A Visual Analytics Based Decision Support Methodology

For Evaluating Low Energy Building Design Alternatives

by

Ranojoy Dutta

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2013 by the
Graduate Supervisory Committee:

T Agami Reddy, Chair
Marlin Addison
George Runger

ARIZONA STATE UNIVERSITY

December 2013

ABSTRACT

The ability to design high performance buildings has acquired great importance in recent years due to numerous federal, societal and environmental initiatives. However, this endeavor is much more demanding in terms of designer expertise and time. It requires a whole new level of synergy between automated performance prediction with the human capabilities to perceive, evaluate and ultimately select a suitable solution. While performance prediction can be highly automated through the use of computers, performance evaluation cannot, unless it is with respect to a single criterion. The need to address multi-criteria requirements makes it more valuable for a designer to know the "latitude" or "degrees of freedom" he has in changing certain design variables while achieving preset criteria such as energy performance, life cycle cost, environmental impacts etc. This requirement can be met by a decision support framework based on near-optimal "satisficing" as opposed to purely optimal decision making techniques. Currently, such a comprehensive design framework is lacking, which is the basis for undertaking this research.

The primary objective of this research is to facilitate a complementary relationship between designers and computers for Multi-Criterion Decision Making (MCDM) during high performance building design. It is based on the application of Monte Carlo approaches to create a database of solutions using deterministic whole building energy simulations, along with data mining methods to rank variable importance and reduce the multi-dimensionality of the problem. A novel interactive visualization approach is then proposed which uses regression based models to create dynamic interplays of how varying these important variables affect the multiple criteria, while providing a visual range or band of variation of the different design parameters. The MCDM process has been incorporated into an alternative methodology for high performance building design referred to as Visual Analytics based Decision Support Methodology [VADSM]. VADSM is envisioned to be most useful during the conceptual and early design performance modeling stages by providing a set of potential solutions that can be analyzed further for final design selection. The proposed methodology can be used for new building design synthesis as well as evaluation of retrofits and operational deficiencies in existing buildings.

I dedicate this thesis to my parents and sister for unconditionally supporting me in all my endeavors and for being my refuge in difficult times.

I also dedicate this work to my dear friends Saurabh, Namita, Apoorva and Ranjini (Appu & Ranju) for their support and encouragement and for making the last two years so very enjoyable and fruitful.

ACKNOWLEDGEMENTS

I take this opportunity to express my gratitude to my thesis committee members, who helped shape this research into an extremely interesting and satisfying endeavor.

I thank Prof. George Runger for helping me understand the nuances of statistical learning and for his many deep insights into the subject, which played a key role in working out critical difficulties along the way.

I thank Prof. Marlin S. Addison for laying down a research path for me to follow and for his continued support and interest in the future development of a practical application based on this study.

To Prof. T. Agami Reddy I owe the greatest measure of gratitude for his tireless and invaluable efforts in making this research a reality and for always encouraging me to push the boundaries of my knowledge.

TABLE OF CONTENTS

Page

CHAPTER

CHAPTER Page

APPENDIX

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

1: BACKGROUND

Designing buildings to be energy efficient can be described as a multi-criteria optimization problem whose complexity originates from the large number of variables involved, the dynamic nature of building loads and processes, the intricacy of interaction effects among variables, and the inability of the designer to view cause and effect in multi-dimensional space. This complexity requires a performance based automated methodology which only detailed simulations can provide as opposed to prescriptive approaches based on a designer's intuition and experience (heuristics). The design problem is further complicated by the fact that certain variables will be partially defined or even initially unknown, and there could be multiple design solutions. Finally, the choice of one design alternative can yield energy savings for one end-use (such as reduction in ambient light loads due to day lighting) while simultaneously resulting in an energy penalty for another end-use (increased cooling load due to additional solar gain). Finding the best solution has thus been traditionally viewed as a difficult multi-objective optimization problem that can only be tackled by computers, which have the advantage of computational speed, parallel processing, and accuracy. In multi-objective optimization problems, the searching of a single optimal solution is of little value, since the objectives are often competitive. Instead, a number of feasible intermediary solutions that will satisfy the decision maker are searched through an interactive procedure.

In the context of building energy simulation, such optimization will require numerous simulations and intelligent selections of parameter inputs, sub-system and equipment selections. Current design practices typically pursue individual design solutions since the human cognitive system works in a sequential manner and has difficulty considering many variables simultaneously. The ability of computers to explore multiple solution paths and analyze in parallel simplifies the problem of finding a set of local optimal designs and perhaps a global optimum. Numerous optimization tools already exist and the mathematics of optimization have been well established. However, purely automated optimization routines employing brute force methods, though able to handle a large number of constraints and variables, often provide a mathematically optimal solution that may not be practical or even desired due to aesthetic reasons, program restrictions, or specific owner preferences. Thus, the need to address multi-criteria requirements makes it more valuable for a designer to know the “latitude” or “degrees of freedom” he/she has in changing certain design variables while achieving satisfactory levels of energy performance as well as addressing other relevant criteria like life cycle cost , environmental impacts etc. This requirement can be addressed by a decision support framework based on near-optimal Satisficing (Satisfy+ Suffice) as opposed to single optimal decision making techniques.

While performance prediction can be highly automated through the use of computers, performance evaluation cannot, unless it is with respect to a single criterion. Multicriterion decision-making is the main non-delegable design task that requires human intervention. The rest of the design tasks, however, can and should be automated for faster and more accurate results. Additionally, computers can facilitate the evaluation process though appropriate user interfaces that provide graphical representation of data and allow for direct comparison of multiple solutions with respect to multiple performance considerations. Thus, the design of high performance (low energy) buildings requires a synergy between automated performance prediction & visualization with the human capabilities to perceive, relate and ultimately select a satisficing solution. Currently such a comprehensive design framework is lacking and hence this thesis presents a new methodology for low energy building design evaluation.

2: SCOPE AND OBJECTIVES

The primary objective of this research is to facilitate a complementary relationship between human designers and computers for Multi-Criteria Decision Making **(MCDM)** in the domain of low energy building design. The MCDM process has two elements (Wright & Loosemore, 2001)

1) A procedure to ***search*** for one or more solutions that reflect the desired pay-off between the criteria.
2) The designer must make a ***decision*** as to which pay-off between the criteria results in the most desirable design solution;

In the present research, the **MCDM search** element has been executed using data mining techniques while the **MCDM decision** making component has been supported through interactive visualizations. The complete MCDM process has been incorporated into a new methodology for low energy building design referred to as a **Visual Analytics based Decision Support Methodology [VADSM].**

Visual analytics is the science of analytical reasoning facilitated by interactive visual interfaces. It has the potential to unify the language of building design which is primarily visual (form-based / graphical) with traditional analysis techniques that are predominantly numeric. Visual analytics can extend the benefits of analytical techniques such as simulation to a wider audience of designers who are more comfortable with the visual exploration of data.

The proposed methodology VADSM begins by identifying key design variables and their ranges defined by the owner/designer, and two or more response variables deemed key for decision making, such as annual energy use or peak electric demand. In the conventional use of simulation tools the inputs are in essence already decided and the resulting outputs are a function of those choices. This strategy only allows a limited trial and error design analysis. However, if simulated outputs are used instead to fine tune the inputs, then that would potentially transform a design analysis operation into a design synthesis opportunity. This is one of the key concepts incorporated in VADSM.

Once the variable ranges have been determined, appropriate experimental design techniques are adopted to generate a feasible number of simulation runs (variable combinations) with respect to run time, and to ensure uniform sampling over the entire solution space. The batch simulated data can then be analyzed using state of the art data mining algorithms to ascertain variable importance, and irrelevant variables can be discarded to create simpler predictive models using traditional regression based techniques. The subsequent stage requires a Graphical User Interface (GUI) to provide designers a way of visualizing the predictive models and perform what-if scenarios in real time. The **Decision Support Model Viewer (DSMV)** application (Sec. 6.2.4) has been developed to fulfill this requirement. It allows designers to quickly and easily specify the characteristics of potential designs through direct manipulation of multiple inputs and get real time information about their energy performance. The DSMV also allows a designer to visually keep track of how a specific change in a single variable affects the “degrees of freedom” of other variables, by dynamically updating variable ranges.

The key objectives of VADSM are the following:

(i) utilize detailed building energy simulation programs for design synthesis,

(ii) Develop a decision-support tool which allows the user to explore design “latitude” or “range of variability” of different design variables.

(iii) generate a set of “satisficing” solutions rather than one unique solution,

(iv) Provide real-time feedback on how design decisions involving a change in key design variable values impact building performance criteria.

(v) Create a learning tool which assists in developing a designer’s intuition

VADSM is envisioned to be most useful during the conceptual and early design performance modeling stages by proving a set of potential solutions that can be analyzed further for final design selection. Due to its reliance on statistical methods for estimating building energy performance, VADSM is not meant to be a substitute for detailed simulation during the final design phase .The proposed methodology, however, can be used for new building design as well as identifying operational improvements and/or evaluating efficiency retrofits in existing buildings.

3: LITERATURE REVIEW

3.1 Building Design as a Multi-criterion Decision Making (MCDM) activity

Designing buildings to be energy efficient is by no means a straightforward process; building materials, building components, and building systems all have individual as well as interacting impacts on building energy use. The complexity of the design problem originates from the large number of variables involved, the dynamic nature of building loads and processes, and the intricacy of interaction effects among variables. The choice of one design alternative can yield energy savings for one end-use while simultaneously resulting in an energy penalty for another end-use. Choosing from the wide variety of innovative technologies and energy efficiency measures available today, a decision maker (DM) has to compensate environmental, energy, financial and social factors in order to reach the best possible solution that will ensure the maximization of the energy efficiency of a building while satisfying the final user/occupant/owner needs (Diakaki, Grigoroudis, & Kolokotsa, 2008). What is required is a model that will allow designers to explore the consequences of decisions relating to these variables at the conceptual stage of design, and hence design a building that achieves a good balance between multiple objectives (D'Cruz & Radford, 1987) .

(Diakaki et al., 2008) suggest two approaches to solving this problem. According to the first approach, an energy analysis of the building under study is carried out, and several alternative scenarios, predefined by the energy expert, are developed and evaluated. These specific scenarios, which may vary according to buildings' characteristics, type, use, climatic conditions, etc., are pinpointed by the building expert and are then evaluated mainly through simulation. The selection of the alternative scenarios, energy efficiency measures and actions that will be finally employed is largely based on the energy experts' experience. The second approach includes decision supporting techniques, such as multicriteria-based decision making (MCDM) methods (Zionts, 1979) that are employed to assist in a final decision being reached. The coupling between design criteria and its impact on the design solutions can be evaluated through the application of such methods.

