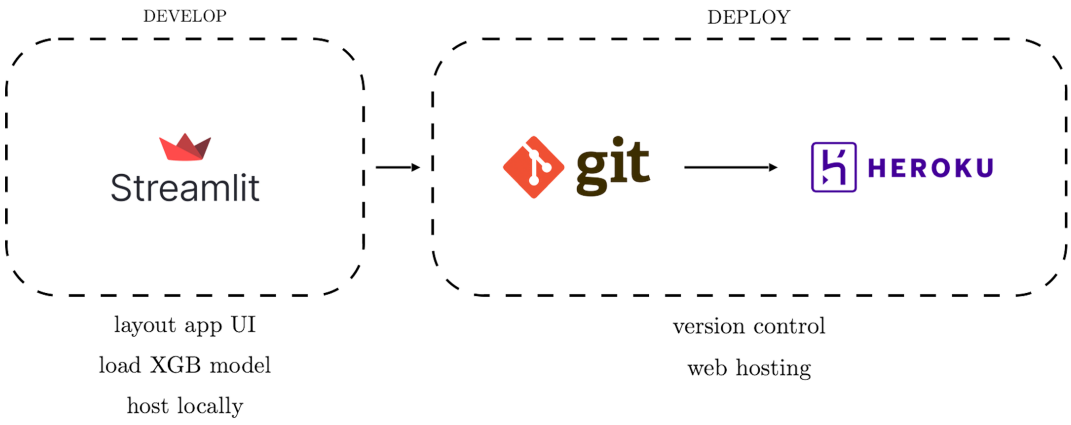


Figure 3.7: Flowchart of web application development process

Creation of the web application dashboard focused on the potential range of users including designers, developers, and homeowners alike. In order to facilitate this goal, research into user friendly toolkits, libraries, and other options for development of a web application was conducted. *Streamlit*, a Python-based dashboard style web app builder was ultimately selected. The ability of Streamlit to maintain a continuous programming language across front-end and back-end development, along with no requirement for knowledge in web design. Streamlit is developed as a singular Python script that runs top to bottom each time any object on screen is updated or interacted with (buttons, text entry, etc.). This trait solves some problems but creates others, from unloading cached user data to having to load the machine learning model again each time. In the end, these limitations were overcome through determination and assistance from users on the Streamlit forums and wiki. Another benefit

of Streamlit is the ability to develop and debug the application locally, while a stable version can be deployed remotely on the cloud through Streamlit or another hosting service provider. Each time a small modification was pushed through to test a change the application updated in real-time, smoothing out the development process. Ultimately, the application was hosted using *Heroku*, a data-science dashboard hosting service.

3.4.1 Application workflow

As the web application runs top to bottom through the code upon every update, it makes sense to work through the mechanics of the tool in the same order. Development of the web application and model training occurred in tandem, allowing for the web tool to verify model results using its data visualization. Additionally, verifying model results in this way allowed for thorough testing of the web application. Step one is to load the *pickled* XGBoost model, detailed in Section 3.3.2. However, the top to bottom runtime means this would need to occur each time, becoming computationally expensive. Except this can be ignored following information from the Streamlit documentation allowing for the app to cache this data into the session state, a repository of relevant data from the current session. Any data from functions to dataframes to individual variables can be stored in the session state by simply prefixing the object name with *st.session_state.* Data stored in this way persists until the cache is cleared by an administrator for this specific user, the user refreshes the browser tab, or an undefined period of inactivity occurs.

Next, a sidebar on the left-hand side of the window collects user input for computing energy predictions. This is followed by a "Predict" button, which when activated bundles the user-selected feature values into a prediction input dataframe for the ML model to predict from. When users select values within the sidebar, their values are matched against values in a corresponding Python dictionary object in order for the inputs to match what is expected of the ML model. For example, when a user selects a DADU typology value of "1 story 2 zones", this string is found in the typology dictionary, allowing for both *num_stories* and *num_zones* to be set from one user input. For all design features other than floor area, WWR, and the depth of both cellulose and polyiso insulation, the range of options mirrors the simulation parameters detailed in Figure 3.1. Within these outliers, input floor area allows any integer including and between 400 and 1000 square feet. Window-to-wall ratio is similarly allowed

between 0.00 and 0.90, with precision to the tenths place as opposed to the delineated steps of 0.10 used during dataset simulation. Polyiso insulation occupies the range between 0.00 and 1.00 inches in steps of 0.25, while cellulose insulation ranges between 0.00 and 10.00 inches incrementing in steps of 0.50. Once the desired options are selected, users can click "Predict" to advance.

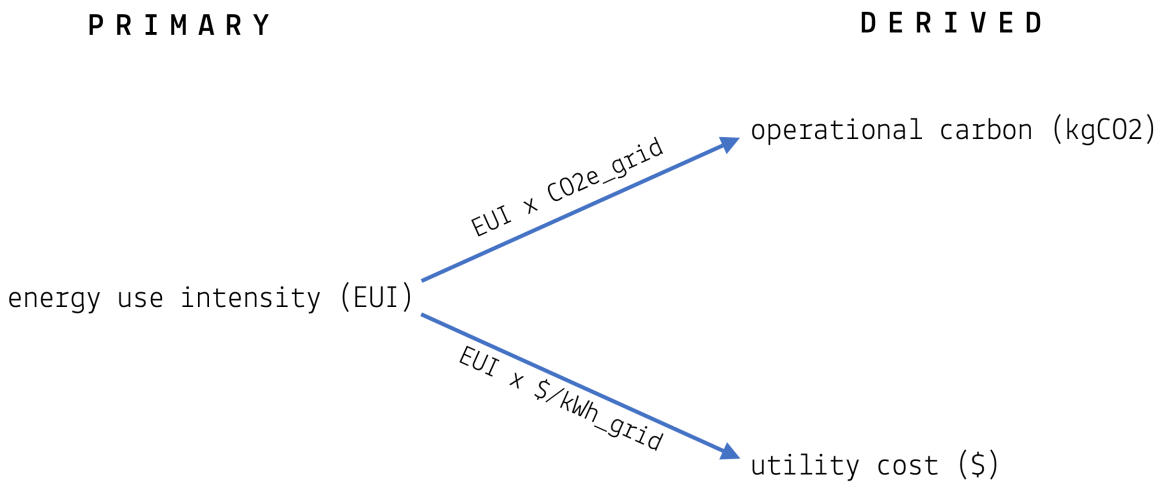


Figure 3.8: Flowchart of simulation outputs

Once clicked, the predict button toggles a boolean statement containing a function call to create the input dataframe using sidebar inputs. Next, the *predict_eui()* function takes this input dataframe and returns a single EUI prediction. In order for this energy consumption to be reflective of an industry standard mini split heat pump, the predicted EUI is divided by a coefficient of performance (COP) of 3.5 (Winkler 2011). The value of 3.5 was calculated by using an average annual outdoor air temperature of 52° F in Seattle, along with a theoretical setpoint of 70° F. This scalar modification is necessary to correct for the fact that the Honeybee model calculated heating EUI using ideal air loads. This EUI is translated into kWh/m^2 by multiplying by a conversion factor of 3.2, in order to more easily calculate operational carbon and utility costs. Operational carbon is determined by removing the

floor area normalization and then factoring in the 0.096116 $kgCO_2/kWh$ Washington grid carbon density (United States Environmental Protection Agency (EPA) 2022). Utility costs are calculated using the metric EUI and Seattle City Light average of $0.1189 per kWh of electricity (Eisenbach 2022). These results, along with the user input are added as a row in the results dataframe, allowing the user to compare outcomes and also to download before ending their session.

3.4.2 User interface

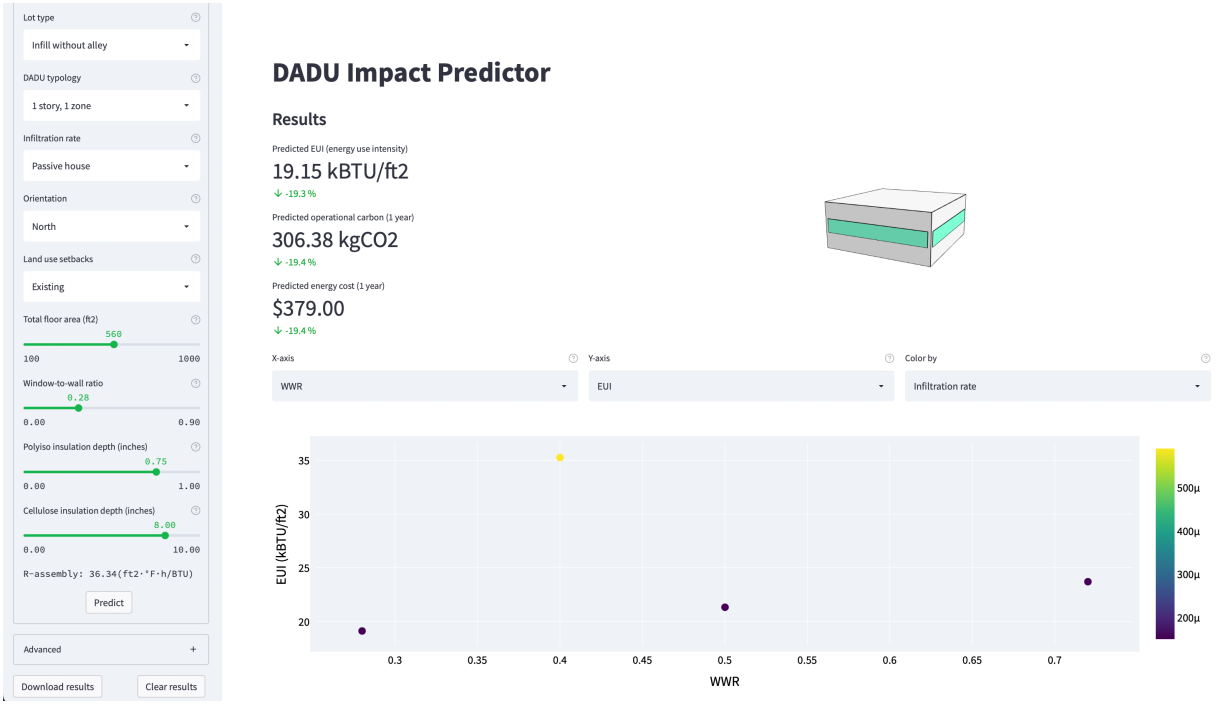


Figure 3.9: User interface of metamodel web application

Within the aforementioned sidebar, input methods for lot type, DADU typology, infiltration rate, orientation, floor area, WWR, setbacks, polyiso insulation depth, and cellulose insulation depth nearly mirror the final iteration of inputs for prediction results. The only difference being that users select these two amounts of insulation, before a function within the web tool calculates and displays the R-value of wall assembly. The input method for these options varies between slider and selection menu widgets. After this there is an "Advanced" tab, allowing users to change the time period for operational carbon and utility costs between options of 1 month, 6 months, 1 year, 5 years, and 10 years, with 1 year being

the default value. Additionally, the advanced options section allows for the viewing of the results dataframe (used mostly for troubleshooting and advanced users). Finally, the bottom of the sidebar contains options to either download or clear the results. Clearing the results scrubs all values from the results dataframe, but user selections from the previous prediction may still remain in the sidebar. Alternatively, choosing download prompts the user to allow for a .CSV download. This file is named using user system time and date information for safe keeping.

