Design Document: Predicting Energy use of a Residential Building in the United States

Casey Owen, Casey.Owen@tufts.edu

I. Introduction

Reducing energy use in homes in the residential sector is a crucial goal because of the wide array of benefits it provides – cost savings, emissions reductions, and improving grid resiliency. Perhaps most obvious is cost savings: the average American household spends more that $2,200 per year on energy bills, with nearly half of this going to heating and cooling costs [1]. Reducing residents' energy bills can provide additional economic freedom while lowering financial stress. Producing this power, whether it be electricity, natural gas or another fuel, also creates carbon emissions which accelerate climate change. Overall, roughly 20% of American energy-related greenhouse gas emissions stem from heating, cooling, and powering households [2]. Reducing energy use also reduces stress on the electric grid during peak periods, minimizing the risk of power outages and disruptions, thereby improving public safety [3].

This project focuses on educating homeowners about what energy efficiency project is right for them as a tool to reduce energy use. These projects can range from something as simple as replacing an appliance with a more efficient version, to as complicated as completely revamping their Heating, Ventilation, and Air Conditioning (HVAC) systems. Many homeowners without professional experience in HVAC may not know where to start.

Today, homeowners have the option to get an energy audit – hiring a professional to walk through the home and assess it's energy use, suggesting possible issues and upgrades. While this method may be the most accurate and personalized available, it can be expensive – the average home energy audit cost is $425, depending on the size of the home [4]. This figure can be hard to commit to, especially when the amount of potential money saved is unknown.

This project intends to create a free tool for homeowners that assists them in finding possible energy projects for their home. The user will enter the general location and basic information about their home, and receive a list of relevant measures they could pursue, along with quantitative estimates of energy, emissions, and money saved. This tool is not intended as a replacement for an energy audit, but rather as a tool to help the homeowner gather information about their home and learn about their options so they can make more educated decisions.

Additionally, this tool will leverage weather data in its predictions. Because of this, when the assumed weather conditions change, the estimated energy use will change. Using publicly available weather data based on climate models that incorporate climate change, the user have the option of entering a desired 20-year period anytime in the 21st century to see the how their projected energy use and viable projects change as a result.

This design document describes the machine learning model and methods that will be built to leverage the National Renewable Energy Laboratory (NREL) ResStock™ dataset [5] to predict an individual home's energy use and emissions.

II. Data Sources

A. ResStock™

The ResStock™ dataset, developed by the NREL for the U.S. Department of Energy, was created to help states, municipalities, utilities, manufacturers, and researchers understand the energy use of the U.S. residential building stock. [6] This publicly-available dataset makes two significant contributions to understanding American residential buildings – generating a representative building stock of the country with building features, and then running them through physics-based energy simulations.

First, NREL collected information on the building stock, including conditional probability distributions of building characteristics [7]. Then, the dataset samples from these distributions to create 2.2 million representative building models, which can be thought of as fictional buildings. None of these buildings are based exactly off of real buildings, matching their precise features, but the population of buildings are designed to resemble the stock of residential buildings. Some examples of features of these buildings are characteristics such as floor area in square feet, number of bedrooms, what HVAC equipment is installed, and insulation levels in various locations. Over 150 individual characteristics are included in total.

Second, these fictional buildings were passed into industry-standard energy modeling tools OpenStudio® and EnergyPlus® in order to determine their total energy use based on software physics simulations. These simulations also generate end-use load profiles which shows how much each individual energy-user in the home (air conditioners, washing machine, furnace, etc.) has contributed to the total energy use over time. Early versions of this dataset performed this task only on baseline energy use, which refers to how much energy the buildings consume without any changes. These results were then validated against actual energy uses of similar real buildings, and the results calibrated to fix discrepancies with modeling. In later versions of the dataset, additional "what-if" scenarios were modeled, where potential changes to the building were considered using the application of measure packages. Measure packages are sets of possible measures. For example, one possible measure package might include two measures - one for replacing the lights with LEDs, and one for improving home insulation.