3.1.1 Multi-objective vs Single-objective Optimization

Optimization is an essential process in many business, management, and engineering applications where multiple and often conflicting objectives need to be satisfied. Solving such problems has traditionally consisted of converting all objectives into a single objective (SO) function. The ultimate goal is to find the solution that minimizes or maximizes this single objective while maintaining the physical constraints of the system or process (Ngatchou, Anahita Zarei, & El-Sharkawi, 2005). The optimization solution results in a single value that reflects a compromise between all objectives. Conversion of the multiple objectives into an SO function is usually done by aggregating all objectives in a weighted function, or simply transforming all but one of the objectives into constraints. This approach to solving multi objective (MO) optimization problems has several limitations (Ngatchou et al., 2005):

1) It requires a-priori knowledge about the relative importance of the objectives, and the limits on the objectives that are converted into constraints
2) The aggregated function leads to only one solution;

3) Trade-offs between objectives cannot be easily evaluated; and

4) The solution may not be attainable unless the search space is convex.

This simple optimization process is no longer acceptable for complex systems such as buildings with multiple conflicting objectives. Compared to SO problems, MO problems are more difficult to solve, because there is no unique solution; rather, there is a set of acceptable sub-optimal solutions. In multi-objective optimization problems, the searching of a single optimal solution is futile, since the objectives are often competitive. Instead, a feasible intermediary solution that will satisfy his/her preferences is searched out through an interactive procedure involving the decision maker. A strategy suited to this type of search has been demonstrated by (Addison, 1988) based on the idea of **satisficing**, a term coined by H.A. Simon in the context of economic theory (Simon, 1957) . Simon proposes the idea of bounded rationality and suggests that in general, individuals look for alternatives which are "good enough" rather than optimal. An alternative is "good enough" if it satisfies the individual's aspiration levels and suffices in the absence of a practicably obtainable optimum. In the context of building design, these aspiration levels may alternately be considered performance thresholds (Addison, 1988).

3.1.2 Multi-objective Search – Pareto Optimality

Multi-objective optimization is a scientific area that offers a wide variety of methods with great potential for the solution of complicated decision problems .The concept of multi-objective optimization is attributed to the Italian economist Vilfredo Pareto (1848-1923) who used it in his studies of economic efficiency and income distribution . After several decades, this concept, referred to as Pareto efficiency or Pareto optimality, was recognized in operations research and eventually found extensive applications in engineering optimization. Pareto optimality makes use of the concept of dominated and non-dominated solutions. A solution is Pareto optimal if it is not dominated by any other solution. In Figure 1 the points represent feasible solutions to a multi-objective minimization problem, where values for each of the two objective functions are assigned to the x and y axes. A solution dominates another if it is better than the other for at least one objective function and at least as good on all the others (Caldas & Norford, 2003).

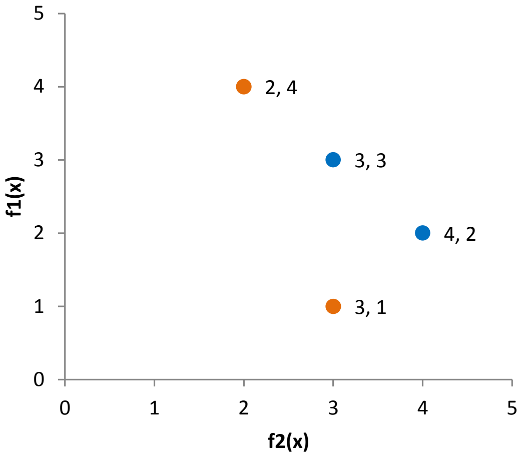


Figure 1 : Dominated and non-dominated [Pareto] solutions

Point (3, 1) dominates point (4, 2) because it has both lower x and y values. It also dominates point (3, 3) because it has a lower y value for the same x value. Points (3, 1) and (2, 4) are not dominated with respect to each other, and are therefore both Pareto-optimal solutions. They represent trade-offs between the two objective functions. Point (2, 4) performs better than (3, 1) in terms of the x values but the inverse is true for y values. Once all the dominated solutions are eliminated, the DM is left with a Pareto optimum set of solutions for final selection.

(Gero, D'Cruz, & Radford, 1983) were among the first to propose a multi criteria-model in order to explore the trade-offs between the building thermal performance and other criteria such as capital cost, and usable area of the building during building design. However, the number of design variables and decision options were restricted to enable Pareto optimal solutions to be identified through the process of exhaustive enumeration and tests of domination. In a subsequent paper (D'Cruz & Radford, 1987), provided as a continuation to the earlier research, additional performance criteria were added and the optimization problem was solved using a dynamic programming optimization algorithm instead of exhaustive enumeration. Although solutions where obtained, they were not sufficient to allow the pay-off between the criteria to be examined. A solution to this deficiency was examined through the use of a multi-criterion Genetic Algorithm (GA) optimization method by (Wright & Loosemore, 2001). More recently, other researchers have also employed multicriteria techniques to similar problems ((Diakaki et al., 2008) .

Although several “traditional” methods exist, these often require a sequential and therefore computationally intensive approach to finding the Pareto set of solutions (Wright & Loosemore, 2001). Rather than progressively minimizing a single possible solution, GA’s operate with a set of possible solutions (known as the population). This enables several members of the Pareto optimum set to be found in a single run of the algorithm. A genetic algorithm starts by generating a number of possible solutions (individuals) to a problem, evaluates them and then applies the basic genetic operators (reproduction, crossover and mutation) to that initial population according to the fitness of each individual. This process generates a new population with higher average fitness than the previous one, which in turn will be evaluated. The cycle is repeated for the number of generations set by the users, which is dependent on problem complexity (Caldas & Norford, 2003). A more detailed discussion of genetic algorithms is beyond the scope of the present research and the interested reader is directed to the works of David E. Goldberg and Kalyanmoy Deb.

3.2 Data Mining

The amount of data available nowadays to scientists, engineers and business mangers is vast. Almost all the data is stored electronically in databases and commonly connected and accessed via cloud infrastructure. Additionally, the rate of growth of data sets exceeds by far the coping ability of traditional "manual" analysis techniques. Hence, if one is to utilize the data in a timely manner, it would not be possible to achieve this goal if a traditional data analysis approach were followed. Effectively this means that most of the data would remain unused or un-analyzed. Data can grow along two dimensions: the number of fields (also called dimensions or attributes) and the number of cases. Human analysis and visualization abilities do not scale to high-dimensions and massive volumes of data.

Data mining techniques allow for the possibility of computer-driven exploration of data. This opens up the possibility for a new way of interacting with databases: specifying queries at a much more abstract level than SQL (Structured Query Language) permits. This addresses the *query formulation problem:* how can we provide access to data when the user does not know how to describe the goal in terms of a specific query or even as a computer program in a stored procedure? (U. Fayyad, 1997) . Data mining also facilitates data exploration for problems that, due to high dimensionality, would otherwise be very difficult to explore by traditional statistical methods and conventional graphing techniques. Data Mining is the mechanized process of identifying or discovering useful structure in data (U. M. Fayyad, Grinstein, Wierse, & NetLibrary, 2002). Data mining techniques are deployed to scour large databases in order to find novel and useful patterns that might otherwise remain unknown (Tan, Steinbach, & Kumar, 2005).

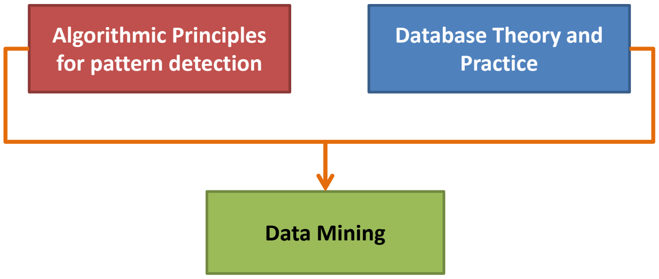

Algorithmic Principles for pattern detection
Database Theory and Practice
Data Mining

Figure 2 : Foundations of Data Mining

Stemming from a purely computational approach, two distinct groups working on two fundamental aspects of data mining have emerged out of the field of computer science. The first focused on **data storage and information retrieval** technology as related to database theory and practice. The second notion of data mining, as **algorithmic principles** that enable the detection or extraction of patterns, evolved under the field of pattern recognition, and later under artificial intelligence (AI), machine learning (ML) and most recently the field of Knowledge Discovery in Databases (KDD) (U. M. Fayyad, Piatetsky-Shapiro, & Smyth, 1996). The last three or four years of the twentieth century saw the successful merging of database inspired techniques with KDD algorithms .

3.2.1 Data mining objectives

Data mining objectives are generally divided into two major categories (Tan et al., 2005)

Predictive tasks: The objective is to predict the value of a particular attribute based on the values of other attributes. The attribute to be predicted is commonly known as the target or dependent variable while the attributes used for making the prediction are known as the independent variables.

Descriptive tasks: The objective is to derive patterns (correlations, trends, clusters etc.) that summarize the underlying relationships in data. Descriptive data mining tasks are often exploratory in nature and require post processing techniques to validate and explain the results.

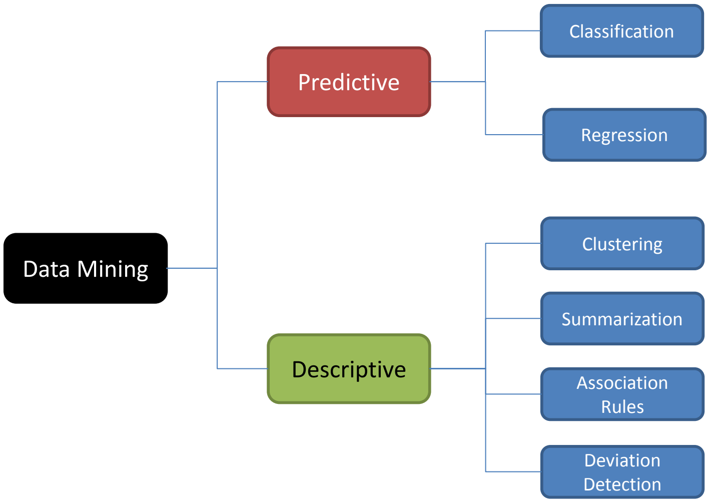


Figure 3 : Data Mining Tasks

Figure 3 depicts a further breakdown of the two broad categories into five classes of data mining methods. Fayyad et al. (U. M. Fayyad et al., 1996) provide a list of the tasks required to meet the primary goals of data mining.

Classification is the process of assigning the most likely categories to features or trends within the data. Identification of interesting features within the data is a form of classification. This is used for discreet target variables. For example, predicting whether a web user will make a purchase at an online store is a classification task since the outcome is binary

Regression is development of a function that approximates the mathematical relationship between attributes and a continuous target variable. Forecasting the future price of stock is a regression task because price is a continuous valued attribute.

Clustering is also known as segmentation and seeks to find groups of closely related observations that belong to a cluster such that those observations are more similar to each other than observations that belong to other clusters. Cluster analysis is described as unsupervised learning since the categories are learnt from the data itself and are not pre-defined by the investigator. The grouping of news articles determined by the frequency of certain key words is a clustering exercise.