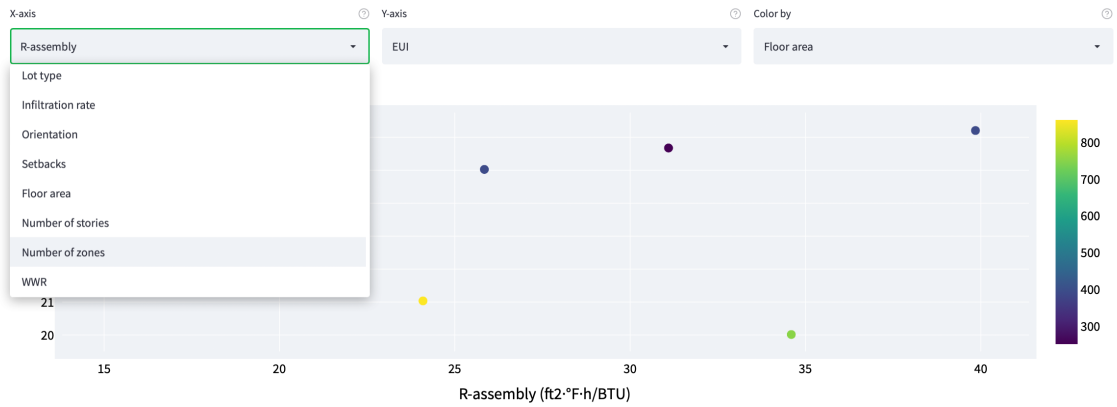


Figure 3.10: Web application results plot interface

To the right, two half page panels border the top of the interface containing prediction metrics on the left and a simple interactive 3D model of the DADU to the right (Figure 3.9. The metrics section includes results of EUI, operational carbon, and utility costs, with the latter two displayed along the prediction period set by the user (default: 1 year). After a second prediction has completed, the percent change for these three values can be seen. On the right hand side, this 3D model makes use of a *Plotly 3D mesh* object and can display DADU typology and WWR information independently (Figure 3.10). Users can rotate and zoom in or out within these models for a better understanding. Below these two half-width panels, a 2D scatter plot was added using Plotly. This graph allows users to plot features on the X-axis and predicted values (EUI, CO_2, utility cost) on the Y-axis. X-axis options include: site, infiltration rate, orientation, setback, gross floor area, WWR, and R-wall assembly. As each prediction is made and added to the results dataframe, this plot adds the prediction as a point for comparison. To allow for cleaner visualization, a colorscale option was included with the same selections as the X-axis menu. For example,

one can visualize the impact of floor area on operational carbon, colored by window-to-wall ratio across all previous predictions. This inclusion became mutually beneficial to both the model training/validation and web application development processes. Model results were evaluated within the web tool, and many times pointed out bugs in the software to remedy. Plotting multiple predictions simultaneously was fundamental to the comparative analysis component of this research as well.

3.5 Comparative analysis

Figure 3.11: DADUs used in comparative analysis (left: Cedar DADU, right: Cloud DADU) (CAST Architecture 2021)

Methods for validating model accuracy are statistical in nature and provide an assessment of the real-world accuracy of the application. Model accuracy, in this metamodeling context, refers to how far the predictions are from the simulated result- not the actual, physical result. Additionally, testing the limits of accuracy of the model can be useful in evaluating its efficacy. In order to understand how this is imperative, we must return to the concept of the train-test split, where data was partitioned before model training. This was to make sure the loss score calculated for the trained model was indicative of predictions against new, never before seen data. Comparisons between predicted (by web application), simulated (by EnergyPlus), and measured (metered energy consumption) would prove to be ideal. However, energy use intensity data to match each set of predicted and simulated values is not available. In kind, a comparative analysis method was utilized to contrast predictions made via the

web application and EnergyPlus simulation results of real-world DADU designs. CAST Architecture supplied plans and details of two DADUs that they had designed through their CAST Cottages arm: the *Cedar* DADU and *Cloud* DADU. The Cedar is part of the 10 Seattle pre-approved DADU plans featured on the ADUniverse site, while the Cloud was designed for a private client City of Seattle 2019a. These designs have both quantitative and qualitative differences from the training dataset. Formal variations such as shed roofs and *non-shoebox model* plans should theoretically result in substantial differences in results. Further, one DADU featured in the comparison has a gross floor area of 1200 ft^2, while the surrogate model was only trained using data with floor area of up to 1000 ft^2.

3.5.1 Energy model evaluation

Consisting of two schemes, the one-story Cedar DADU includes the Cedar-A and Cedar-B options of 476 and 636 ft^2, respectively. Cedar-A is a 1-bedroom DADU with a full bathroom and kitchen. Cedar-B is identical but the additional gross floor area included is due to an additional bedroom. The 3-bedroom, 2-bathroom Cloud DADU comprises 1221 ft^2 and two-stories. To simulate the energy consumption, a variation of the Honeybee script used to generate training data for the surrogate model was used. Instead of a list of design combinations, each DADU was 3D modeled independently. In order to represent the building geometry for Honeybee, each was modeled as simplified volumetric solids. To test results across all three typologies, Cedar-A was treated as a one-zone scheme, whereas the Cedar-B and Cloud DADUs were represented as two zones (see Figure 3.12). Glazing was modeled according to the plans as individual surfaces within Rhino, as opposed to being applied using the Honeybee *aperture by ratio* component informed by the window-to-wall ratio.

Wall assemblies were constructed using the same Honeybee components as from the dataset generation script, with the two Cedar DADU instances having R-21 walls and the Cloud DADU with R-31. This value was selected by using the typical exterior wall construction for each DADU as the generalized exterior wall construction within Honeybee. Design variables were recorded much the same way from the original script by using the created *dwelling* class object and recording applicable information. In contrast to the design inputs from the design space generation step, values such as WWR and floor area were calculated rather than specified as a numerical input. WWR was determined by dividing the total glazing

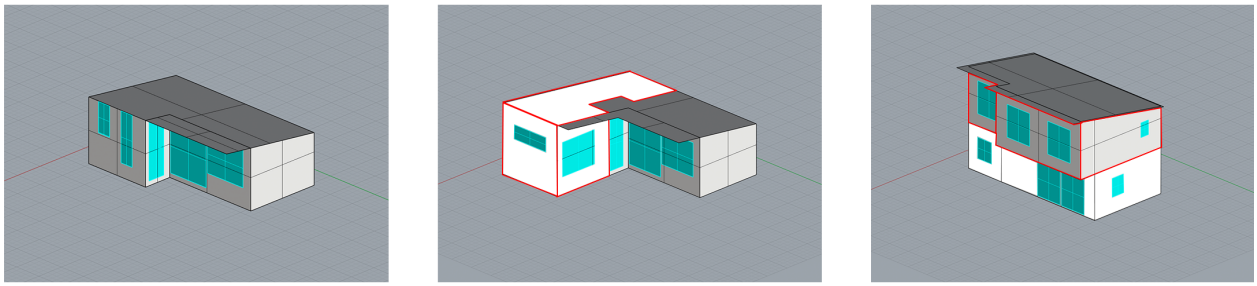


Figure 3.12: 3D models of each CAST Architecture DADU (from left to right: *Cedar-A*, *Cedar-B*, and *Cloud*) (zones demarcated in red, where multiple)

surface area by the surface area of the DADU walls. All variables not specified by design documentation for these DADUs were kept constant, including a typical infiltration rate of 0.00059 $\frac{\mathrm{m}^3}{\mathrm{sec}\cdot\mathrm{m}^2\mathrm{facade}}$, infill-alley site typology, north orientation, and existing setbacks. This specific site was selected due to its buildable space being able to contain the 1221 ft^2 Cloud cottage, as well as for average shading in comparison to the other 3 modeled sites. Upon simulation completion, EUI results were appended to the *dwelling* instance before being written to .CSV format using the previous *write_csv()* method. Refer to Figure 4.4 in Section 4.3 for specific design criteria for each DADU instance. Because many of these features were not rounded, even values (e.g. 0.17 WWR), predictions were made outside of the web application by loading an instance of the trained model and feeding it each set of simulation data: design inputs and accompanying EUI.

Chapter 4

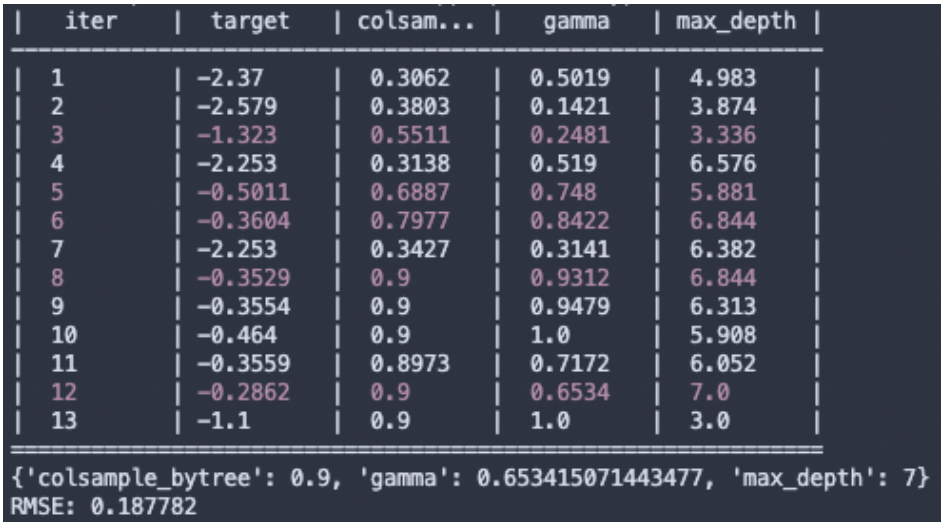
RESULTS

Outcomes from this research represent three general areas of inquiry: metamodeling accuracy, comparative analysis results, and evaluation of policy ideas. Connecting the results from each of these three areas allows for the answering of the original research questions (see Section 1.3). Metamodeling accuracy will demonstrate the validity of prediction results within the context of the simulations. The ability of the metamodel to predict credible energy use intensity of DADU designs it has not previously encountered by combining the preceding accuracy findings with the insights from the comparative analysis. Policy questions surrounding waiving of rear yard setbacks and allowing for the sharing of DADU walls across lot lines were investigated by comparing the metamodel predictions across attributes in question.