The baseline and "what-if" scenario results will be the primary data sources for this project. They will allow the user to predict their home's energy use and how much money they could save from various measures.

B. fTMY3 Weather Data

An important input to any energy model is the weather. Many energy uses, especially heating and cooling, are extremely dependent on the weather. One possible option when modeling is to use historical weather data, but that comes with the choice of what historical year to pick (if modeling energy use over a year). If an atypically hot year is chosen, cooling energy use will be over-represented, and vice versa, if an atypically cold year is chosen, heating energy use will be over-represented.

The TMY (Typical Meteorological Year) datasets were created to solve this problem, so that energy modelers could use a typical year in their models. These datasets, also created by the NREL, consider a period of historical weather which consists of 68 data fields of meteorological properties, collected at hourly intervals, spanning approximately 30 years. An algorithm is used to find the most the most typical of each month in the 30-year period. [8] For example, the algorithm first looks through all 30 Januarys at a given location, determines which one is “most typical” using statistical methods, and selects that January for use in the TMY dataset. This is then repeated for all 12 months and the results are concatenated to form the most typical year, with smoothing applied at the month interfaces.

The original TMY dataset was created using historical data from 1952-1978, and the third and most current version, called TMY3, uses years 1976-2005. This TMY3 data is used as a standard through the energy modeling industry, and is what was used as inputs to the simulations of the ResStock™ dataset. However, typical weather data is not static over long periods of time, and homeowners are likely more interested in how much energy their homes will consume in the future rather than in weather conditions in the past. This creates the need for fTMY3 data – future TMY3 data.

This project will utilize an fTMY3 dataset created by Chowdhury et al [9]. This publication combines the publicly available TMY algorithm and the combined results of six selected global climate models to create TMY data at six different 20-year periods - 1980-1999, 2000-2019, 2020-2039, 2040-2059, 2060-2079, and 2080-2099. This was done for every county in the continental United States. This data integrates the projected impact of climate change into weather data that can be used for building simulations and models.

III. Methods

This project will use machine learning models to predict the energy that an individual home uses given a 20-year period, and some user-entered features about their home. The output of the models will be available as a free website tool, that gives both baseline energy use, and the savings results of several pre-selected measure packages. Energy use will be provided as both electricity use, and non-electricity energy use.

A. Data Preprocessing

The source of raw data for training the model that predicts energy use is the ResStock™ dataset that contains each modeled building, along with its features and various outputs. There are 2.2 million buildings, each with 186 features and 114 outputs. The outputs include the breakdown of energy use by specific fuel and equipment, but for the purposes of this project I am only interested in two – total electricity use and total non-electricity energy use, both in kWh. Non-electricity use will primarily be heating fuel such as natural gas or fuel oil - it is assumed that the user will know what heating fuel they use, if not electricity, to interpret this number correctly.

Data cleaning is done using the python library pandas. The first component is to step irrelevant columns. These include metadata and features that, based on my domain knowledge, will have a minimal effect on energy use. Some columns then require splitting and renaming, if they include multiple important properties that will be specified separately.

Then, all categorical features must be encoded, which in this case is almost all features. I have manually determined which features are ordinal vs. nominal, and specified an appropriate encoding for each. Most nominal variables receive binary encoding since cardinality of approximately 5-10 unique values for each feature is common, and this creates many fewer additional columns than one-hot encoding. Ordinal encodings are relatively straightforward for most of these columns – for example a heating setpoint of 60 °F, 65 °F, or 70 °F correspond to values of 60, 65 and 70. Although, there are some that require more hand-crafted solutions. After this, Principal Component Analysis (PCA) is applied to reduce the dimensionality of the matrix. A MinMaxScaler is then applied to bound the columns to be between 0 and 1.

B. Incorporating Weather Data

The weather data for the given simulation must then be added to the training set. For training, the weather data used is the TMY3 dataset that the building simulations used as inputs. This is included with the other ResStock™ data. Custom aggregate statistics are then calculated for each county in the continental U.S. and added as features for each building in the larger table, prior to PCA and scaling. Since these values are all numerical, they do not need to be encoded.