Summarization is the process of finding a compact representation for data. There are two classes of methods which represent taking horizontal (cases) or vertical (fields) slices of the data. In the former, one would like to produce summaries of subsets: e.g. producing sufficient statistics. In the latter case, one would like to predict relations between fields. This class of methods is distinguished from the above in that rather than predicting a specified field (e.g. classification) or grouping cases together (e.g. clustering) the goal is to find relations between fields. One common method is called association rules (Agarwal, Mannial, Srikant, Toivonen, & Verkamo, 1996). Associations are rules that state that certain combinations of values occur with other combinations of values with a certain frequency and certainty. A common application of this is market basket analysis where one would like to summarize which products are bought in conjunction with what other products.

Dependency modeling is used to discover patterns that describe strongly associated features in data. It is a process of modeling dependencies or causality between variables. The discovered patterns are typically represented in the form of implication rules or feature subsets. Useful applications include finding groups of genes that have related functionality or identifying web pages that are accessed together.

Deviation or Anomaly Detection – This is the task of identifying observations whose characteristics are significantly different from the rest of the data or which fall outside some normal change. The goal of an anomaly detection algorithm is to discover the real anomalies and avoid falsely labeling normal objects as anomalous. The distinguishing feature of this class of methods is that the ordering of observations is important and must be accounted for. Applications include detection of fraud, network intrusions and disease propagation.

3.2.2 Data mining and knowledge discovery in databases

Data Mining is an integral part of KDD which is the overall process of converting raw data into useful information. However, there is potential for confusion about the distinction between KDD and data mining. Fayyad et al. (U. M. Fayyad et al., 1996) claim that KDD is the process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data ,whereas data mining is simply the application of algorithms for extracting patterns from data. Data mining is to be viewed as a subset of the KDD process. Figure 4 depicts this process of KDD which consists of a series of transformation steps, from data preprocessing to post processing of data mining results for knowledge extraction.

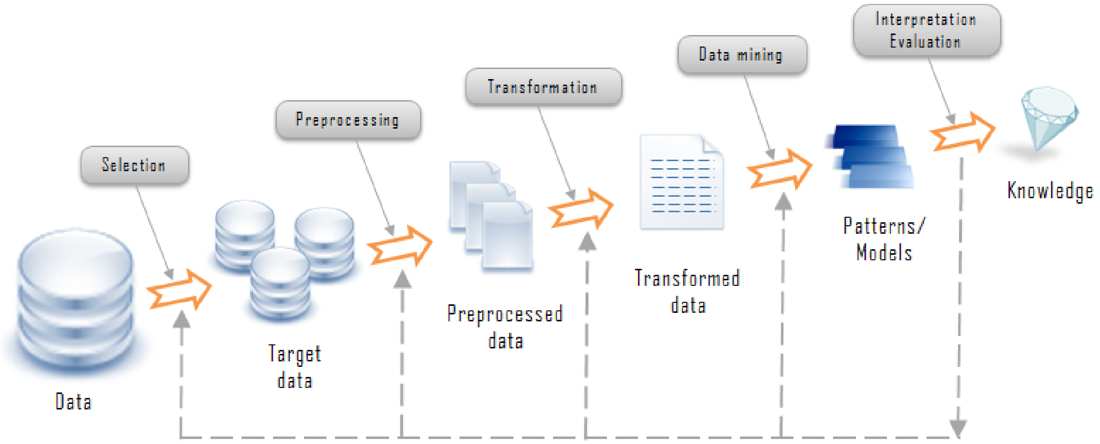


Figure 4 : The Process of Knowledge Discovery in Databases. (U. M. Fayyad et al., 1996)

The basic steps involved in KDD are described by (U. M. Fayyad et al., 1996) and (Han & Kamber, 2001)

1. Developing a pool of expert knowledge and end-user goals

2. Selecting the data for which KDD is to be performed
3. Data cleaning and pre-processing (handling noise and missing data)
4. Data reduction and transformation : reducing the number of attributes to the minimum necessary to meet the end user goals
5. Choosing the data mining task: deciding whether the end user goal can be met by classification, regression, clustering etc.
6. Choosing the data mining algorithm
7. Searching for patterns or rules using data mining
8. Pattern interpreting i.e., the user examines the results of the preceding steps and may decide to repeat them if necessary
9. Consolidating discovered knowledge i.e., incorporating new knowledge into the data base.

The KDD process may contain loops between any two of these steps and may involve several iterations of any subset of this list.

3.3 Data mining applications in the building energy domain

3.3.1 General introduction

The energy performance of a building is influenced by many factors, such as weather conditions, thermal properties of the construction materials, occupant behavior, sub-level components such as lighting, HVAC systems, their performance and schedules. Due to the complexity of the problem, precise consumption prediction is quite difficult. In recent years, a large number of approaches for prediction, either elaborate or simplified, have been proposed and applied to a broad range of problems. These approaches (Zhao & Magoulès, 2012) include engineering based methods such as white box models based on physical principles, statistical methods such as Least Squares Regression, Fourier series models and machine learning methods such as Artificial Neural Networks, Support Vector Machines, Decision Tree Induction etc. Such research work has been carried out in the process of building design, operation or retrofit of contemporary buildings; varying from a building's sub-system analysis to regional or national level modeling. Predictions can be performed on the whole building or sub-level components by thoroughly analyzing each influencing factor or approximating the usage by considering several major factors. Building energy simulation software based on physical principles calculate the thermal dynamics and energy behavior of buildings and are widely used to analyze or forecast energy consumption in order to facilitate the design and operation of energy efficient buildings. Simulation software may provide reliable solutions to estimate the impact of building design alternatives; however this process can be very time-consuming, requiring detailed inputs and user-expertise in a particular program. Moreover, the accuracy of the estimated results may vary across different building simulation software packages (Crawley, Hand, Kummert, & Griffith, 2008). Hence, in practice many researchers have begun to rely on machine learning tools to study the effect of various building parameters on certain variables of interest because this is easier and much faster if a database of the required ranges of variables is available for training the model (Dong, Cao, & Lee, 2005).

Traditionally, least squares regression analysis has been the most popular technique in predicting energy consumption. However many studies in the general research area of energy performance of buildings (EPB) using classical regression techniques have made simplifying mathematical assumptions relying on linear correlations and normality which are known to be ill-suited for many complicated applications where normality assumptions do not hold (Tsanas & Xifara, 2012). State of the art nonlinear and nonparametric machine learning techniques such as Random Forest and Artificial Neural Networks overcome these limitations inherently and do not require any prior knowledge of variable distribution or structure of the feature space. Moreover pattern extraction using data mining can enhance the designer’s understanding through quantitative expressions of the factors that affect the quantity (or quantities) of interest that the building designer or architect may wish to focus on. A useful pattern could provide the building designer a strategy to increase the energy efficiencies of their buildings. A discovered pattern might relate to a single parameter like building insulation values that reduce energy use by 40-50%, with other factors held constant. Patterns might also be much more complex, taking into account several different building components such as walls, windows, doors, and roof and specifying the conditional probability of improving energy efficiency. Another related example involving useful patterns lies in the area of load forecasting for the electricity supply industry, where data mining can be used to detect correlations between climatic conditions and other characteristics that influence load demands, such as the time of day or week. Again, this is a situation where the amount of data in combination with the complexity of potential correlations makes it difficult to manually determine these patterns from the data set (Morbitzer, 2003).

Recognizing the complexity of building energy performance prediction and the inadequacy of some of the traditional statistical approaches efforts to integrate machine learning with EPB has sparked enormous interest (Tsanas & Xifara, 2012). In the context of EPB, various machine learning techniques such as support vector machines (Dong et al., 2005) artificial neural networks (Kalogirou, 2000), CART (Yu, Haghighat, Fung, & Yoshino, 2010) and Random Forest (Tsanas & Xifara, 2012) have been explored to predict various quantities of interest. A brief overview of these techniques and a discussion of their applications for building energy prediction is provided in the following section.

3.3.2 Artificial Neural Networks (ANNs)

ANNs is a nonlinear statistical technique principally used for prediction and is the most widely used artificial intelligence method in the domain of building energy performance (Zhao & Magoulès, 2012). Neural network models were originally developed by researchers trying to mimic the neurophysiology of the human brain. A neural network can be any model in which the output is computed from the inputs by compositions of basic functions. One of the most commonly used neural network models consists of several "neurons" that are connected to each other through multiple layers. Neural networks perform well in applications when the functional form is nonlinear, and are especially useful for prediction problems where prior knowledge on the relationship between inputs and outputs are unknown (Tso & Yau, 2007).

In the past twenty years, researchers have applied ANNs to analyze various types of building energy consumption in a variety of conditions, such as heating/cooling load, electricity consumption, sub-level component operation/optimization, and estimation of parameters. Neural Network applications for building energy analysis were pioneered by the University of Colorado in the early 90’s (Krarti, Kreider, Cohen, & Curtiss, 1998). Kalogirou (Kalogirou, 2000) has published many papers on building applications using ANN, including a bibliographic review summing up the applications of ANNs in the field of energy-engineering systems. (Olofsson & Andersson, 2001) developed a neural network which makes long-term energy demand (the annual heating demand) predictions based on short-term (typically 2–5 weeks) measured data with a high prediction rate for single family buildings. Since the measured data was short term, in order to train the neural network on the seasonal variation they used energy calculation software to generate heating demand for a synthetic building comparable to the actual one. An early study by (Krarti et al., 1998) successfully trained ANNs to estimate the energy savings due to retrofits in existing commercial buildings. A joint U.S Japanese research project (Kawashima, Dorgan, & Mitchell, 1995) investigated several time series modeling methods for hourly thermal load prediction over a 24h time horizon and compared the accuracy of each model. The results indicated that an artificial neural network (ANN) produced the most accurate thermal load predictions. The ANN model was then applied to two measured building loads from another research project and the results confirmed its accuracy.

In the area of building electricity usage prediction (Wong, Wan, & Lam, 2010) used a neural network to predict energy consumption for office buildings with day-lighting controls in subtropical climates. A total of nine variables were used as the input parameters – four variables related to the external weather conditions, four for the building envelope designs, and the last variable was day type. They used EnergyPlus as the building energy simulation software to generate daily building energy use data for the training and testing of ANNs. The outputs of the model included estimated daily electricity use for cooling, heating, electric lighting and total building. ANNs are also used to analyze and optimize sub-level components behavior, mostly for HVAC systems. (Lee, House, & Kyong, 2004) used a general regression neural network to detect and diagnose faults in a building's air-handling unit. (Lundin, Andersson, & Östin, 2004) used ANNs to estimate building energy performance parameters like total heat loss coefficient, heat capacity and the gain factor, which are important for reliable energy demand forecasts. (Zhao & Magoulès, 2012) provide an extensive review on the various applications of ANNs in the building energy domain.

ANN is a powerful technique with proven potential for building energy prediction; however, it is limited by a lack of interpretability and the fact that it requires a large amount of learning data and a complete database (that is no missing data in the databases and the same amount of information for each variable). The following statistical learning technique called support vector machines overcomes these difficulties.