First, the accuracy of the final metamodel will be examined in section 4.1. Metamodel accuracy broadly refers to the ability of the model to return a result within an acceptable threshold of error. Tolerable error is subjective and regularly based on the use of the surrogate model. As the purpose of this application is to deliver homeowners, designers, and developers with performance estimates in the early design stages specifically, extreme accuracy is not critical. Correspondingly, results in section 4.3 from the comparative analysis show whether or not the metamodel is well generalized- that is, capable of realistic predictions on novel data. Finally, outcomes explored at the simulation, metamodel, and application stages regarding the practicality of proposed policy ideas will be explored in section 4.4.

4.1 Model accuracy

This section will begin with an evaluation of the final XGBoost model utilized for predictions shown throughout the results chapter. While not fully tallied, the total number of trained models saved was likely greater than one hundred over the course of the development pro-



iter	target	colsam...	gamma	max_depth
1	-2.37	0.3062	0.5019	4.983
2	-2.579	0.3803	0.1421	3.874
3	-1.323	0.5511	0.2481	3.336
4	-2.253	0.3138	0.519	6.576
5	-0.5011	0.6887	0.748	5.881
6	-0.3604	0.7977	0.8422	6.844
7	-2.253	0.3427	0.3141	6.382
8	-0.3529	0.9	0.9312	6.844
9	-0.3554	0.9	0.9479	6.313
10	-0.464	0.9	1.0	5.908
11	-0.3559	0.8973	0.7172	6.052
12	-0.2862	0.9	0.6534	7.0
13	-1.1	0.9	1.0	3.0

Figure 4.1: Model training results log file

cess. As discussed in Section 3.4, the paradigm of designing the web application in tandem with the training of machine learning models allowed for the hot-swapping of these models once a superior variant was discovered. Inspecting the console log files from training each model, it can be recognized that many times it takes until nearly the last iteration to find a satisfactorily optimal result. The logs (Figure 4.1) from the final model for example show that its 12th iteration was found to be the most optimal hyperparameter combination. This ultimate model featured an RMSE loss value of only 0.188. As RMSE units match the base unit of the independent variable, we can understand this as on average, predictions are off by only ± 0.188 kBTU/ft^2. Using only 9999 simulation values, model accuracy was found to have a comparably large 1.24 RMSE (see Figure 4.2). If predictions are made across the entire testing dataset and then plotted against the ground truth values, we acquire a plot representative of model accuracy (see Figure 4.3). Overall, model error between these two points in time decreased by around 85%.

Hyperparameters discovered for the best model are as follows: colsample_bytree = 0.9, gamma = 0.65341, and max_depth = 7. This translates to a subsampling rate of 90% for each column, meaning 10% of data was discarded each time a new tree was grown. Gamma equivalent to this value is relatively average and should not have affected the chance of partitioning at leaf nodes extremely heavily. Unusually, a maximum depth of 7 splits deep per tree is rather small- most prior models had values in the 20-30 range.

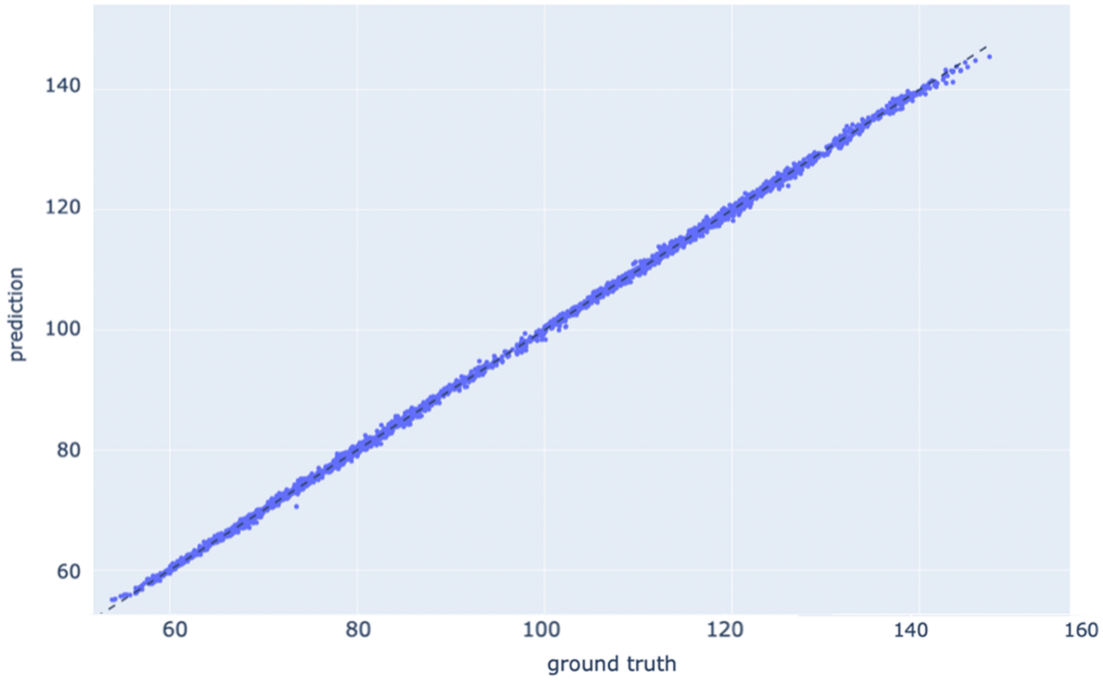


Figure 4.2: Model accuracy at dataset size: $n = 9999$

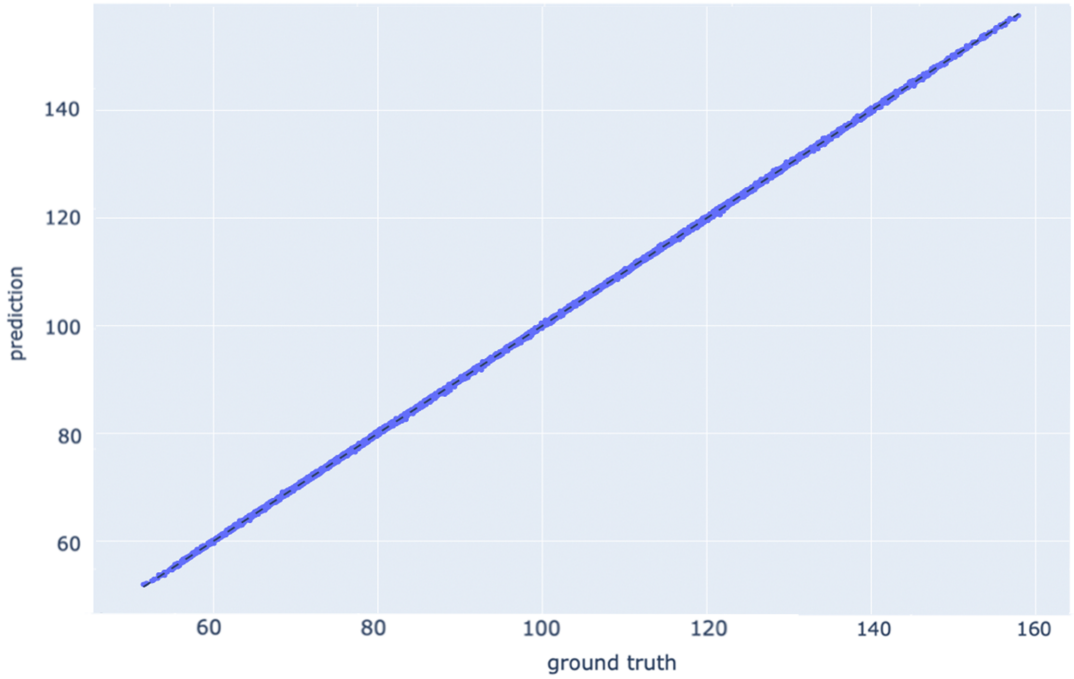


Figure 4.3: Model accuracy at dataset size: $n = 51840$

4.2 Relative impacts of features

As a research question, the impact of each feature on EUI was of principal interest. Thankfully, XGBoost's tree regression method interfaces well with calculating feature importance through the use of a metric called an F-score. This score measures the discrimination of two sets of real numbers. In this case, a weight is computed by counting the number of times each feature appears within all trees during model training. With these weights in hand, the F-score can be computed between each weight set. This fairly straightforward criterion can be useful for feedback in the feature selection process for data scientists, but also in this case for understanding the proportional impact of each variable on DADU energy use intensity. One drawback of this approach however is that F-scores do not reveal mutual information between features (Chen and Lin 2006).

The resultant feature importance plot generated from the selected metamodel displays window-to-wall ratio (WWR), surface-area-to-volume ratio (SVR), R-value, and gross floor area as the top four impactors. As these F-scores are not normalized in the graphic, we can reduce these four scores by dividing each by the lowest of the four, that being *size* (or gross floor area) at 980. Upon doing so, we achieve F-scores as follows: WWR = 2.34, SVR = 1.81, R-value = 1.52, and gross floor area = 1. From this data we can conclude that WWR has 2.34 times greater influence over EUI predictions than gross floor area. SVR, a factor of 1.81 times more impactful than gross floor area on energy use, and so on.

In addition, SHAP plots can be used to gain insight into feature importance results. SHAP values (or **SH**apley **A**dditive ex**P**lanations) are based on game theory, standing features in as theoretical players in a scorable cooperative game (Lundberg and Lee 2017). The score of the game is the independent variable to predict in machine learning scenarios. Features can display a high or low impact akin to F-scores, but these can be combined with classifications of positive or negative contributions. Additionally, SHAP values can display the impactfulness of these contributions across all values of each feature in contrast with F-scores (Trevisan 2017). This can be useful in scenarios where a feature has a non-linear relationship with the prediction, beneficially impacting the outcome at low values but having negative consequences at high values, for example. The range of each feature in Figure 4.5 is actually composed of points representing every prediction made. Hence why points plotted across