Table 1: Custom Weather Data Aggregates

Annual Weather Data Aggregates			
Average Temperature	Average Relative Humidity	Average Wind Speed	Maximum Direct Normal Radiation
Maximum Temperature	Average Dew Point	Maximum Wind Speed	Average Direct Normal Radiation
Minimum Temperature	Maximum Dew Point	Maximum Global Horizontal Radiation	Maximum Diffuse Horizontal Radiation
Total Heating Degree Days (HDD)	Minimum Dew Point	Average Global Horizontal Radiation	Average Diffuse Horizontal Radiation
Total Cooling Degree Days (CDD)			

These features were chosen to be most representative of the factors that affect a building's energy use. They are all related to the transfer of heat between a building and its environment, which impacts the HVAC system. Heating Degree Days (HDD) and Cooling Degree Days (CDD) serve as the most direct measure of how much heating and cooling is needed in a given year, respectively. The formulas for each are:

$$HDD = Max\left(0, \frac{65\,°F - hourly\ temperature}{24}\right) \qquad CDD = Max\left(0, \frac{hourly\ temperature - 65\,°F}{24}\right)$$

When these are then summed across a year of hourly temperature data, they provide a proxy for how much total heating and cooling is needed over the course of a year. This is based on the assumption that no conditioning is required at an outdoor temperature of 65 °F, and the load scales linearly with temperature from that point.

When performing model inference, a similar procedure will be carried out. The same aggregate values will be calculated, this time from the fTMY3 data file created by Chowdhury et al [9], corresponding to the user's selected county and 20-year period.

C. Modeling

The machine learning algorithm I chose to use in this project is XGBoost, a popular implementation of the gradient boosting algorithm. Gradient boosting describes a type of ensemble learning where a series of simple, shallow decision trees are trained to predict the error left over by the previous tree. Only the first tree is trained on the output data directly.

XGBoost is known for its high accuracy in interpolation and fast inference times on large, tabular datasets with complex interactions in both regression and classification. That describes this project's data very well, except for when extrapolating to unseen weather conditions. See a discussion of this issue on page 9.

I also considered using a random forest regressor, as well as other gradient boosting implementations such as LightGBM and CatBoost, but ultimately settled on XGBoost due to its reputation for speed, accuracy, and memory efficiency.

Several models will be trained overall. For each measure, two models will be trained – one that predicts electricity use, and one that predicts non-electricity energy use (natural gas and/or fuel oil). For example, if there are 10 measures + the baseline, and each requires two models, then 22 models will be created.

a. Training

The models will be trained on the High-Performance Computing (HPC) Cluster at Tufts, which provides large-scale computer resources.

To find the optimal model, hyperparameter tuning will be performed using random search cross validation. This strategy consists of randomly selecting hyperparameters from a grid of possible values, such a learning rate and maximum tree depth, and performing cross validation on each created model.

The performance metric to be used to select the best model will be Root Mean Square Error (RMSE), which will be the model that consistently makes the best predictions. It is suitable for this task since the output is a continuous numeric value, and predictions that are very far away from the target should be significantly penalized.

b. Inference

When performing model inference, the user must provide the county of their home, and may provide as many features about their home as they know and want to provide. However, there are over 150 possible features to choose from, and it is unreasonable to ask the user to enter a value for all of these. This presents an issue – something must be assumed for all remaining values for the model to calculate a result.

One solution is to simply use the mean or mode of all remaining features. This is not a very accurate strategy. Not only may this home be very far away from the user's home, but the assumed values combined may create an impossible home. For example, if the average home has 1 story, but the user has selected that they have an attic, the resulting combination of these will create an impossible house – a 1 story house with an attic. Since there will be no examples of a home like this in the training data, the model may not return reasonable results in this case.