3.3.3 Support Vector Machines (SVMs)

Support vector machine (SVM) was introduced in 1995 by (Cortes & Vapnik, 1995). This artificial intelligence technique is usually used to solve binary classification problems or regression. For a complete technical overview of SVM refer to (Tan et al., 2005) or (Hastie, Tibshirani, & Friedman, 2009). In the building energy field, SVM is mainly used for the forecasting of energy consumption (hourly, monthly etc.). One huge advantage is that it can work with heterogeneous databases where all variables do not have the same amount of information or have missing data (Foucquier, Robert, Suard, Stéphan, & Jay, 2013). Also, there are fewer free parameters to optimize compared to ANNs.

The use of SVM in the forecasting of energy consumption in buildings is fairly recent. (Dong et al., 2005) were the first to use SVM for the prediction of building energy consumption. Based on this study, they found the performance of SVM in terms of CV (< 3%) to be better than related results using ANNs and genetic programming. (Li, Meng, Cai, Yoshino, & Mochida, 2009) used SVMs to predict the hourly cooling load of an office building and also found the performance of the support vector regression to be better than the conventional back propagation neural networks. An extensive review of applications of SVMs for predicting energy consumption of buildings as well as HVAC fault detection and diagnostics is provided by (Zhao & Magoulès, 2012). One drawback with SVMs is that the training time varies between quadratic and cubic with respect to the number of training samples (Dong et al., 2005). The other difficulties lie in selecting the best kernel (weighting) function corresponding to a dot product in the feature space and the parameters of the kernel function.

3.3.4 Classification and Regression Trees (CART) and Random Forest (RF)

CART is a conceptually simple, yet powerful, nonlinear method that works by successively splitting the input feature space into smaller and smaller sub-regions (Breiman, Friedman, Olshen, & Stone, 1984). This procedure can be visualized as a tree (called a Decision Tree) that originates from a root node and splits into successively smaller branches terminating in leaf nodes. Each branch represents a sub-region of the input variable ranges and the leaves contain data that "traverses" from the root node down the branches based on optimum splitting values of selected attributes. The tree grows until a certain stopping criterion has been met or the data in a node is completely homogenous. The tree is able to process both numerical and categorical variables, and perform classification and prediction tasks rapidly without much computation. A major advantage of the decision tree over other modeling techniques is that it produces a model which may represent interpretable rules or logic statements. A single decision tree, however, is susceptible to noise (over fitting) and is considered a high variance model.

A natural extension of CART is random forests (RF), which is simply a collection (or ensemble) of many trees (Breiman, 2001). By averaging predictions across multiple trees the overall variance is effectively reduced. The training procedure is the same as in CART with two key differences: a randomly chosen subset of candidate variables are used to select the optimal variable for each split and each tree is trained on a bootstrapped sample generated from the original training data. Practice has shown the RF algorithm works extremely well in many diverse applications. Moreover, RF has the desirable ability of promoting the most important input variables towards predicting the output variable as part of their inherent learning strategy (Hastie et al., 2009). CART and RF are discussed at length in the methodology section of this study.

Applications of CART and RF in the building energy domain are sparse. (Tso & Yau, 2007) compared the prediction accuracy of three different approaches: regression, decision trees and neural networks for electrical energy use of residential households in Hong Kong. Results arising from this study were used by utility companies in assessing electricity energy consumption patterns and selecting a more accurate approach to estimating future energy demand. In the summer phase, the decision tree model resulted in a fewer numbers of significant factors influencing energy consumption as compared to the neural network and stepwise regression models. The decision tree model, with its simpler structure, was also found to be marginally more accurate than the other models, based on Root Mean Squared Error (RMSE) criteria. (Kim, Stumpf, & Kim, 2011) used the C4.5 decision tree algorithm for selecting the building elements most likely to improve energy efficiency in the design of a mid-size community emergency station. The case study revealed that data mining based energy modeling can help project teams discover useful patterns to improve the energy efficiency during the design phase. (Yu et al., 2010) applied the decision tree method to Japanese residential buildings for predicting and classifying building EUI (Energy Use Index) levels. The results demonstrated that the use of the decision tree method can classify and predict building energy demand levels accurately (93% for training data and 92% for test data), identify and rank significant factors automatically, and provide the combination of significant factors as well as threshold values that will lead to high building energy performance.

(Tsanas & Xifara, 2012) used Random Forest (RF) to study the effect of eight input variables (relative compactness, surface area, wall area, roof area, overall height, orientation, glazing area, and glazing area distribution) on two output variables, namely heating load (HL) and cooling load (CL), of residential buildings. They compared the RF to a classical linear regression technique (Iteratively Reweighted Least Squares – IRLS) and found that RF greatly outperformed IRLS in finding an accurate functional relationship between the input and output variables. Classical regression settings (such as IRLS) may fail to account for the presence of multi-collinearity, wherein variables appear to have large magnitude but opposite sign regression coefficients (Hastie et al., 2009). On the contrary, the RF learning mechanism randomizes the selection of a subset of features for each split, and thus can internally account for redundant and interacting variables (Breiman, 2001) .

A body of research thus clearly supports the fact that machine learning techniques are viable alternatives to physical modeling and traditional statistical analysis of building energy data. However, the disadvantages of such tools are that they often require extensive training data and are complex black box models that are not interpretable without advanced statistical knowledge. With the exception of CART, none of the other machine learning techniques allow much model visualization; which is often a critical component for knowledge discovery that cannot be provided by automated data mining alone. The following section presents selected techniques of high dimensional data visualization that can enhance the process of knowledge discovery through visual analytics.

3.4 Visualization of High Dimensional Data

Visualization is the visual representation of data designed to capture inherent correlations or patterns. Data is translated into graphical representation by means of the combined use of points , lines , a co-ordinate system, numbers, symbols, words , shading, and color (Tufte, 2001). The simple line graph or scatter plot has been used for visualization for hundreds of years. Perhaps they are the most widespread method of understanding the interaction of two variables. Understanding the expression of one value as a function of the second is easier if the function is plotted on a graph. The relationships between three variables can be partially understood by a three-dimensional view. The ability to understand the interactions or correlations between more than three variables becomes severely compromised when standard visualization tools are relied upon.

High-dimensional data contains all those sets of data that have more than three variables. The extraction of relevant and meaningful information out of high dimensional data is notoriously complex and cumbersome. The curse of dimensionality is a popular way of stigmatizing the whole set of troubles encountered in high-dimensional data analysis; finding relevant projections, selecting meaningful dimensions, and getting rid of noise, being only a few of them (Bertini, Tatu, & Keim, 2011). Multi-dimensional data visualization also carries its own set of challenges; for example the limited capability of any technique to scale to more than a handful of data dimensions. The following section is a brief survey of selected techniques for High Dimensional Data Visualization. The different techniques can be distinguished between icon-based, hierarchical and geometrical methods (U. M. Fayyad et al., 2002).

3.4.1 Icon-based methods

Icon-based methods are approaches that use icons (or glyphs) to represent high-dimensional data by mapping data components to graphical attributes. The most famous technique is the use of **Chernoff faces** (D. Keim, 1995). In this case, a data point is represented by an individual face while the facial features map to the data dimensions. This is a widely-accepted way of visualizing multidimensional data that capitalizes on human sensitivity to faces and facial features (Parsaye & Chignell, 1993). Figure 5 depicts an annotated Chernoff Face that can be used to represent data with 11 dimensions or attributes. Each dimension can in turn have multiple levels. For example, five different sizes of the eyes could correspond to the five levels of an attribute.

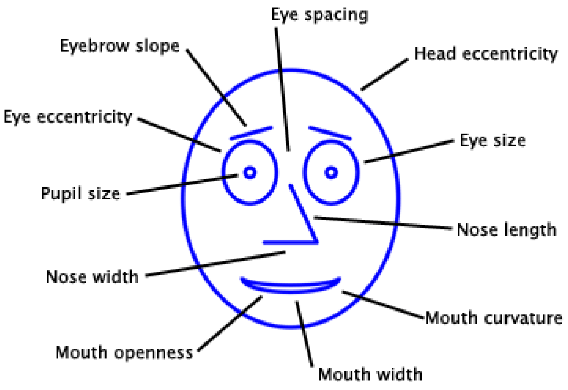


Figure 5 : Annotated Chernoff Face (http://bradandkathy.com/software/faces.html)

Probably the most common icon-based technique is the use of **star glyphs** to denote data points (Hoffmann & Grinstein, 2002) . A star glyph consists of a center point with equally angled rays. These branches correspond to the different dimensions and the length of the limbs mark the value of this particular dimension for the studied data point. A polygon line connects the outer ends of the spokes .This is similar to plotting points in polar co-ordinates instead of the familiar Cartesian co-ordinates. In a typical display, there is a star glyph for every n-dimensional data point. An illustration of the star glyphs approach is provided in Figure 6 below.

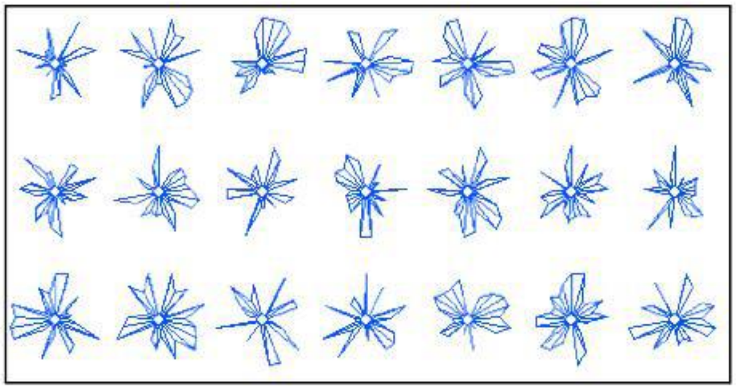


Figure 6 : Star Glyphs (Hoffmann & Grinstein, 2002)

These icon-based techniques are very vivid but have several disadvantages. A very severe problem is the organization of the glyphs on the screen as no coordinate system representing two of the dimensions is provided. Another obstacle is the amount of variables and the size of the data set itself. If the number of rays become too high the distinction between the different spokes and the values they represent is no longer clear or discernible. Lastly, these types of representations may not be well suited for engineering applications that require quantitative comparisons.

3.4.2 Hierarchical methods

The most important representative of the group of hierarchical visualization techniques is **dimensional stacking**. It is a method of embedding coordinate systems recursively into each other (Grinstein, G., Hoffmann, P., Pickett, R., 2002).This method is very useful for hierarchical data sets that only have a small number of dimensions as otherwise the embedding process will make the resulting plot too crowded. The question of labeling can also become difficult with higher dimensions. Figure 7 shows an example of dimensional stacking with five products, five territories, two sales channels, two methods of payment and five quarters (Mihalisin, 2002). Additionally, the number of items sold can be depicted using a color scale as shown in the upper right of Figure 7.