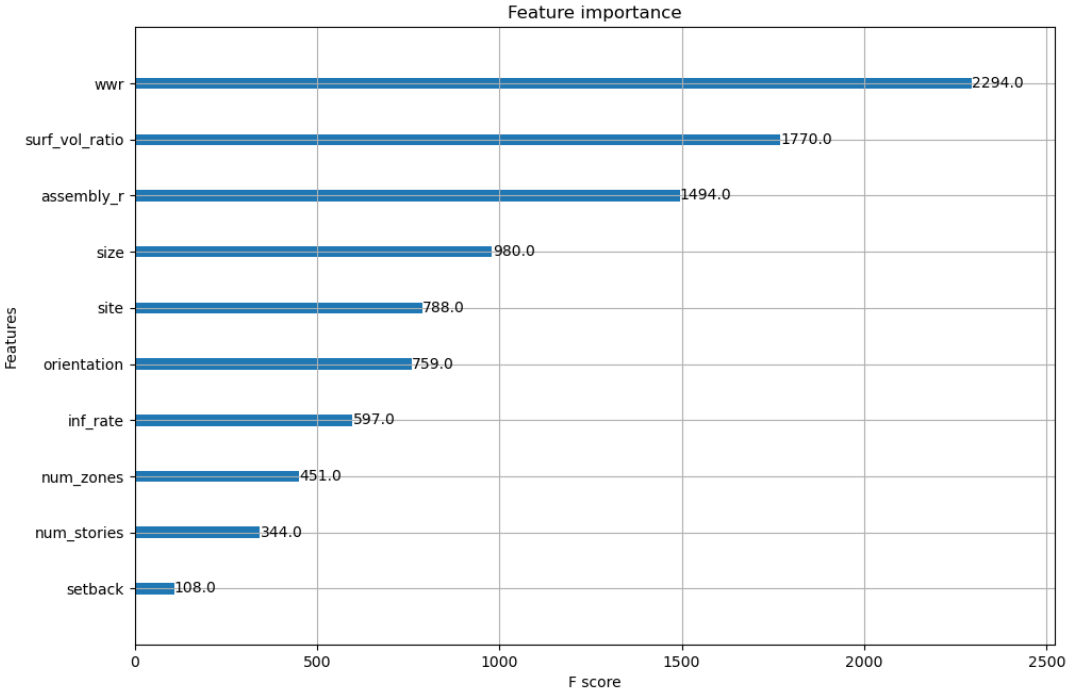


Figure 4.4: Feature importance plot

Feature	Value
Site	infill-no alley
Typology	1-story, 1-zone
Size	400 ft^2
Infiltration rate	0.00059 m^3/sec
WWR	$(0.1, 0.9)$
R-value	36.34 $\frac{ft^2 \cdot °F \cdot hr}{BTU}$
Setbacks	existing
Orientation	north

Table 4.1: Input design space for WWR impact

infiltration rate, while only possessing two options, these points are not perfectly coincidental. Values closer to red represent predictions where the feature was of a higher numerical value, while purer blue hues display minimum feature values. Negative SHAP value translates to a reduction in EUI predicted, while a positive SHAP on average equates to higher EUI.

High window-to-wall ratio, surface-area-to-volume ratio, and infiltration rate, among others, are heavily correlated with an increase in energy use intensity. In contrast, a higher gross floor area and wall R-value lead to a lower EUI prediction. Gross floor area directly contributes to a lower EUI, as the metric is normalized by floor space. This trait typically carries forward well enough in larger scale buildings, as building loads scale with physical size. However, in the design space operated on within this research, building loads are constant and the mini-split heat pump HVAC system is also not scaling to square footage. Likewise, greater levels of insulation allow structures to maintain interior temperatures closer to the setpoint for longer periods of time, in turn increasing energy efficiency. Less straightforward variables such as site display muddied results. Site variation can be explained in that for the most part, all four sites shared similar levels of shading and lot sizes.

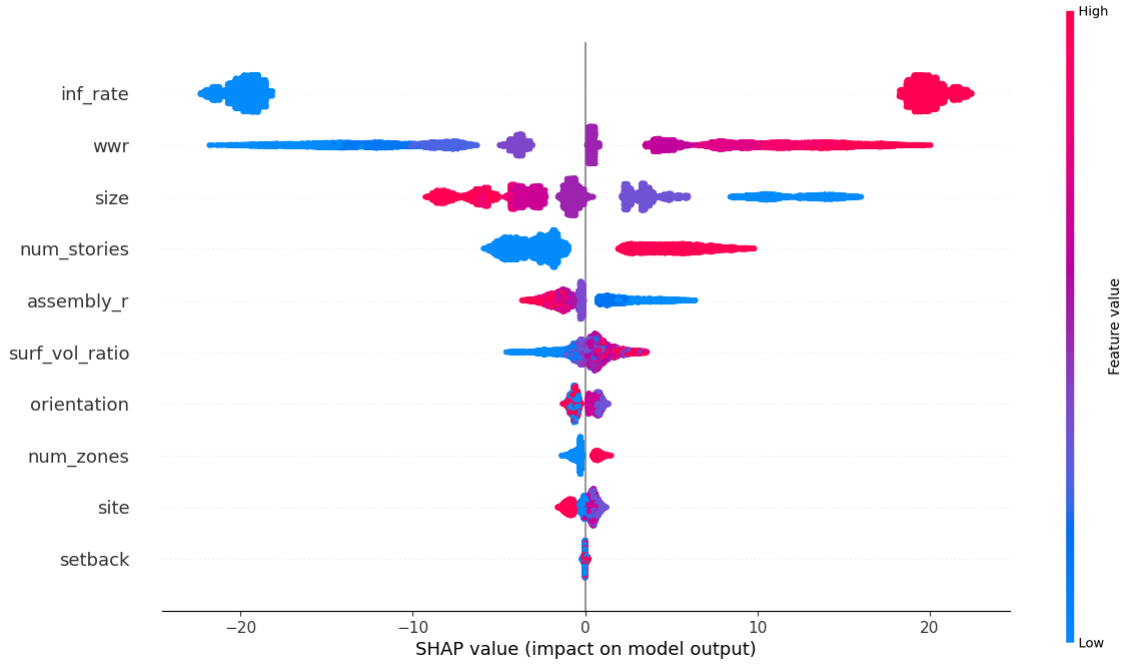


Figure 4.5: Feature SHAP value plot

4.2.1 Window-to-wall ratio impact

Based on the results from the final feature importance plot, it can be concluded that together, R-value of the wall assembly, window-to-wall ratio, and surface-area-to-volume ratio have the greatest effect upon energy use intensity in DADUs. Beginning with the window-to-wall ratio, all simulation data matching the criteria shown in Table 4.1 while varying WWR were plotted against web application predictions in Figure 4.6 using the same static criteria while similarly modulating WWR. From this graph, we can see that DADU window-to-wall ratio has a fairly linear relationship with EUI in a vacuum. Increased glazing coverage generally leads to an increased energy use intensity in the absence of other changes (plug loads, HVAC system changes, etc.).

4.2.2 Surface-area-to-volume ratio impact

Moving on, both prediction and simulation results concerning surface-area-to-volume ratio were plotted against EUI in similar fashion (see Table 4.2 and Figure 4.7). Important to note, as SVR is not an option for user input within the web tool, gross floor area was instead

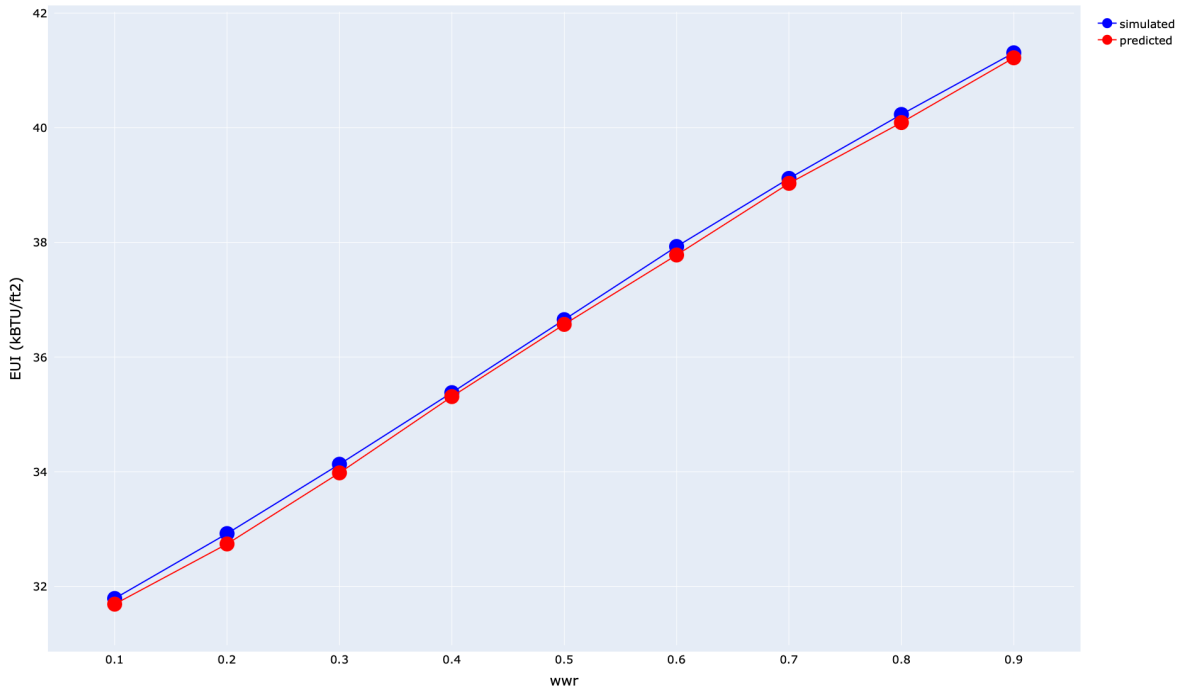


Figure 4.6: Window-to-wall ratio versus energy use intensity

varied. Surface-area-to-volume ratio is instead calculated based upon dimensions acquired via gross floor area and the number of floors (i.e. height data), so this operated as a functional proxy. In contrast to the energy impact of window-to-wall ratio, surface-area-to-volume ratio exhibits a positive linear relationship with EUI until SVR reaches 0.425, which is when the relationship appears to become nonexistent when viewing the predicted data. Comparing this simulated data shown in Figure 4.7 in blue, we can see that only the (0, 0.425) SVR conjecture is supported by the "true" results. This anomaly is likely due to the method in which XGBoost operates for predictions: tree-based logic. Perhaps XGBoost lacks the ability to interpolate for regression predictions, instead returning the nearest valid prediction value; however, no literature supporting this notion was identified.