An alternative solution, which I have selected to implement, is to filter the existing dataset by all provided features, and then sampling a handful of these houses to feed into the model. One advantage of this is that it all houses that are input to the model are "possible" houses. Another advantage is that it allows the website to provide the user with a range of possible energy values that people with similar homes might experience, so as to provide them with a degree of uncertainty in the results. The website can then report the median, 25^{th} percentile, and 75^{th} percentile results to the user. The sample size will be chosen to balance computation time and accuracy – it will be at least 5-10 homes, to reduce the change of the sample being entirely outliers.

Using this sampling strategy provides a third benefit – accounting for varying emissions and utility rates.

There are standard conversion factors that can be assumed to convert energy use to carbon emissions and energy cost, but these one-size-fits-all conversion factors have limited accuracy. Every home will have a slightly different utility rate due to regional energy prices. Also, every home will have slightly different emissions per unit of electricity due to electric grid composition. For example, homes served by electric grids with a greater concentration of renewable energy sources will contribute fewer emissions from 1 kWh of electricity use than homes served by a higher-emitting electric grid.

Fortunately, these varying factors are also included in the ResStock™ dataset. They will be accounted for at inference time by taking the average of the sampled homes.

fTMY3 data at the matching county will be added to the samples before being input to the model. Inference will be performed on all models at runtime. Computation power for all of these models will be a significant consideration – for a full discussion of this limitation, see page 8.

D. Included Measures

The table below shows the measures intended to be included. This list was chosen to give a wide variety of types of projects that can be done.

Table 2: List of Measures to be Modeled

Measures			
Number	ResStock™ Reference Package	Name	Description
1	2021.1.1	Basic Enclosure Package	Attic floor insulation, general air sealing, duct sealing, and drill-and-fill insulation into uninsulated wood walls
2	2021.1.2	Enhanced Enclosure Package	Measure 1 + interior insulation to foundation walls, conditioned basements and crawlspaces, insulate finished attics
3	2021.1.4	Heat Pumps, High-Efficiency, Electric Backup	Install high efficiency heat pump for heating
4	2021.1.6	Heat Pump Water Heaters	Replace water heater with heat pump water heater
5	2021.1.8	Whole-Home Electrification, High Efficiency	Convert all non-electric equipment to high efficiency electric – heat pump, heat pump water heater, heat pump dryer, electric range
6	2021.1.10	Whole-Home Electrification, High Efficiency + Enhanced Enclosure Package	Measures 2 and 5
7	2024.2.2.02	Windows, ENERGY STAR	Replace existing windows with ENERGY STAR windows for any less-efficient windows.
8	2024.2.3.07	Lighting, Universal LEDs	Replaces existing incandescent or CFL lighting with LED lighting
9	2024.2.4.05	Replace Natural Gas Furnace with 96% Efficiency	Replace existing natural gas furnace with high-efficiency natural gas furnace
10	2024.2.4.08	Replace Natural Gas Boiler with 96% Efficiency	Replace existing natural gas boiler with high-efficiency natural gas boiler

IV. Technologies and Tools

A. Technologies Used

Programming Languages: Python, Javascript, PHP, MySQL
Software Tools: Docker, Github
Libraries and Frameworks: xgboost, scikit-learn, flask, pickle, pandas
Cloud Platform: Google Cloud Run

B. Technology Pipeline

The ResStock™ data files are all available online, and can be accessed by querying the appropriate URLs. A Python script will be written to load this data into a dataframe and train the models. The model-training script will be run on Tufts HPC, and will pickle the resulting models and save them to files.

Inference will be performed by a Flask application, which is a web framework for Python. The Flask application will create an API endpoint, that, when queried, loads the correct models from file, and calculates and returns the results with the given parameters. This application will be Dockerized so that it can more easily be hosted on the hosting service, Google Cloud Run.

The website front-end will be created using JavaScript. JavaScript will be used to create a user interface where the user can enter information about their home, and see the model results. The JavaScript Fetch API will be used to query the Flask application based on the user's input. If server-side code is necessary, it will be written in PHP and likely use MySQL to store relevant data.