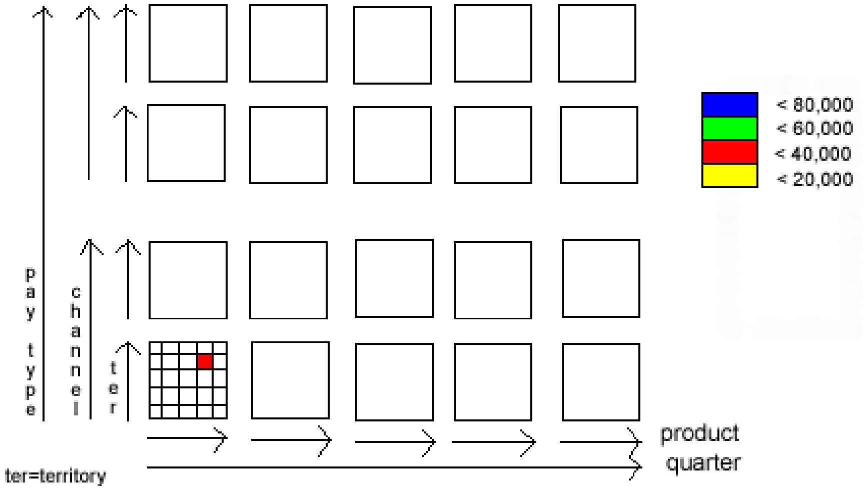


Figure 7 : Dimensional Stacking (Mihalisin, 2002)

A technique that displays the correlation between dimensions (not the data itself) recursively is the **fractal foam** (Hoffmann & Grinstein, 2002) .The starting point is a chosen dimension that is depicted by a colored circle. Attached to this circle are further circles, which symbolize the other dimensions. The size of these rings corresponds to the correlation between the inner circle and the fastened ones. A high correlation requires a large circle. Fixed to the second layer of circles is a third layer which describes the correlation of these dimensions and so on. An example of fractal foam used on the Iris Data can be seen in Figure 8 . The sepal length is center (white), petal length - right (red), petal width - top (yellow) and sepal width - bottom (green)

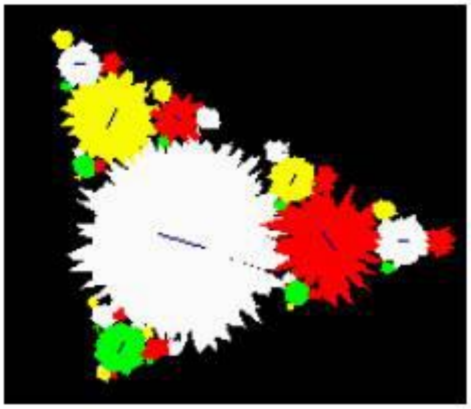


Figure 8 : Fractal Foam display of the Iris Data (Hoffmann & Grinstein, 2002)

3.4.3 Geometrical methods

Geometrical methods cover a large group of visualization techniques. Probably the most commonly used one is the method of **parallel coordinates** (PC). Alfred Inselberg began the work on parallel coordinates in 1981 while working at the IBM research laboratory

(Siirtola & Räihä, 2006) . He suggested that PC can yield graphical representations of Multi-Dimensional relations rather than just finite point sets.

In a PC plot, the dimensions are represented by parallel lines, which are equally spaced. They are linearly scaled so that the bottom of the axis stands for the lowest possible value whereas the top corresponds to the highest value. A data point is now drawn into this system of axes with a polyline, which crosses the variable lines at the locations the data point holds for the examined dimension. A simple example with three points and four dimensions is shown in Figure 9 .The points displayed are A = (1; 3; 2; 5), B = (2; 4; 1; 6) and C = (1; 4; 3; 5).

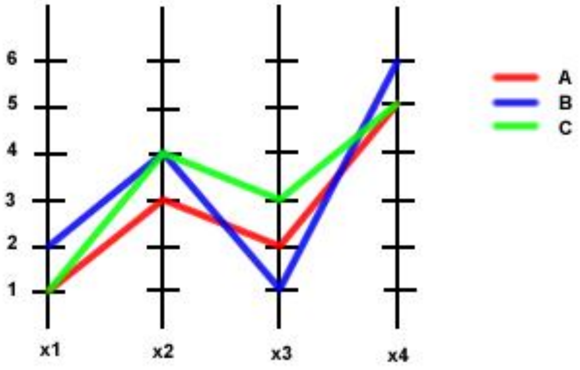


Figure 9 : Parallel Co-ordinates Plot (Hoffmann & Grinstein, 2002)

The number of polylines and variables that can be added is only limited by the size of the computer screen and the limitation of visualization by the human eye (Peterson, 2009). PC plots are also able to visualize non-numerical data with each axis having its own scale and data range. This makes the PC plot a powerful tool with which to visualize multidimensional data (Siirtola & Raiha, 2006). It should be noted, however, that for numeric variables that vary widely in magnitude or have outliers, the attributes need to be standardized before plotting ,else the variation on the smaller variables might not be visible at all.

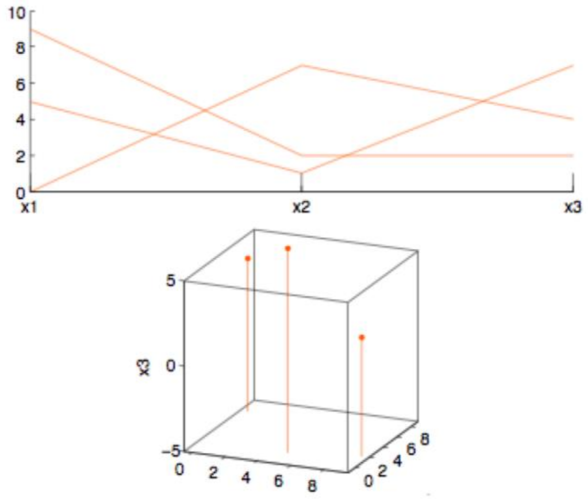


Figure 10 : Three Dimensional Points in Parallel Co-ordinates

Another interesting geometrical visualization technique is the use of **Andrew's curves** (Hoffmann & Grinstein, 2002). This method plots each data point by applying a transformation of the form

$$\boldsymbol{F(t) = \frac{X_1}{\sqrt{2}} + X_2 . \sin t + X_3 . \cos t + X_4 . \sin 2t + X_5 . \cos 2t + \cdots}$$

(Eqn. 1)

where *t* goes from –π to π and X_1, X_2, etc. are the columns (i.e., variables) of data. One Andrews curve is generated for each row of data (Figure 11). The advantage of this algorithm is that it is easily applied to data with a large amount of dimensions. The disadvantage is the long computational time, as every data point requires the calculation of a trigonometric function.

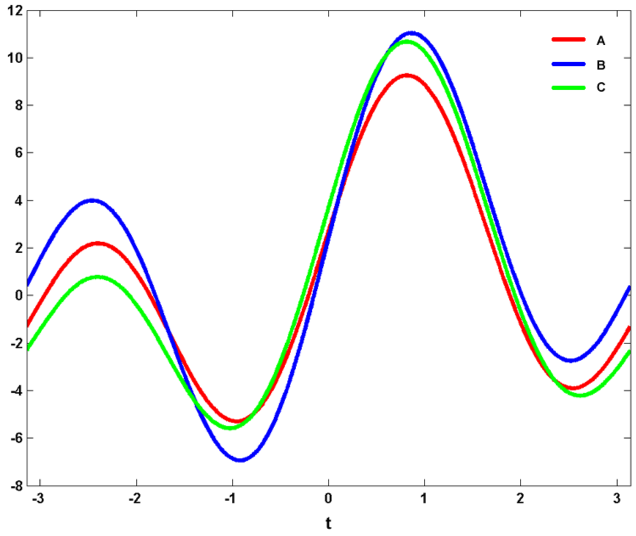


Figure 11 : Andrews Curves (Hoffmann & Grinstein, 2002)

Radial Coordinate Visualization (RadViz) uses the elastic spring paradigm (Hoffmann & Grinstein, 2002) . From a center point n equally spaced limbs of the same length spread out, each representing one dimension. The ends of the lines mark the dimensional anchor (DA) of the respective variable, which are connected forming a circle. Before the data points can be visualized by this technique they need to be normalized. After that, one end of a spring is fastened to each dimensional anchor, the other end to the data point. The spring constant of each spring is the value of the data point of the respective dimension. In order to determine the location of the data point, the sum of the spring forces needs to equal zero. Figure 12 shows the result of this method applied to the well-known Iris data set. An advantage of RadViz is that it preserves certain symmetries of the data set. The major disadvantage is the overlap of points.

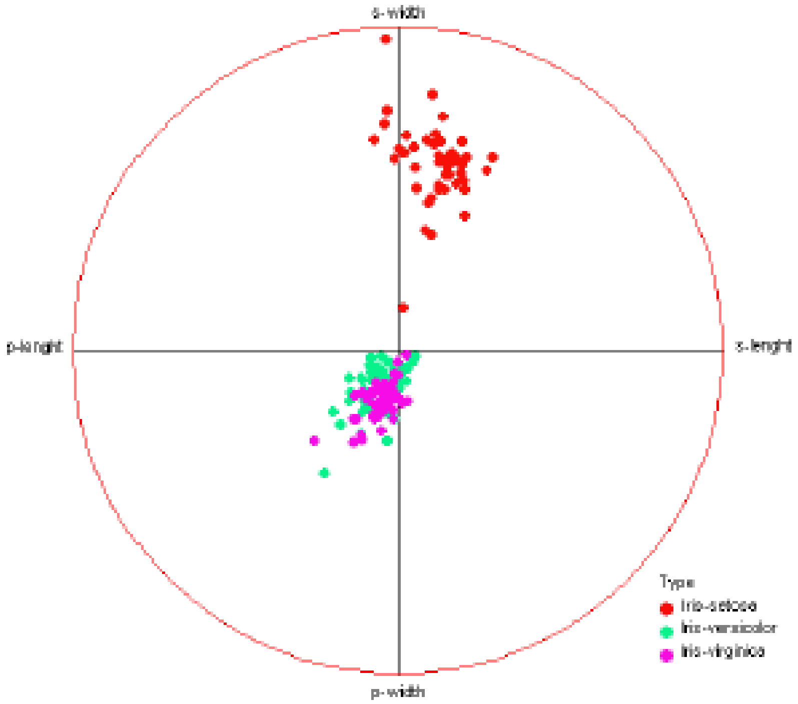


Figure 12 : RadViz of the Iris Data Set (Hoffmann & Grinstein, 2002)

All the techniques explained above allow one to visualize data sets without trying to change them in order to simplify the visualization. Another class of visualization techniques involves non-linear projection methods that reduce the size of the dimension vector (Dimensionality Reduction). These include techniques like Multidimensional scaling which tries to preserve distances between data points , and Self Organizing Maps (SOMs) a method of artificial neural networks that focus on the maintenance of structure (Grinstein, G., Hoffmann, P., Pickett, R., 2002) . Yet another class of techniques rely on pixel oriented visualization where each data value in the data set is represented by one pixel in the display (D. A. Keim, 2000). There are still more techniques of high dimensional visualization that have recently come out of computer and cognitive sciences; however, a review of those is outside the scope of this research.

Due to its simplicity in construction and display, the parallel coordinates plot was selected as the basis for the development of an interactive Graphical User Interface (GUI) used to analyze energy simulation input-output relationships. This is presented in detail in chapter 7.