4.2.3 R-value impact

In the final analysis of wall assembly R-value (R-assembly), results were plotted using this same methodology. Simulation data matching Table 4.3 criteria were plotted along with predictions via the web application versus energy use intensity while varying the R-assembly

Feature	Value
Site	infill-no alley
Typology	1-story, 1-zone
Size	$(200, 1000)$ ft^2
Infiltration rate	0.00059 m^3/sec
WWR	0.4
R-value	36.34 $\frac{\text{ft}^2.^\circ\text{F}\cdot\text{hr}}{\text{BTU}}$
Setbacks	existing
Orientation	north

Table 4.2: Input design space for SVR impact

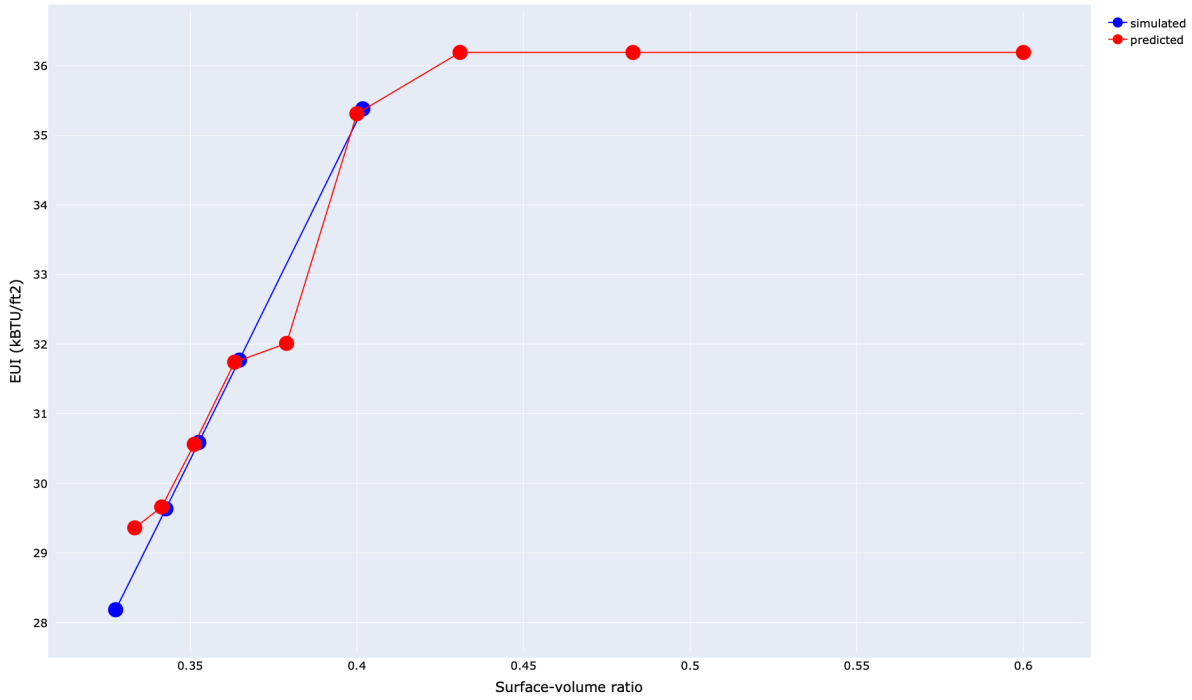


Figure 4.7: Surface-area-to-volume ratio versus energy use intensity

Feature	Value
Site	infill-no alley
Typology	1-story, 1-zone
Size	400 ft^2
Infiltration rate	0.00059 m^3/sec
Window-to-wall ratio	0.4
R-value	$(3.0, 36.34)$ $\frac{ft^2 \cdot °F \cdot hr}{BTU}$
Setbacks	existing
Orientation	north

Table 4.3: Input design space for R-value impact

feature. In Figure 4.8, we see that simulated results illustrate a seemingly negative exponential relationship between wall assembly R-value and energy use intensity. Increased insulation represented by R-value leads to greater reductions in EUI between R-10 and R-20 than reductions in EUI past wall assemblies of R-30 or greater. These results are consistent with architectural wisdom, as increased insulation creates a reduction in heat loss through the building envelope. In contrast, prediction results closely follow this same pattern, however the line fit to these data appears to be nearly a stepwise function. These continuations of the same EUI even as R changes occur when the value of R is between those discrete R-values simulated in the training set using the six wall assemblies. This is likely due to the same behavior explained above and seen in Figure 4.7.

4.3 Comparative analysis outcomes

Evaluation of three CAST Architecture DADU plans using both metamodel predictions and EnergyPlus simulations delivered compelling results. Once simulations were run for each of the three DADU schemes, relevant data values were recorded to a .CSV file in the same format as the training data. The number and naming of all columns was required to mirror the training data in order to be successfully fed into the metamodel to return predictions. Refer to Table 4.4 to view criteria for each of the three DADU schemes. Once submitted to the XGBoost regressor model, three corresponding EUI predictions were returned.

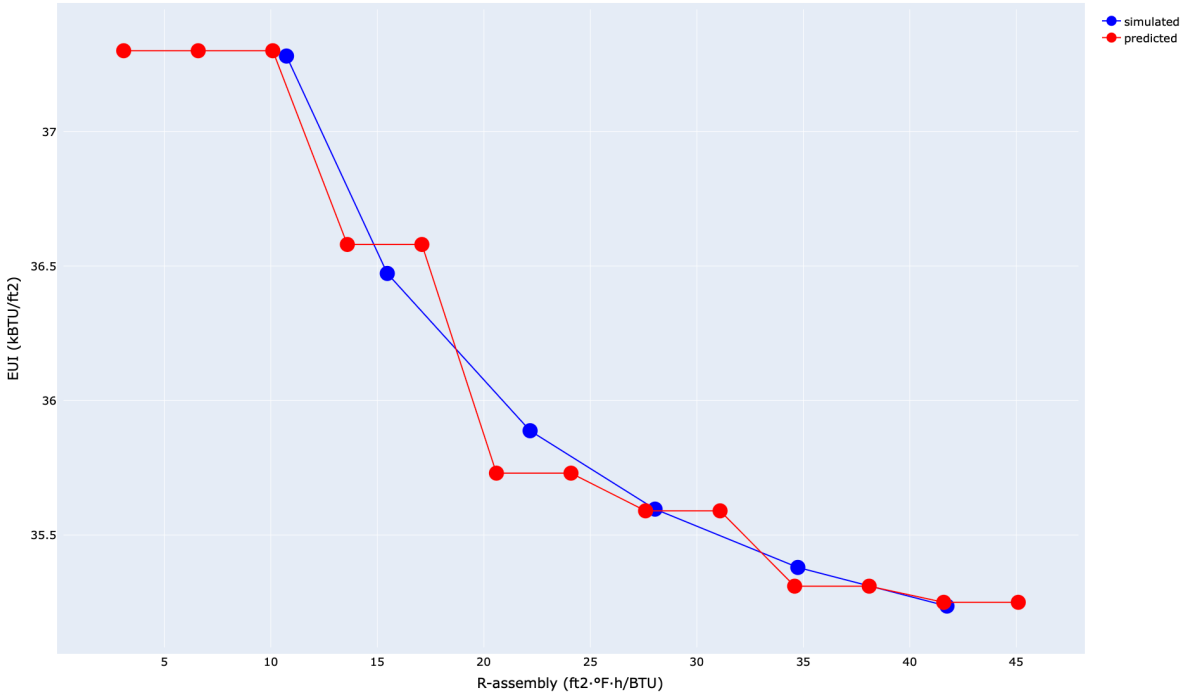


Figure 4.8: R-value of wall assembly versus energy use intensity

	DADU		
Feature	Cedar-A	Cedar-B	Cloud
Site	corner-no alley	corner-no alley	corner-no alley
Typology	1-story, 1-zone	1-story, 2-zone	2-story, 2-zone
Size (ft^2)	476	636	1221
Infiltration rate (m^3/sec)	0.00059	0.00059	0.00059
WWR	0.206	0.199	0.1646
R-value ($\frac{ft^2 \cdot °F \cdot hr}{BTU}$)	22.49	22.49	31.41
Setbacks	existing	existing	existing
Orientation	north	north	north

Table 4.4: DADU simulation features for comparative analysis

Recording these to their respective .CSV files, the next step was to calculate the difference between the simulated and predicted EUI for each DADU design. For each of the three schemes, predicted values severely underestimated the energy use intensity in comparison to the Honeybee simulation results.

Additional data points analogous to the schemes were predicted in order to situate these results within the web application prediction range much better. Keeping all features constant and equal to the three DADU designs individually while varying gross floor area allowed for a fair assessment. All three DADUs were simulated sharing identical site selection, setbacks, infiltration rate, and R-wall assembly (for both Cedar cottages). Outside of these commonalities, DADU typology (number of stories and number of zones), window-to-wall ratio, and surface-area-to-volume ratio varied between schemes but held constant in relation to each DADU. Gross floor area was allowed to vary from 450 to 1300 ft^2 in order to encompass each of the simulated DADUs.

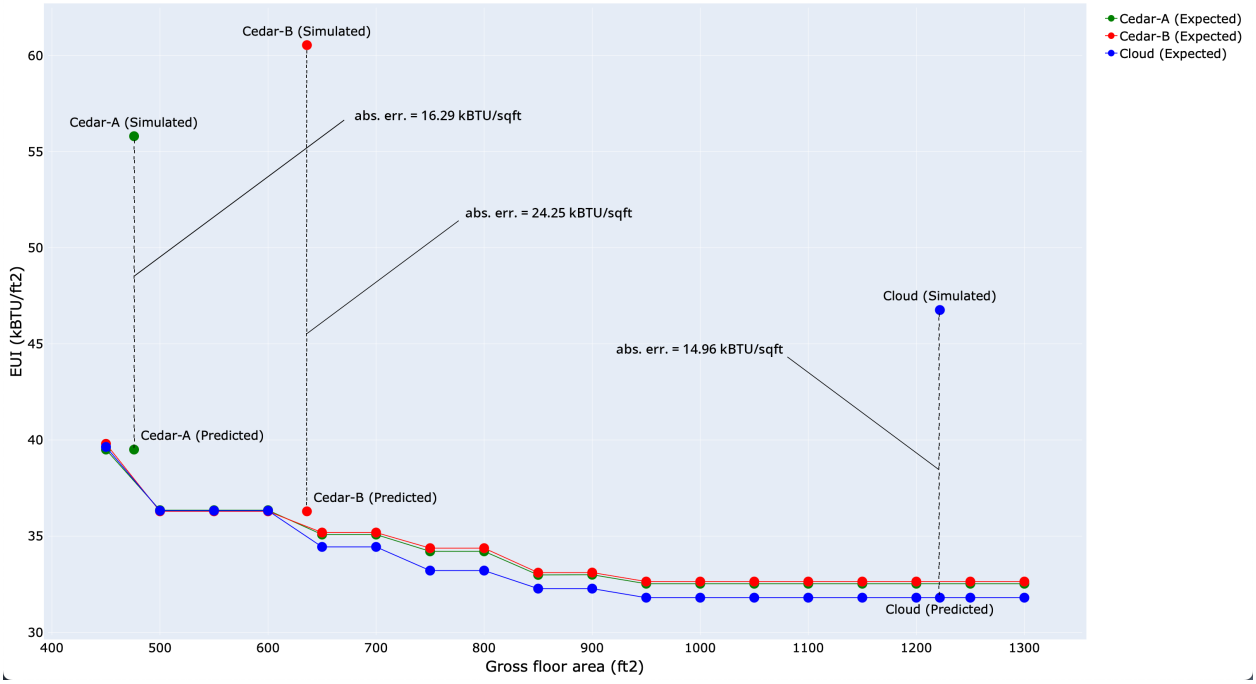


Figure 4.9: Comparison between simulated, predicted and expected EUI for each DADU

Results from simulation of the three DADUs, corresponding energy predictions, and surrounding, analogous prediction results were then plotted collectively in a scatter plot showing

gross floor area versus energy use intensity (Figure 4.9). Absolute errors of 16.29 kBTU/ft^2, 24.25 kBTU/ft^2, and 14.96 kBTU/ft^2 were calculated for the Cedar-A, Cedar-B, and Cloud DADUs respectively. Computing simple MAE returns a loss of 18.5 kBTU/ft^2, while the resultant RMSE comes in at 18.95 kBTU/ft^2. Both values are reflective of extremely high error, albeit with $n = 3$ samples. The Cedar-B and Cloud DADUs were quite similar in error at 16.29 kBTU/ft^2 and 14.96 kBTU/ft^2, respectively. Cedar-B received much worse results, missing the mark by 24.25 kBTU/ft^2. Most surprisingly, the absolute error between simulated and predicted EUI for the Cloud DADU is lower than both Cedar DADU variants, as the gross floor area of Cloud (1221 ft^2) is outside of the floor area range in which the XGBoost model was trained. Additionally, all predicted values beyond this upper bound of gross floor area the model has seen, became constant and equivalent to the prediction at 1000 ft^2. Therefore simplifying the Cloud DADU prediction by using a gross floor area of 1000 ft^2 in place of the 1221 square foot reality offered no recourse.