C. Website Interface

The website interface will consist mainly of home features that the user can enter. Questions will be mostly multiple-choice, fundamental questions about one's home that are generally knowable by people without professional expertise in residential equipment. Some example questions may include:

- How many bedrooms is your house?
- How many square feet is your house?
- Is your cooking range electric or gas?
- Does your home have a central air-conditioning system, or room air conditioners?
- Do you heat your home with a heat pump, natural gas, fuel oil, or another type of electric heat?

It will be explained to the user that they do not have to select an answer to any question they do not know – they will get results regardless. Although, the more information they enter, the more accurate their results will be. This is a result of the sampling strategy used in inference, and is an important function of the website since the target audience is laypeople who are just getting started and may not know every answer. The only mandatory information that must be provided is the user's county, since this is something all users will know, is very predictive of energy use, and is required to sample homes in the correct region.

The outputs of the site will be interactive graphics comparing the results of the given measures, and showing the user the range of outcomes for each measure. It will also show a graphic of the user's projected baseline energy use throughout the 21st century.

V. Limitations

A. Inference time

Likely the most significant limitation that will affect this project is the processing time required at inference. Training will also be a significant workload, but since it will happen offline ahead of time, over the course of weeks with GPU resources available, it is unlikely to be a bottleneck for the projects viability. Due to the high number of models created (approximately 22, each running on 5-10+ data points), a significant amount of computation will be required for inference.

There are a variety of trade-offs available that will allow this inference time to be reduced, but all have drawbacks:

a. Reduce the number of columns with PCA. As part of the PCA algorithm, the user can set how many feature dimensions they would like to keep – the lower this number the greater the computational efficiency will be due to having fewer features. This is an effective strategy since it will lose only the least important columns at explaining the data's variation. However, since this is a lossy algorithm, information will be lost and the models will become less accurate.

b. Reduce the maximum depth of the gradient boosting algorithm. Gradient boosting uses a series of trees, each trained off the residual errors of the previous tree. The deeper the trees that are used, the more complex the model – one side effect of this is longer inference time. Keeping the models small will improve inference time, but will also make them simpler, which may decrease accuracy.

c. Use more compute resources. This may include allocating more CPUs to perform the inferences in parallel, or using GPU-accelerated API hosting. This comes primarily at a monetary cost, and may be used in the final version only.

If loading time is still lengthy after these changes, simply providing the user with an informative progress bar on the loading screen would be some benefit.

B. Extrapolating to Unseen Weather Conditions

Regression algorithms may make two types of predictions – interpolation, and extrapolation. Interpolation refers to predictions between training data points, where all feature values are within the limits of the training dataset. Extrapolation refers to predictions outside these limits – making predictions about "unseen" data. Figure 1 from Giavatto and Fleck [10] below shows a graphical depiction of this concept.