3.5 Visual Analytics

Historically, analysis techniques such as statistics and data mining developed independently from visualization and interaction techniques (D. A. Keim, Kohlhammer, Ellis, & Mansmann, 2010). However, some innovative insights changed the scope of the fields into what is today called visual analytics research. Statistical data analysis is useful if the relationship between variables is well defined. However, if the analyst does not know what to expect from the data then it often becomes necessary to visually explore the data in order to identify an appropriate statistical analysis technique (J. Haberl & Abbas, 1998). One of the most important developments in early visual analytics was recognizing the need to move from confirmatory data analysis (using charts and other visual representations to just present results) to exploratory data analysis (interacting with the data/results). This idea was first presented to the statistics research community by John W. Tukey (Tukey, 1977).

With improvements in computer graphics software, graphical user interfaces and interaction devices, a growing research community devoted their efforts to information visualization. At some stage, the potential of integrating the user in the knowledge discovery and data mining process through effective visualization techniques and interaction capabilities was recognized and this led to visual data exploration and visual data mining (D. A. Keim, 2001).

Two of the early uses of the term visual analytics can be traced back to the mid-2000s ; first by (Pak Chung Wong & Thomas, 2004) , and a year later in the R&D agenda, Illuminating the Path, which defined Visual Analytics as the science of analytical reasoning facilitated by interactive visual interfaces (Thomas & Cook, 2006) . To be more precise, visual analytics is an iterative process that involves collecting information, data preprocessing, knowledge representation, interaction, and decision making. The ultimate goal is to gain insight into the problem at hand, which may be described by vast amounts of scientific, forensic or business data from heterogeneous sources. To achieve this goal, visual analytics combines the advantages of machines with the strengths of human cognition. While methods from KDD, statistics and mathematics efficiently drive the automatic analysis side, the addition of human capabilities to perceive, relate and conclude have turned visual analytics into a very promising field of research (D. A. Keim, Mansmann, Schneidewind, & Ziegler, 2006).

Information overload is a well-known phenomenon of the present information age. Due to the progress in computer power and storage capacity over the last few decades, data is being produced at an incredible rate and the ability to collect and store data is growing faster than the ability to analyze it. The overarching vision of visual analytics research is to turn this information overload into an opportunity. The transformation of data into meaningful visualizations is a non-trivial task that cannot be automatically improved through steadily growing computational resources alone (D. A. Keim et al., 2006) . Decision-makers should be enabled to examine massive, multi-dimensional, multi-source, time-varying information streams to make effective decisions in time-critical situations (D. A. Keim et al., 2010). The specific advantage of visual analytics is that decision makers may focus their full cognitive and perceptual capabilities on the analytical process, while allowing them to apply advanced computational capabilities to augment the discovery process. Each approach has its advantages and its weaknesses. Whereas algorithms working in isolation can miss out on the “wisdom” that is readily available from human knowledge of the problem and the data, strictly manually guided approaches can easily cause users to lose their way with high dimensional data (U. M. Fayyad et al., 2002). Thus for informed decisions, it is indispensable to combine the flexibility, creativity, and background knowledge of human decision makers with the enormous storage capacity and processing power of today’s computers.

3.5.1 Scope

On a grand scale, visual analytics solutions provide technology that combines the strengths of human and electronic data processing. The challenge is to identify the best automated algorithm for the analysis task at hand, identify its limits which cannot be further automated, and then develop a tightly integrated solution which adequately combines the best automated analysis algorithms with appropriate visualization and interaction techniques (D. A. Keim & Andrienko, 2008). Visualization becomes the medium of a semi-automated analytical process, where humans and machines cooperate using their respective distinct capabilities for the most effective results (D. A. Keim &

Andrienko, 2008). The user has to be the ultimate authority in giving the direction of the analysis along his or her specific task. At the same time, the system has to provide effective means of interaction to concentrate on this specific task. On top of that, in many applications different people work along the path from data to decision. A visual representation will sketch this path and provide a reference for their collaboration across different tasks and abstraction levels. The diversity of these tasks cannot be tackled with a single theory. Visual analytics research is thus highly interdisciplinary and combines various related research areas such as visualization, data mining, data management, data fusion, statistics and cognition science (Figure 13)

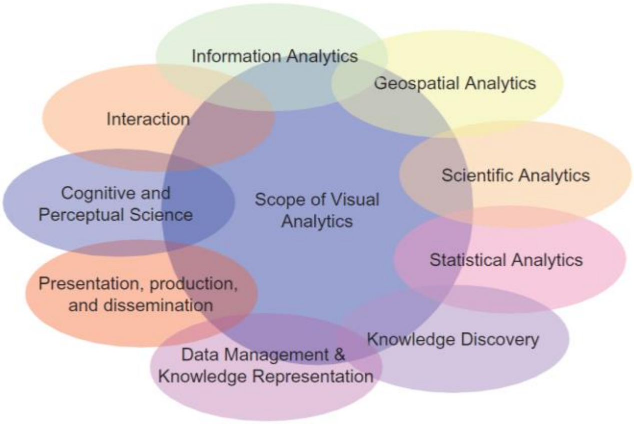

Information Analytics
Geospatial Analytics
Interaction
Scientific Analytics
Cognitive and Perceptual Science
Scope of Visual Analytics
Statistical Analytics
Presentation, production, and dissemination
Knowledge Discovery
Data Management & Knowledge Representation

Figure 13 : Visual Analytics as a highly interdisciplinary field of research (D. A. Keim & Andrienko, 2008)

3.5.2 Visual Analytics versus Information Visualization

Historically, visual analytics evolved out of the fields of information and scientific visualization. However visual analytics is more than just visualization and by definition is an integrated approach combining visualization, human factors and data analysis (D. A. Keim & Andrienko, 2008). The term visualization is meanwhile understood as “a graphical representation of data or concepts" (Ware, 2000). While there is certainly some overlay and some of the information visualization work is certainly highly related to visual analytics, traditional visualization work does not necessarily deal with analysis tasks nor does it always also use advanced data analysis algorithms. Most research efforts in Information Visualization have concentrated on the process of producing views and creating valuable interaction techniques for a given class of data (social network, multi-dimensional data, etc.). However, much less has been suggested as to how user interactions on the data can be turned into intelligence to tune underlying analytical processes. This is one place where Visual Analytics differs most from Information Visualization, namely, it gives higher priority to data analytics from the start and through all iterations of the sense making loop.

3.5.3 The Visual Analytics Process

The visual analytics process combines automatic and visual analysis methods with a tight coupling through human interaction in order to gain knowledge from data. Figure 14 (D. A. Keim et al., 2010) shows an abstract overview of the different stages (represented through ovals) and their transitions (arrows) in the visual analytics process.

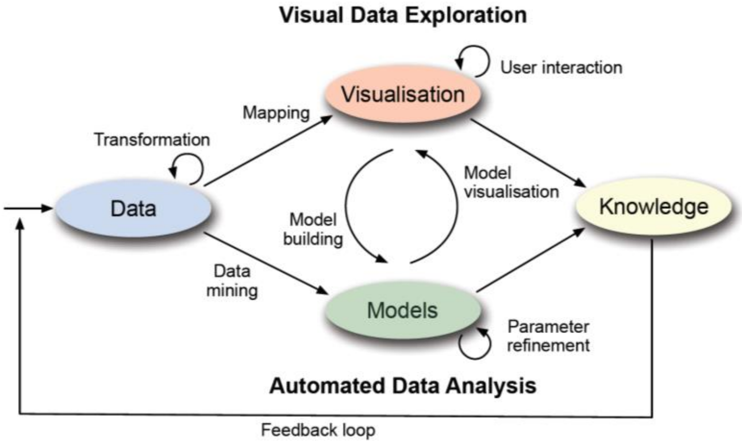


Figure 14 : Visual Analytics Process (D. A. Keim et al., 2010)

In many application scenarios, heterogeneous data sources need to be integrated before visual or automatic analysis methods can be applied. Therefore, the first step is often to preprocess and transform the data to generate different representations for further exploration (as indicated by the Transformation arrow in Figure 14). After the transformation, the analyst may choose between applying visual or automatic analysis methods. If an automated analysis is used first, data mining methods are applied to generate models of the original data. Once a model is created the analyst has to evaluate and refine the model, which can best be done by interacting with the data. Visualizations allow the analysts to interact with the automated methods by modifying parameters or selecting other analysis algorithms. Model visualization can then be used to evaluate the findings of the generated models. Alternating between visual and automatic methods is characteristic of the visual analytics process and leads to a continuous refinement and verification of preliminary results (D. A. Keim et al., 2010). The feedback loop stores this knowledge of insightful analysis and assists the analyst in drawing faster and better conclusions in the future.

3.5.4 Applications

Visual Analytics is a highly application oriented discipline driven by practical requirements. Visual analytics is essential in application areas where large information spaces have to be processed and analyzed. Major application fields are physics and astronomy. Monitoring climate and weather is also a domain which involves huge amounts of data collected by sensors throughout the world and from satellites at short time intervals. A visual approach can help to interpret these massive amounts of data and to gain insight into the dependencies of climate factors and climate change scenarios that would otherwise not be easily identified.

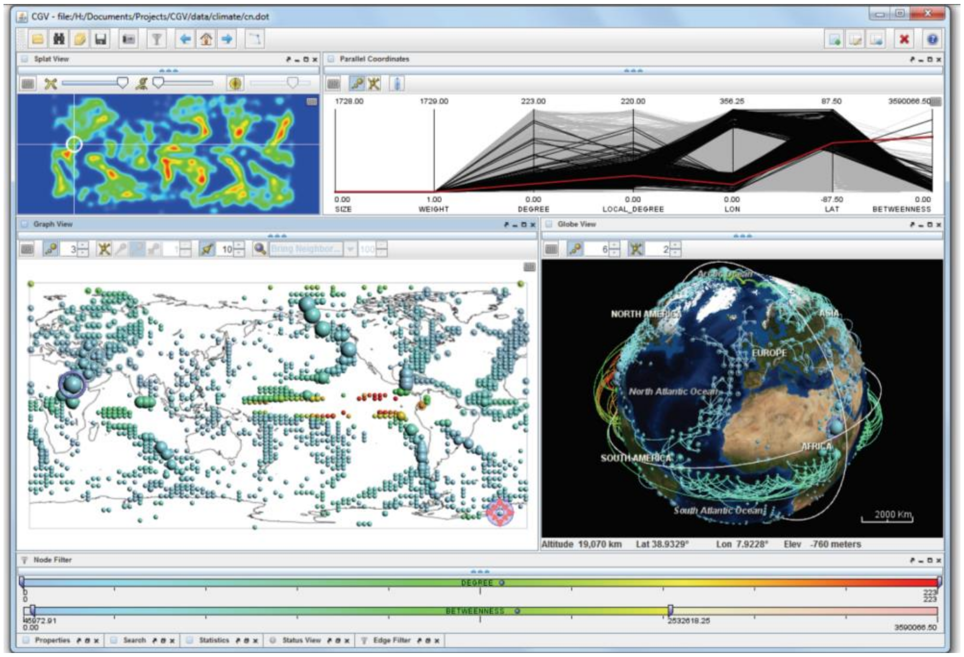


Figure 15 : Visual Analytics in Action: Simulation of Climate Models (Tominski, Abello, & Schumann, 2009)