4.3.1 Energy model discrepancies

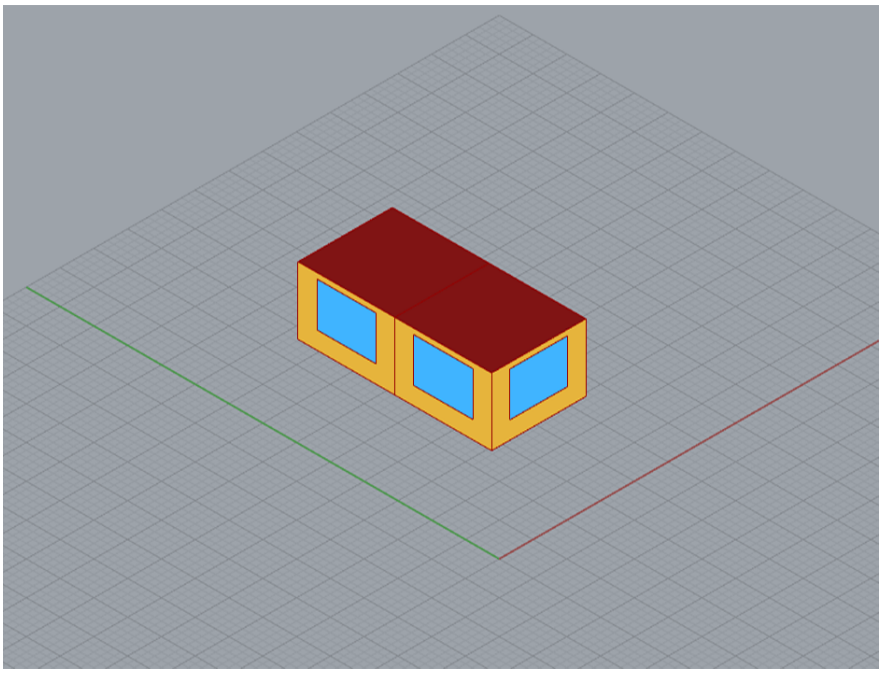


Figure 4.10: Typical Honeybee DADU model

Discrepancies between simulated and predicted energy use intensity values in the comparative analysis DADUs is likely due to differences in energy model representation. Various differ-

ences were identified before and during the analysis stage including roof type, general DADU form, and glazing distribution. Parametric Honeybee models created during the dataset simulation portion of this thesis research utilized what is referred to as a *shoebox model* , or a simple model reminiscent of a shoebox and lacking any formal movements. Comparing between Figure 4.10 and Figure 3.12, the differences in modeling become apparent. As part of this generalization and oversimplification, simulated DADU roofs were represented solely as flat roofing. Additionally, glazing distribution during the training data simulation stage was demonstrated as uniform in nature, losing any model specificity regarding views. During this comparative study, each was simulated using true roof pitch, with each being a form of shed roof. Furthermore, DADU forms were represented as their true extents, and not as simple boxes. Window placement was modeled as it exists, either in built form or as specified in the construction documents, rather than as an equivalent window-to-wall ratio on all facades. All three DADU models also include varying amounts of external shading, another feature omitted from the dataset simulation, with Cedar-A and Cedar-B including 98.86 ft^2 and 133.34 ft^2 of shading above each respective primary entry, while the Cloud DADU counted 123.31 ft^2 of external overhanging shading nearly equally in all directions. Ultimately, the culprit behind the discrepancies between simulated and predicted EUI for the DADUs in this analysis likely originated from these variations in 3D modeling and simulation.

4.4 Policy implications

Lastly, results generated from the web application were directed towards answering the pair of land use policy-oriented research questions outlined in Section 1. No discernible difference between existing and waived side yard setbacks in the rear yard were discovered. Web tool results for 5 pairs of DADUs holding all variables constant except gross floor area, which was set to 200, 400, 600, 800, and 1000 square feet were compared between the existing and proposed setbacks. Figure 4.11 clearly demonstrates the lack of impact on energy use intensity between different setback paradigms. In support of this information, Figure 4.5 further shows the negligible impact of setbacks on energy use intensity. It is worth noting that this is potentially due to methodology lacking more limiting criteria to simulate human choice: views, budgets, etc. Further, waiving side yard setbacks in addition to rear setbacks within the rear yard has been promoted as significant. Only a handful of minor tweaks

Figure 4.11: Setbacks versus EUI, varying gross floor area

to the parametric implementation of setbacks in the Grasshopper script could remedy and represent this scenario. Consequently however, the dataset in its entirety would need to be regenerated.

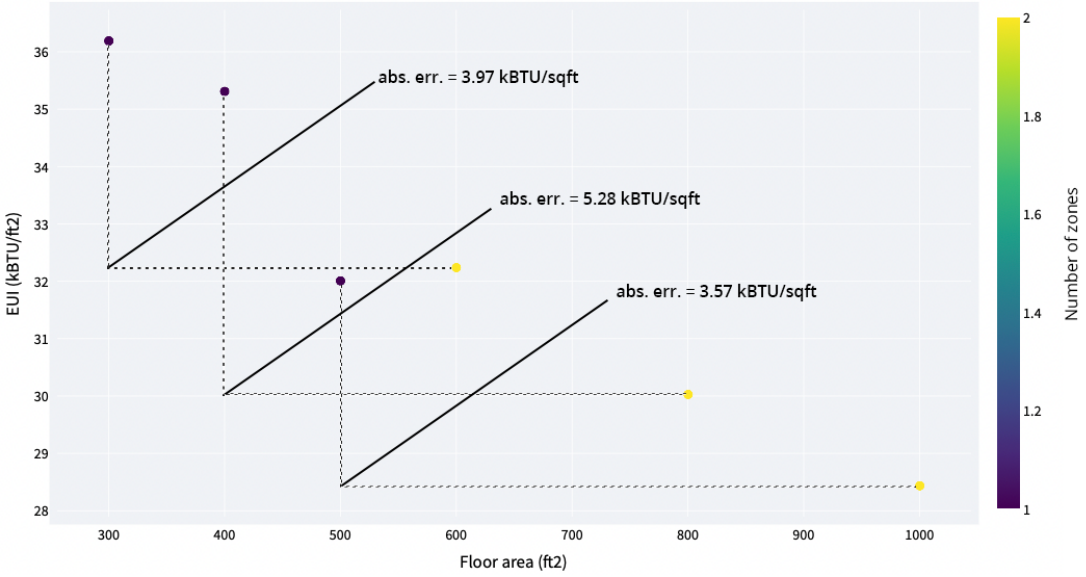


Figure 4.12: Gross floor area versus EUI, varying gross floor area

On the other hand, web application predictions into the impact sharing of walls between

DADUs suggest the hypothesis was in fact correct. This study compared the energy use intensity of 1-story, 2-zone DADUs of x floor area against 1-story, 1-zone DADUs of *0.5x* floor area with all other design features held constant. Results shown in Figure 4.12 display the 2-zone variants outperforming the single zone DADUs by an average reduction in EUI of 4.27 kBTU/ft^2. Yellow data points represent the 2-zone, shared wall DADUS while the dark blue points depict the 1-zone, *0.5x* area DADU variants. Dotted lines between points denote a related pair, with the associated absolute error shown. However, *SHAP value* impact corresponding to the number of zones, shown in Figure 4.5 does not do the difference in energy consumption justice. Outcomes here are aligned with expectations, as construction of a standalone DADU results in greater surface-area-to-volume ratio than if one were to build the DADU adjacent to a neighbor's existing accessory dwelling.

While the metamodel performs well under the limited design constraints used in training, the comparative analysis revealed the limitations of the model (i.e., roof types, variable WWRs, and overall massing configurations). Successful implementation of this or similar methodologies could become an integral part of the ADU and housing conversation around the United States. Prospective work moving forward can add quantitative data features and/or qualitative energy and 3D modeling improvements to more accurately predict energy use intensity in real-world examples, such as the CAST Architecture-designed DADUs used as a case study within the scope of this project.

Chapter 5

CONCLUSION

Throughout this research, the goal of reducing the energy use intensity of detached accessory dwelling units in Seattle, Washington was stressed and further explored through simulation, metamodeling and analysis. Predictions using the developed web application are instantaneous and accurate within ± 0.188 kBTU/ft^2. The hope is that putting forth this methodology can advance the discussion surrounding housing affordability, energy efficiency, and carbon not just in Seattle, but across the United States. Proposed methods in this research specifically will be of greater use for cities in regions with less temperate climates, more expensive electricity, and less decarbonized energy sources. In order to answer the three primary research questions outlined in Section 1.3 and to conclude this thesis, findings on the model accuracy, generalization, and ability to inform accessory dwelling unit policy will be specified.

Accuracy of the metamodel is of great relevance, as an inaccurate prediction is of no value. However, when considered within the context of metamodel development, inaccurate predictions play an important role as the accuracy and precision of metamodels are subjectively set by use-case and practitioner. For applications such as this, high scientific accuracy is not of utmost importance. Decisions made in the early architectural design stages tend to relate more to macro-scale formal moves, material selection, and the weighing of various design goals to balance. Decisions made here can be changed quickly and iteratively, as long as the project stays at this stage. Therefore, only a reasonable level of accuracy is truly required to inform designers whether their modifications are working in the right direction. On the other hand, homeowners are likely less concerned over absolute accuracy and are instead interested in how much glazing they can incorporate or the amount of money that can be saved by using certain materials. Again, with a testing error of the final metamodel of only 0.188 RMSE, we would label this model as accurate.

However, taking the comparative analysis results into account, the conclusion cannot be as simple as saying "the model is accurate". The original labeling in the previous paragraph requires a heavy dose of asterisks beside "accurate". Labeling the quality of the metamodel correctly, it should be referred to as "strict predictive accuracy", being quite accurate specifically when predicting on a DADU design similar to the dataset it was trained on. Normally, in the realm of machine learning, this condition would be cited as an example of overfitting. An argument can be made that it is, but not in the traditional sense. Instead, the model is hypersensitive to the modeling method used and level of detail expended to represent the simulated structures used to train the dataset. To conclude, the model fails at generalizing upon new data. Specifically, real world DADU designs that are distinct enough from the Honeybee shoebox models. It was believed through careful selection of the features to include in the design space generation that a corresponding metamodel would be suitable to generalize across all DADU designs. The most likely remedy is simply increasing the number of design features and samples used to train the metamodel.