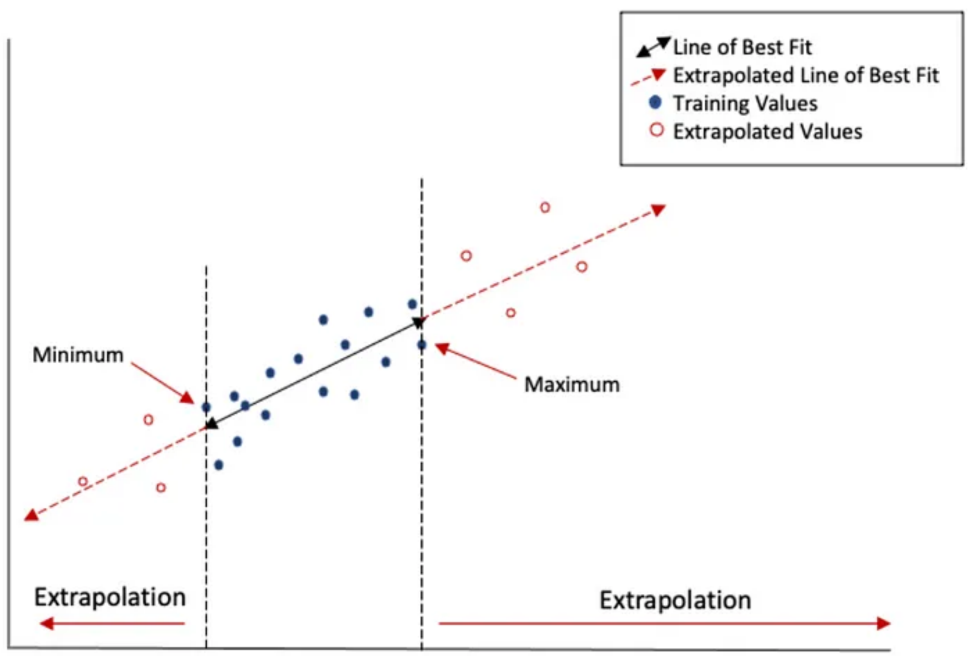

Figure 1: Interpolation vs. Extrapolation [10]

An important feature of gradient boosting models is that, while they are among the state-of-the-art at interpolation, they cannot extrapolate well. According to Malistov and Trushin, "Therefore, the prediction quality [of gradient boosted decision tree algorithms] degrades if one of the features, such as time, lies outside the boundaries of the training data set." [11]

Extrapolation in this project's model is used when climate change weather data is incorporated – later in the 21st century, already hot states (particularly in the U.S south) will see temperatures hotter than exist in the training data due to global warming. Therefore, the features that refer to those temperatures will be outside the training data limits and the model will be extrapolating to capture these effects. However, currently cold-weather states who do not see temperatures hotter than the modern-day south, even far into the future, will still be interpolating since the weather conditions have been seen before.

A possible remedy to this would be to use a different model type than gradient boosting – a linear regression model, for example. This model may perform better in extrapolation, but is likely to perform worse in interpolation tasks. Given this trade-off, I have chosen to use a gradient boosting model, XGBoost, since I believe the primary use case of this web tool will be interpolation. Users will mainly be looking at measure impact in the near future, or live in cold-weather states– and interpolation accuracy is the most important benchmark for this context.

Additionally, in a comparison of selected models performance in extrapolation, Giavatto and Fleck [10] find that, although linear regression has a smaller relative performance drop from interpolation to extrapolation, the absolute performance of tree-based models is still better in extrapolation than linear regression on several datasets. Although only one source, these results indicate that the sacrificed performance in extrapolation may be small, if present at all.

VI. Project Logistics

A. Stretch Goals

a. Quantitative Model Comparison

It would be beneficial to do a quantitative comparison of different machine learning algorithms, before settling on one. Other than XGBoost, there are other popular gradient boosting implementations, including CatBoost and LightGBM. Random forest regression is also known for its interpolative accuracy, and linear regression could be evaluated as well. Comparing these choices with a quantitative test on a hold-out set of the baseline use would be useful. I could then make a more informed choice about the model with the best speed and accuracy, rather than just assuming XGBoost will suffice.

Additionally, I could test their extrapolative power. By holding out the southern-most states with the hottest weather during training, I could quantitatively test which models perform best on weather data hotter than it has seen before.

b. Website Enhancements

If I am able to finish early, there are improvements I could make to the website to improve the user experience, that are not strictly important to the results, including:

- Detailed loading screen while results are retrieved, with progress bar
- User profile that saves past results to a database
- Additional interactive visualizations
- Links from measures to resources that help users find incentives for a given project, or give advice on how to get started on implementing a measure

B. Backup Plans

If inference time is taking intractably long and can't be mitigated, I could cut down the number of models run by only removing some measure packages, considering only baseline usage at the extreme case. I would consider the minimum viable product to be one that still allows the user to see how their baseline energy may vary with climate change, and different home features.