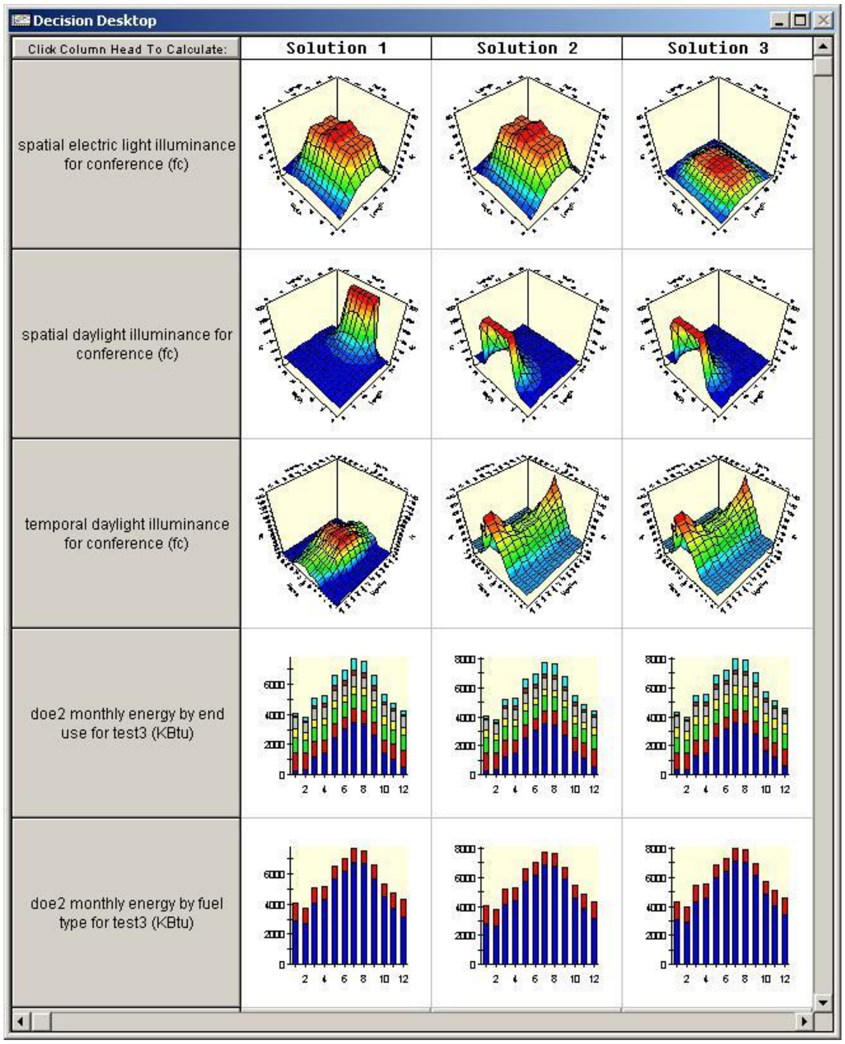
Figure 15 represents visual support for the simulation of climate models provided by CGV (Coordinated Graph Visualization), a highly interactive graph visualization system

(Tominski et al., 2009). The area of bio-informatics uses visual analytics techniques to analyze large amounts of biological and medical data. Another major application domain for visual analytics is business intelligence.

In the domain of building energy analysis there has been research done in the use of machine learning techniques such as neural networks and support vector machines for automated fault detection analysis and prediction of building energy consumption (Zhao & Magoulès, 2012). This aspect has been discussed earlier in section 3.3. There is also a history of R&D on visualization techniques (J. Haberl & Abbas, 1998) and graphical user interfaces (Papamichael, 1999) for the analysis of vast amounts of simulation outputs. However this present study did not find any commercial or research tool in the field of building energy simulation that effectively combines both (data mining and visualization) in the true definition of visual analytics as discussed previously in this chapter.

3-D surface plots were used to view small differences between the simulated data and the measured data for non-weather dependent loads (J. S. Haberl, Bronson, Hinchey, & O'Neal, 1993). For weather dependent loads carpet matrix plots were used to detect different trends between DOE-2 simulations and measured consumption. While these techniques do assist the building energy analyst to review large amounts of building energy consumption data for errors or to establish time and temperature related trends, the maximum number of dimensions (variables) that could be accommodated at a time in a single display is still limited to four – three axial and one using color. The fourth dimension could also be time, as illustrated by (J. Haberl, Sparks, & Culp, 1996) in their use of animated (time sequenced) displays of energy use data. These techniques are rather limiting for the visual analysis of higher dimensional datasets involved in energy simulations as well as usage data recorded by sensors and BMS at short time intervals.

There are very few examples of graphical user interfaces that allow a designer to use simulation results for design synthesis. One interesting prototype is the Building Design Advisor (BDA), a software environment developed at Lawrence Berkeley National Labs (LBNL, 2006), designed to facilitate informed decisions from the early schematic phases of building design to the detailed specification of building components and systems (Papamichael, 1999). To do that, the BDA supports the integrated, concurrent use of multiple simulation tools and databases, and makes their output available in forms that support multicriterion judgment. The BDA provides a graphical user interface that consists of two main elements: the Building Browser and the Decision Desktop. The Decision Desktop allows building designers to compare multiple alternative design solutions with respect to multiple design considerations, as addressed by the analysis and visualization tools and databases linked to the BDA. The parameters displayed in the Decision Desktop are selected in the Building Browser. The Decision Desktop (Figure 16) supports a large variety of data types, including 2-D and 3-D distributions, images, sound and video, which can be displayed and edited in their own windows. The limitation here, again, is from the data mining and knowledge discovery perspective; potentially useful patterns are not highlighted by machine learning algorithms, rather, the user has to discover these through trial and error visual analysis alone.



Decision Desktop

Click Column Head To Calculate:	Solution 1	Solution 2	Solution 3
spatial electric light illuminance for conference (fc)			
spatial daylight illuminance for conference (fc)			
temporal daylight illuminance for conference (fc)			
doe2 monthly energy by end use for test3 (KBtu)			
doe2 monthly energy by fuel type for test3 (KBtu)			

Figure 16 : BDA Decision Desktop (LBNL, 2006)

4: PROPOSED METHODOLOGY

4.1 Visual Analytics Based Decision Support Methodology [VADSM]

This thesis proposes a new methodology called VADSM to facilitate the generation & evaluation of building design alternatives subject to user-defined selection criteria. This methodology is especially pertinent to high performance (low energy) buildings.

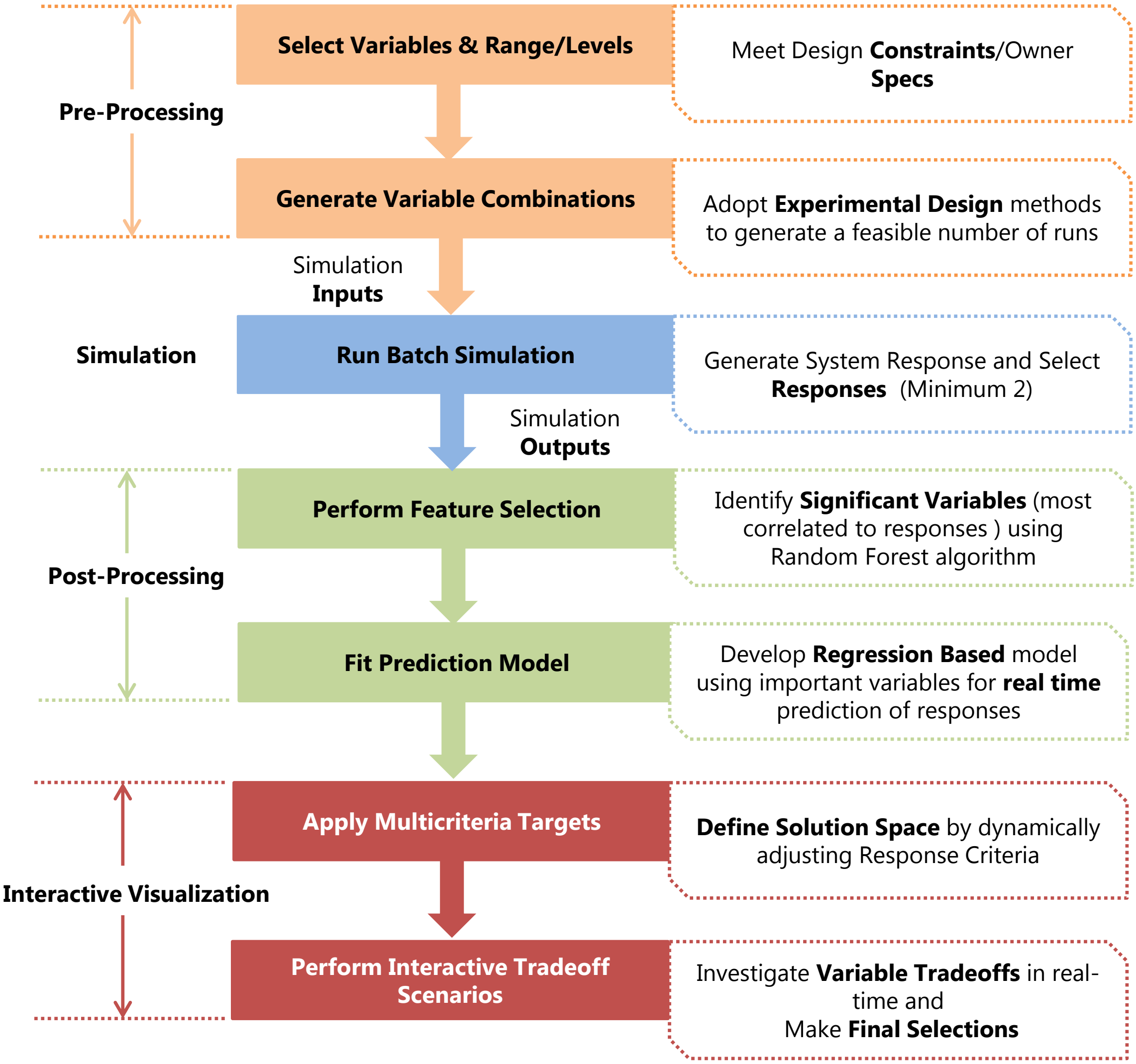

Pre-Processing
Select Variables & Range/Levels
Meet Design Constraints/Owner Specs
Generate Variable Combinations
Adopt Experimental Design methods to generate a feasible number of runs
Simulation Inputs
Simulation
Run Batch Simulation
Generate System Response and Select Responses (Minimum 2)
Simulation Outputs
Post-Processing
Perform Feature Selection
Identify Significant Variables (most correlated to responses) using Random Forest algorithm
Fit Prediction Model
Develop Regression Based model using important variables for real time prediction of responses
Interactive Visualization
Apply Multicriteria Targets
Define Solution Space by dynamically adjusting Response Criteria
Perform Interactive Tradeoff Scenarios
Investigate Variable Tradeoffs in real-time and Make Final Selections

Figure 17 : VADSM Flow Diagram

The VADSM methodology consists of the following four stages:

Stage 1: Pre-processing or selection of independent design variable combinations

The pre-processing stage involves selecting design variables of interest and identifying practical ranges based on the building type, project requirements and owner specifications. If non-linear relationships between predictors and response are suspected then a minimum of three levels for each variable should be selected. Depending on the number of factors and levels, the number of evaluative combinations can range from thousands to millions. An appropriate experimental design technique such as Latin Hypercube sampling is thus necessary to generate a feasible number of runs from a simulation run time perspective. This stage may be considered semi-automated since the variable selection is still largely dependent on the user but the experimental design application is automated.