Policy implications were successfully explored, albeit outcomes stemming from simulated changes to setbacks resulted in less differences than originally hoped. With one hypothesis disproved and the other supported, it is safe to assume that this method of incorporating land use policy questions into a metamodel has been successful. Moving forward, other ideas can be tested by adding additional features and increasing the size of the dataset. It is important to use the tool in this fashion, as Seattle comprehensive plan is enacted for 20 year periods. With climate change and housing affordability being such pressing issues, missing the opportunity this time around could be catastrophic.

5.1 Future work

The logical next step to improve this application is to expand the existing scope of predictions by adding new features of interest. Optionally, the dataset could instead be bolstered with more samples or simulations using the same set of features, with a greater degree of detail by increasing the granularity of simulations (i.e., 0.0-0.9 window-to-wall ratio in 0.01 increments as opposed to 0.1) Adding new features can be utilized to test novel policy or design ideas in a safe environment before pushing out to the world. In order to remedy failure to generalize upon new data, specific building features could be added, such as: simple building forms

(long, box, L-shaped, etc.), roof forms (flat, shed, pitched, etc.), and exterior shading (none, minimal, maximal). Accessory dwelling units are just one typology of subversive means to increase housing supply. Similar ideas exist under other names, both familiar and exotic, and can greatly benefit from energy prediction such as: town homes, cottage clusters, triplexes, etc. In order for this to work, one would need to simulate and train on a large set of building energy simulation examples similar to those mentioned above, likely while including the building, roof, and exterior shading features. In doing so, there would be a need to add more depth to the meaning and number of features representing building typology, rather than simply number of stories and energy zones.

Another route could be to improve the web application by adding additional functionality like map lookup, user modes, and the inclusion of embodied carbon. Map lookup refers to the ability to integrate *Google maps* or *OpenStreetMap* into the web application to allow users to search by address. The site data and lot information from the selected address would be incorporated into predictions by some means to improve the validity of results. In terms of user modes, one can imagine a toggle existing to swap between a designer mode and homeowner mode, for example. As mentioned in the introductory paragraph of this chapter, interests generally vary between homeowners and designers. Designer mode would presumably most closely resemble the current incarnation of the web tool, while the homeowner mode could simplify the data visualization experience and refine the means of communicating the energy and carbon metrics. On the topic of carbon, the inclusion of an embodied carbon metric would greatly benefit Seattle in particular. As the grid is essentially net zero carbon, the only real carbon emissions in question are due to creation, transportation, and implementation of construction materials. Interfacing with *EC3*, or a similar embodied carbon database would be required in order to keep values up to date.

Times are changing for architecture and housing in general. New paradigms are demonstrated frequently in terms of both technology and philosophy. Machine learning is a scary notion to most people, either out of confusion or from low-quality sci-fi movies. However, at its core, machine learning, data science, and related fields are beginning not to control, but to empower designers to create their best work. In the age of big data, these algorithms are the sole means for designers to interface with and make sense of this magnitude of

information, often real-time (Bottazzi 2019). Machine learning workflows are being used to better understand the housing market in order to inform equitable housing policies in China and in urban planning to understand the impacts of diverse metro information (Hu et al. 2019; Hanna 2020). Integration of design, data, and a focus on community and environment give me hope for the future. Designing these processes in holistic and even transdisciplinary ways can ensure that all stakeholders needs are met, as designers, developers, planners, and researchers all attempt to become the field that solves the housing crisis of this decade.

Bibliography

Administration, U.S. Energy Information. 2020. *How much carbon dioxide is produced per kilowatthour of U.S. electricity generation?* Technical report. U.S. Energy Information Administration. Accessed May 14, 2022. https://www.eia.gov/tools/faqs/faq.php?id=74&t=11.

Ahmad, Muhammad Waseem, Monjur Mourshed, and Yacine Rezgui. 2017. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy and Buildings* 147:77–89. ISSN: 0378-7788. https://doi.org/https://doi.org/10.1016/j.enbuild.2017.04.038. https://www.sciencedirect.com/science/article/pii/S0378778816313937.

Ahuja, Sandeep, and Patrick Chopson. 2020. Automation and machine learning in architecture: A new agenda for performance-driven design. Publisher: Wiley Online Library, *Architectural Design* 90 (2): 104–111.

Amasyali, Kadir, and Nora M. El-Gohary. 2018. A review of data-driven building energy consumption prediction studies. *Renewable and Sustainable Energy Reviews* 81:1192–1205. ISSN: 1364-0321. https://doi.org/https://doi.org/10.1016/j.rser.2017.04.095. https://www.sciencedirect.com/science/article/pii/S1364032117306093.

Bicknell Argerious, Natalie. 2020. *Seattle's Quest to Become a 15-Minute City.* Blog, September. Accessed May 15, 2022. https://www.theurbanist.org/2020/09/18/seattles-quest-to-become-a-15-minute-city/.

Big Ladder. 2014. *EnergyPlus File Extensions.* Accessed May 15, 2022. https://bigladdersoftware.com/epx/docs/8-2/getting-started/energyplus-file-extensions.html.

Binkovitz, Leah. 2019. *Gap between income growth and housing cost increases continues to grow.* https://kinder.rice.edu/urbanedge/2019/07/25/gap-between-income-growth-and-housing-cost-increases-continues-grow.

Botella, Elena. 2021. *Investment Firms Aren't Buying All the Houses. But They Are Buying the Most Important Ones.* https://slate.com/business/2021/06/blackrock-invitation-houses-investment-firms-real-estate.html.

Bottazzi, Roberto. 2019. Learning algorithms, design, and computed space. *Enquiry The ARCC Journal for Architectural Research* 16 (2): 6–17.

Brinig, Margaret F., and Nicole Stelle Garnett. 2013. A room of one's own? Accessory dwelling unit reforms and local parochialism. Publisher: American Bar Association, *The Urban Lawyer* 45 (3): 519–569. ISSN: 00420905, 19426593. http://www.jstor.org/stable/24392672.

Brownlee, Jason. 2018. *A Gentle Introduction to k-fold Cross-Validation,* May. Accessed May 26, 2022. https://machinelearningmastery.com/k-fold-cross-validation/.

Buylova, Alexandra. 2020. Spotlight on energy efficiency in Oregon: Investigating dynamics between energy use and socio-demographic characteristics in spatial modeling of residential energy consumption. *Energy Policy* 140:111439. ISSN: 0301-4215. https://doi.org/https://doi.org/10.1016/j.enpol.2020.111439. https://www.sciencedirect.com/science/article/pii/S0301421520301920.

Casini, Marco. 2022. Chapter 5 - Building performance simulation tools. In *Construction 4.0,* edited by Marco Casini, 221–262. Woodhead Publishing Series in Civil and Structural Engineering. Woodhead Publishing. ISBN: 978-0-12-821797-9. https://doi.org/https://doi.org/10.1016/B978-0-12-821797-9.00004-0. https://www.sciencedirect.com/science/article/pii/B9780128217979000040.

CAST Architecture. 2021. *Backyard Cottages.* https://www.castarchitecture.com/backyard-cottages-main.

Chapple, Karen, Dori Ganetsos, and Emmanuel Lopez. 2021. *Implementing the Backyard Revolution: Perspectives of California's ADU Owners,* April. https://www.aducalifornia.org/wp-content/uploads/2021/04/Implementing-the-Backyard-Revolution.pdf.

Chapple, Karen, Jake Wegmann, Farzad Mashhood, and Rebecca Coleman. 2017. *Jumpstarting the Market for Accessory Dwelling Units: Lessons Learned from Portland, Seattle, and Vancouver.*

Chen, Tianqi, and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, August. https://doi.org/10.1145/2939672.2939785. https://doi.org/10.1145%2F2939672.2939785.

Chen, Yi-Wei, and Chih-Jen Lin. 2006. *Combining SVMs with Various Feature Selection Strategies.* https://www.csie.ntu.edu.tw/~cjlin/papers/features.pdf.

Chugh, Akshita. 2020. *MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better?,* December. Accessed May 23, 2022. https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e.

City of Seattle. 2019a. *ADUniverse: ADU Rules.* https://aduniverse-seattlecitygis.hub.arcgis.com/pages/2135866cc0ea4465ae1aea49de3860d1.

———. 2020. *ADUniverse: Pre-approved ADU Designs.* https://aduniverse-seattlecitygis.hub.arcgis.com/pages/gallery.

———. 2019b. *Pre-approved Plans for Accessory Dwelling Units Survey Results,* November. http://www.seattle.gov/Documents/Departments/OPCD/OngoingInitiatives/EncouragingBackyardCottages/OPCDPreApprovedDADUSurveySummary.pdf.

———. 2021. *Seattle 2021 ADU Annual Report (Interactive).* https://storymaps.arcgis.com/stories/f8ee6480b1764b1bab219beec38b2d88.

Cobb, Rodney L., and Scott Dvorak. 2000. *Accessory Dwelling Units: Model State Act and Local Ordinance.* Technical report. AARP. http://chhoa.metamediacom.com/wp-content/uploads/ARRP-Model-State-Local-Ordinance-d17158_dwell.pdf.

Colburn, Gregg, and Clayton Page Aldern. 2022. *Homelessness is a Housing Problem.* University of California Press. ISBN: 978-0-520-38376-0.

Cross, Nigel. 2001. Can a Machine Design? Publisher: The MIT Press, *Design Issues* 17 (4): 44–50. ISSN: 07479360, 15314790, accessed February 12, 2021. http://www.jstor.org.offcampus.lib.washington.edu/stable/1511919.

D'Souza, Jocelyn. 2018. *A Quick Guide to Boosting in ML.* Accessed May 24, 2022. https://medium.com/greyatom/a-quick-guide-to-boosting-in-ml-acf7c1585cb5.

Eisenbach. 2022. *Electricity Rates in the United States.* https://www.electricchoice.com/electricity-prices-by-state/.

Fesler, Stephen. 2018. *Nearly Six Million Residents in Central Puget Sound By 2050, PSRC Says,* January. https://www.theurbanist.org/2018/01/26/nearly-six-million-residents-puget-sound-area-2050-psrc-says/.

Flournoy, Edward Brian. 2021. The rising of systemic racism and redlining in the United States of America. *Journal of Social Change* 13 (1): 6.

Forrest, Adam. 2019. *Vienna's Affordable Housing Paradise.* Blog. https://www.huffpost.com/entry/vienna-affordable-housing-paradise_n_5b4e0b12e4b0b15aba88c7b0.

Gebhardt, Matthew, Beth Gilden, and Yael Kidron. 2018. Accessory Dwelling Units in Portland, Oregon.

Granderson, Jessica, Samir Touzani, Claudine Custodio, Michael Sohn, David Jump, and Samuel Fernandes. 2016. Accuracy of automated measurement and verification (M&V) techniques for energy savings in commercial buildings. *Applied Energy* 173 (July): 296–308. https://doi.org/10.1016/j.apenergy.2016.04.049.