C. Timeline

Up through the time of writing, the week of July 22nd, so far I have:

- Pre-processed and encoded baseline data (I believe this process will generalize to measure data)
- Created and trained prototype model on baseline data that gives plausible results
- Created Flask API endpoint that that loads and uses the model, sampling relevant houses from the user's county
- Created a test website that allows the user to input some features, calls the API and displays the results

See below for a breakdown of my intended project timeline. I have intentionally built more time into website development than I believe is necessary to allow for issues I may run into before then.

Table 3: Project Timeline

Semester	Week Of	Tasks
First Semester	Jul 29	Model comparison and selection
	Aug 05	Tune prototype model to meet performance requirements
	Aug 12	Data pre-processing on selected measure data
	Aug 19	Exploratory data analysis and feature engineering
	Aug 26	Exploratory data analysis and feature engineering
Second Semester	Sep 02	Train all models
	Sep 09	Train models, evaluate performance
	Sep 16	Update test site to show all results
	Sep 23	Fine-tune models for accuracy and performance, re-train if needed
	Sep 30	Fine-tune models for accuracy and performance, re-train if needed
	Oct 07	Create model performance visualizations
	Oct 14	Website/UI Development, Visualizations
	Oct 21	Website/UI Development, Visualizations
	Oct 28	Website/UI Development, Visualizations
	Nov 04	Website/UI Development, Visualizations
	Nov 11	Website/UI Development, Visualizations
	Nov 18	Website/UI Development, Visualizations
	Nov 25	Summarize Findings
	Dec 02	Summarize Findings
	Dec 09	Presentation, documentation

[1] "HeatingCoolingGuide FINAL_9-4-09.pdf." Accessed: Jul. 16, 2024. [Online]. Available: https://www.energystar.gov/sites/default/files/tools/HeatingCoolingGuide%20FINAL_9-4-09.pdf
[2] B. Goldstein, D. Gounaridis, and J. P. Newell, "The carbon footprint of household energy use in the United States," *Proc. Natl. Acad. Sci.*, vol. 117, no. 32, pp. 19122–19130, Aug. 2020, doi: 10.1073/pnas.1922205117.
[3] "Ignoring Resilience Benefits Limits Growth of Energy Efficiency Programs | ACEEE." Accessed: Jul. 17, 2024. [Online]. Available: https://www.aceee.org/blog-post/2024/03/ignoring-resilience-benefits-limits-growth-energy-efficiency-programs
[4] A. Kasch, "How Much Does A Home Energy Audit Cost?," Forbes Home. Accessed: Jul. 17, 2024. [Online]. Available: https://www.forbes.com/home-improvement/solar/home-energy-audit-cost/
[5] "ResStock - NREL." Accessed: Jul. 19, 2024. [Online]. Available: https://resstock.nrel.gov/datasets
[6] "ResStock Analysis Tool." Accessed: Jul. 19, 2024. [Online]. Available: https://www.nrel.gov/buildings/resstock.html
[7] J. Reyna *et al.*, "U.S. Building Stock Characterization Study: A National Typology for Decarbonizing U.S. Buildings," *Renew. Energy*, 2022.
[8] S. Wilcox and W. Marion, "Users Manual for TMY3 Data Sets," *Tech. Rep.*, 2008.
[9] S. Chowdhury, F. Li, A. Stubbings, J. New, D. Rastogi, and S.-C. Kao, "Future Typical Meteorological Year (fTMY) US Weather Files for Building Simulation for every US County in CONUS (Cross-Model Version-SSP3-RCP7.0)." Zenodo, Feb. 23, 2024. doi: 10.5281/zenodo.10698922.
[10] T. Giavatto, "A Guide for Model Selection During Extrapolation," IBM Data Science in Practice. Accessed: Jul. 22, 2024. [Online]. Available: https://medium.com/ibm-data-ai/a-guide-for-model-selection-during-extrapolation-8efa04e9df7b
[11] A. Malistov and A. Trushin, "Gradient Boosted Trees with Extrapolation," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, Dec. 2019, pp. 783–789. doi: 10.1109/ICMLA.2019.00138.