Grillone, Benedetto, Stoyan Danov, Andreas Sumper, Jordi Cipriano, and Gerard Mor. 2020. A review of deterministic and data-driven methods to quantify energy efficiency savings and to predict retrofitting scenarios in buildings. *Renewable and Sustainable Energy Reviews* 131:110027. ISSN: 1364-0321. https://doi.org/https://doi.org/10.1016/j.rser.2020.110027. https://www.sciencedirect.com/science/article/pii/S136403212030318X.

Guo, Shuai. 2020. *An introduction to Surrogate modeling, Part I: fundamentals.* https://towardsdatascience.com/an-introduction-to-surrogate-modeling-part-i-fundamentals-84697ce4d241.

Hanna, Sean. 2020. Further perspectives data in design practice. Publisher: JSTOR, *Design Transactions: Rethinking Information Modelling for a New Material Age,* 142.

Hertz, Daniel. 2015a. *The immaculate conception theory of your neighborhood's origins,* September. https://cityobservatory.org/the-immaculate-conception-theory-of-your-neighborhoods-origins/.

———. 2015b. *What Filtering Can and Can't Do.* Blog. https://cityobservatory.org/what-filtering-can-and-cant-do/.

Housable. 2021. *Sacramento ADUs.* https://www.housable.com/city/sacramento-ca-179074.

Hu, Lirong, Shenjing He, Zixuan Han, He Xiao, Shiliang Su, Min Weng, and Zhongliang Cai. 2019. Monitoring housing rental prices based on social media:An integrated approach of machine-learning algorithms and hedonic modeling to inform equitable housing policies. *Land Use Policy* 82:657 –673. ISSN: 0264-8377. https://doi.org/https://doi.org/10.1016/j.landusepol.2018.12.030. http://www.sciencedirect.com/science/article/pii/S0264837718316429.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An introduction to statistical learning.* Vol. 112. Springer.

Knippers, Jan, Bob Sheil, and Mette Ramsgaard Thomsen. 2020. Innochain: A Template for Innovative Collaboration [in English]. In *Design Transactions,* edited by Bob Sheil, Mette Ramsgaard Thomsen, Martin Tamke, and Sean Hanna, 14–21. UCL Press. https://doi.org/10.2307/j.ctv13xprf6.4.

Koehrsen, Will. 2018. *A Conceptual Explanation of Bayesian Hyperparameter Optimization for Machine Learning.* Accessed May 23, 2022. https://towardsdatascience.com/a-conceptual-explanation-of-bayesian-model-based-hyperparameter-optimization-for-machine-learning-b8172278050f.

Kontokosta, Constantine E., Vincent J. Reina, and Bartosz Bonczak. 2020. Energy Cost Burdens for Low-Income and Minority Households. Publisher: Routledge, *Journal of the American Planning Association* 86, no. 1 (January): 89–105. ISSN: 0194-4363. https://doi.org/10.1080/01944363.2019.1647446. https://doi.org/10.1080/01944363.2019.1647446.

Lei, Lei, Wei Chen, Bing Wu, Chao Chen, and Wei Liu. 2021. A building energy consumption prediction model based on rough set theory and deep learning algorithms. *Energy and Buildings* 240:110886. ISSN: 0378-7788. https://doi.org/https://doi.org/10.1016/j.enbuild.2021.110886. https://www.sciencedirect.com/science/article/pii/S0378778821001705.

Levy, Susie, and Aly Pennucci. 2019. *ADUs - policies and racial equity analysis.* http://www.seattle.gov/Documents/Departments/Neighborhoods/MajorInstitutions/Presentation20190304.pdf.

Logsdon, Michael, and Ben Larson. 2016. Assessment of Ductless Mini-Split Heat Pump Energy Savings in Stack House Apartments.

Lundberg, Scott M, and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30,* edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 4765–4774. Curran Associates, Inc. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

McKinney, Wes. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference,* edited by Stéfan van der Walt and Jarrod Millman, 56 –61. https://doi.org/10.25080/Majora-92bf1922-00a.

Metcalf, Gabriel. 2018. Sand castles before the tide? Affordable housing in expensive cities. Publisher: American Economic Association, *The Journal of Economic Perspectives* 32 (1): 59–80. ISSN: 08953309. http://www.jstor.org/stable/26297969.

Miller, Clayton, Liu Hao, and Chun Fu. 2022. *Gradient boosting machines and careful preprocessing work best: ASHRAE Great Energy Predictor III lessons learned.* ArXiv: 2202.02898 [cs.LG].

Mohler, Rick, Nick Welch, Joseph Hellerstein, Emily Finchum-Mason, Niu Yuanhao, Adrian Tullock, and Anagha Uppal. 2019. *ADUniverse Tool - eScience Institute, University of Washington.* https://uwescience.github.io/ADUniverse/.

Mueller, Caitlin. 2014. *Computational Exploration of the Structural Design Space,* June.

Mueller, Caitlin, and John Ochsendorf. 2013. *An Integrated Computational Approach for Creative Conceptual Structural Design.*

Pant, Ayush. 2019. *Introduction to Linear Regression and Polynomial Regression.* Blog. https://towardsdatascience.com/introduction-to-linear-regression-and-polynomial-regression-f8adc96f31cb.

Quirindongo, Rico. 2021. *Encouraging Backyard Cottages.* Accessed May 15, 2022. https://www.seattle.gov/opcd/ongoing-initiatives/encouraging-backyard-cottages.

Rokach, Lior, and Oded Maimon. 2014. *Data Mining with Decision Trees.* 2nd. _Eprint: https://www.worldscientific.com/doi/pdf/10.1142/9097. WORLD SCIENTIFIC. https://doi.org/10.1142/9097. https://www.worldscientific.com/doi/abs/10.1142/9097.

Routhier, Giselle. 2021. *State of the Homeless 2021.* https://www.coalitionforthehomeless.org/wp-content/uploads/2021/04/StateOfTheHomeless2021.pdf.

Sadeghipour Roudsari, Mostapha, and Michelle Pak. 2013. Ladybug: A Parametric Environmental Plugin for Grasshopper to Help Designers Create an Environmentally-Conscious Design, accessed May 15, 2022. https://www.ibpsa.org/proceedings/bs2013/p_2499.pdf.

Samuel, A. L. 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3 (3): 210–229. https://doi.org/10.1147/rd.33.0210.

Schreck, Ben. 2018. *Feature Engineering vs Feature Selection,* January. Accessed May 23, 2022. https://innovation.alteryx.com/feature-engineering-vs-feature-selection/.

Sheil, Bob, Jane Burry, Jenny Sabin, and Marilena Skavara. 2020. From Making Digital Architecture to Making Resilient Architecture. In *Fabricate 2020,* 12–19. Making Resilient Architecture. UCL Press. ISBN: 978-1-78735-812-6, accessed October 25, 2021. https://doi.org/10.2307/j.ctv13xpsvw.6. http://www.jstor.org/stable/j.ctv13xpsvw.6.

Sra, S., S. Nowozin, and S.J. Wright. 2012. *Optimization for Machine Learning.* Neural information processing series. MIT Press. ISBN: 978-0-262-01646-9. https://books.google.com/books?id=JPQx7s2L1A8C.

Tamke, Martin, Paul Nicholas, and Mateusz Zwierzycki. 2018. Machine learning for architectural design: Practices and infrastructure [in en]. *International Journal of Architectural Computing* 16, no. 2 (June): 123–143. ISSN: 1478-0771, 2048-3988, accessed October 25, 2021. https://doi.org/10.1177/1478077118778580. http://journals.sagepub.com/doi/10.1177/1478077118778580.

Tekin, Eylul. 2021. *A Timeline of Affordability: How Have Home Prices and Household Incomes Changed Since 1960?* Blog, August. https://listwithclever.com/research/home-price-v-income-historical-study/.

Thomalla, Frank, Tom Downing, Erika Spanger-Siegfried, Guoyi Han, and Johan Rockström. 2006. Reducing hazard vulnerability: towards a common approach between disaster risk reduction and climate adaptation. Publisher: Wiley Online Library, *Disasters* 30 (1): 39–48.

Touzani, Samir, Jessica Granderson, and Samuel Fernandes. 2018. Gradient boosting machine for modeling the energy consumption of commercial buildings. *Energy and Buildings* 158:1533–1543. ISSN: 0378-7788. https://doi.org/https://doi.org/10.1016/j.enbuild.2017.11.039. https://www.sciencedirect.com/science/article/pii/S0378778817320844.

Trevisan, Venicius. 2017. *Using SHAP Values to Explain How Your Machine Learning Model Works,* January. https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137.

United States Environmental Protection Agency (EPA). 2022. *Emissions & Generation Resource Integrated Database (eGRID), 2020.* Technical report. Washington, DC: EPA Office of Atmospheric Programs. Accessed May 16, 2022. https://www.epa.gov/egrid.

US Department of Energy. 2021a. *EnergyPlusTM 9.5.0 Essentials.* https://energyplus.net/assets/nrel_custom/pdfs/pdfs_v9.5.0/EnergyPlusEssentials.pdf.

US Department of Energy. 2021b. *EnergyPlusTM Version 9.5.0 Documentation - Engineering Reference,* March. https://energyplus.net/sites/all/modules/custom/nrel_custom/pdfs/pdfs_v9.5.0/EngineeringReference.pdf.

———. 2021c. *EnergyPlusTM Version 9.6.0 Documentation: Input Output Reference.* https://energyplus.net/assets/nrel_custom/pdfs/pdfs_v9.6.0/InputOutputReference.pdf.

Wang, Yuanchao, Z. Pan, J. Zheng, L. Qian, and Li Mingtao. 2019. A hybrid ensemble method for pulsar candidate classification. *Astrophysics and Space Science* 364 (August). https://doi.org/10.1007/s10509-019-3602-4.

Welch, Nick, Mike Podowski, Jennifer Pettyjohn, Scott Domansky, and Bernardo Serna. 2021. *Accessory Dwelling Units Annual Report 2021.* http://www.seattle.gov/Documents/Departments/OPCD/OngoingInitiatives/EncouragingBackyardCottages/OPCD_ADUAnnualReport2021.pdf.

Winkler, Jon. 2011. *Laboratory test report for Fujitsu 12RLS and Mitsubishi FE12NA mini-split heat pumps.* Technical report. National Renewable Energy Lab.(NREL), Golden, CO (United States).

XGBoost Developers. 2021. *XGBoost Docs - Parameters.* https://xgboost.readthedocs.io/en/stable/parameter.html.

Zhong, Hai, Jiajun Wang, Hongjie Jia, Yunfei Mu, and Shilei Lv. 2019. Vector field-based support vector regression for building energy consumption prediction. *Applied Energy* 242:403–414. ISSN: 0306-2619. https://doi.org/https://doi.org/10.1016/j.apenergy.2019.03.078. https://www.sciencedirect.com/science/article/pii/S0306261919304878.