Stage 2: Simulation based generation of system response

Selected variable combinations can now be input into an hourly building energy simulation program for batch processing. The responses could be direct outputs from the chosen simulation program, such as annual energy use/peak demand or could be derived metrics like energy costs or environmental impacts. A minimum of two responses are required in-order to perform multicriterion satisficing. This stage has the potential to be fully automated provided the chosen simulation tool can handle batch processing and has the necessary input-output interoperability with existing spreadsheet applications or can be connected to online databases for storing simulation results.

Stage 3: Post-processing of simulation result**s**

The post–processing stage involves the application of non-parametric statistical learning techniques such as Random Forest on the simulated data to identify variables that are the best predictors. This approach is also known as Feature Selection. This strategy serves the dual purpose of educating the designer of the important variables and also allows the fitting of simpler regression based models that are easier to visualize and manipulate dynamically than black box machine learning techniques. Once the prediction models (one for each response) have been developed the user can use these instead of the actual simulation tool to make real time predictions within the pre-defined solution space. This stage can be fully automated with minimal user intervention provided reliable model selection parameters are already established.

Stage 4: Interactive Visualization of the solution space

The visual analytics stage allows a user to choose their own design selections supported by interactive visualization of the predictive relationships between selected variables and responses. The Decision Support Model Viewer (DSMV) application has been designed to facilitate this activity. It allows the user to effectively reduce the solution space (simulation space) by dynamically adjusting response criteria. With the criteria in place the users can then investigate variable tradeoffs necessary to meet those constraints and make final selections. Insights into the complex nature of building design suggest that this stage cannot be fully automated, but effective visualization tools can significantly aid in the human decision making process.

4.2 Experimental Design

The purpose of experimental activity is to lead to an understanding of the underlying relationship between input (independent) and output (dependent) variables. Experimental design is the aggregation of independent variables, the set of levels of each independent variable, and the combinations of these levels that are chosen for experimental purposes (Berger & Maurer, 2002). The core of an experimental design is to answer the three-part question; which factors should be studied, how should the levels of these factors vary, and in what way should these levels be combined? Two of the methods most relevant to this research are described below

4.2.1 Central Composite Design

Central Composite Design (CCD) is one of the most widely used experimental design techniques for fitting a second order response surface to estimate nonlinear behavior. There are three varieties: circumscribed, inscribed and face centered. The former two varieties require five levels for each factor while the third one requires three levels. The circumscribed CCD technique has been used for this study since it explores the largest variable space by virtue of its design.

A central composite design has three components (Berger & Maurer, 2002)

- A two level (fractional) factorial design, which estimates the main and two factor interaction terms.
- A “star” or “axial” design which in conjunction with the other two components , helps estimate quadratic terms

- A set of center points, which estimates error and helps estimate surface curvature with more stability. It should be noted that for a computer experiment using a deterministic simulation program (as utilized in this study) replication is not required and only one center point is sufficient.

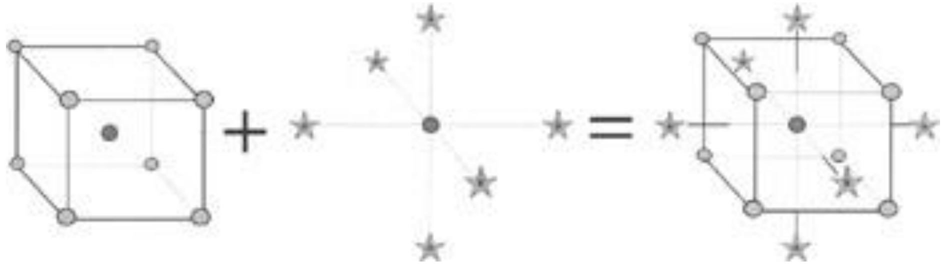


Figure 18 : Circumscribed Central Composite Design (http://www.globalspec.com)

If the distance from the center of the design space to a factorial point is ±1 unit for each factor, the distance from the center of the design space to a star point is $\pm\, \alpha\ with\ |\alpha|\ > 1$. The precise value of α depends on certain properties desired for the design, like orthogonal blocking and on the number of factors involved (Reddy, 2011b). To maintain rotatability, the value of α depends on the number of experimental runs in the factorial portion of the central composite design:

$$\alpha = [number\ of\ factorial\ runs]^{1/4}$$

If the factorial is a full, then

$$\alpha = [2^k]^{1/4}$$

However, the factorial portion can also be a fractional design. The total number of experimental runs for a CCD with k factors is

$$\mathbf{2^{k} + 2k + c}$$

where c is the number of center points.

4.2.2 Latin Hypercube Sampling

Latin hypercube sampling (LHS) is a sampling technique for generating a set of input vectors from a multidimensional distribution (Mckay, Beckman, & Conover, 2000). This sampling method is often used to construct computer experiments for performing sensitivity and uncertainty analysis on complex systems (Helton & Davis, 2003). LHS uses stratified sampling without replacement and can be viewed as a compromise procedure combining many of the desirable features of random and stratified sampling (Reddy, 2011a). A Latin hypercube is the generalization of the Latin square to an arbitrary number of dimensions, whereby each sample is the only one in each axis-aligned hyperplane containing it

Latin hypercube sampling selects *n* different values from each of *k* variables $X_1, \ldots, X_k$ in the following manner. The range of each variable is divided into *n* non overlapping intervals on the basis of equal probability. For each column of *X*, the n values are randomly distributed with one from each interval (0,1/n), (1/n,2/n), ..., (1-1/n,1). The *n* values thus obtained for X_1 are paired in a random manner with the *n* values of X_2. These *n* pairs are combined in a random manner with the *n* values of X_3 to form *n* triplets, and so on, until *n* *k*-tuplets are formed. These *n* *k*-tuplets form the Latin hypercube sample. It is convenient to think of this sample as forming an (*n* × *k*) matrix of input where the i_{th} row contains specific values of each of the *k* input variables to be used on the i_{th} run of the computer model. Refer to (Helton & Davis, 2003) for an exhaustive technical review of LHS and its advantages over other experimental design methods.

4.3 Multiple Linear Regression [MLR]

Multiple linear regression (MLR) is a method used to model the linear relationship between a dependent variable and one or more independent variables. A MLR model is given by

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k + \varepsilon$$

(Eqn. 2)

where, $\beta_0, \beta_1, \beta_2, \ldots, \beta_k$ are the population regression coefficients that have to be estimated. $X_1, \ldots, X_k$ are the independent variables (or regressors) and ε represents a random error component that cannot be explained by the model.

When the number of independent variables, k, is two or more, the (graphical) dimension of the problem increases. The regression ceases to be a line in two dimensional space and becomes instead a hyper-surface in (k+1) dimensional space (Kleinbaum, 2008). The regression equation is the surface described by the mean values of Y at various combinations of X. For the three-dimensional case, the least–squares solution that gives the best fitting plane (Figure 19) is determined by minimizing the sum of squares of the distances between the observed values Y_i and the corresponding predicted values.

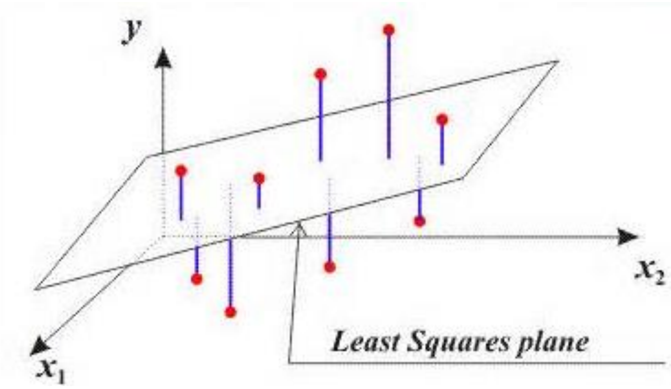


Figure 19 : Best Fitting Plane for Three Dimensional Data

The better the fit the smaller the deviations of observed from predicted values. Thus if

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \cdots + \hat{\beta}_k X_k$$

(Eqn. 3)

denotes the fitted regression model , the sum of squared errors (SSE) or deviations of observed *Y* values from the corresponding values predicted by the fitted regression model is given by

$$SSE = \sum_{i=1}^{n} \left(Y_i - \hat{Y}_i\right)^2 = \sum_{i=1}^{n} \left(Y_i - \hat{\beta}_0 - \hat{\beta}_1 X_{i1} - \cdots - \hat{\beta}_k X_{ik}\right)^2$$

(Eqn. 4)

The least-squares solution then consists of the values $\hat{\beta}_0$, $\hat{\beta}_1$, … , $\hat{\beta}_k$, called the least-squares estimates for which the sum above is a minimum. This approach is known as the ordinary least squares (OLS) method.

The fundamental equation of regression analysis, which holds for any regression situation, is given by

$$\sum_{i=1}^{n}(Y_i - \bar{Y})^2 = \sum_{i=1}^{n}\left(\hat{Y}_i - \bar{Y}\right)^2 + \sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

(Eqn. 5)

where $\overline{Y}$ and $\hat{Y}_i$ denote the mean and fitted values of Y_i .

4.3.1 Assumptions of MLR

Existence: For each specific combination of values of the independent variables, *Y* is a univariate random variable with a certain probability distribution having finite mean and variance.

Independence: The *Y* observations are statistically independent of one another. This condition also applies to the *X* values. Correlated regressors (multi-collinearity) lead to unstable and biased regression coefficients.

Linearity: The mean value of Y for each specific combination of $X_1, X_2 \ldots, X_k$ is a linear function of the regression coefficients ($\beta_0, \beta_1, \beta_2, \ldots, \beta_k$).

Homoscedasticity: The variance of Y is the same for any fixed combination of $X_1, \ldots, X_k$. In general, mild departures do not have significant adverse effects.

***Normality*:** For any fixed combination of $X_1, X_2 \ldots, X_k$, the variable *Y* is normally distributed.

In other words,

$$Y \sim N\left(\mu_{Y|X\,,}\,\sigma^2\right)$$

Or equivalently,

$$\varepsilon \sim N(0, \sigma^2)$$

The assumption is that the random error component has a normal distribution with mean 0 and variance σ^2. The assumption of a Gaussian distribution is needed to justify the use of procedures of statistical inference involving the *t* and *F* distributions.

4.3.2 Model Evaluation

The most widely used measure of model accuracy or goodness-of-fit is the coefficient of determination or R^2 where $0 \leq R^2 \leq 1$

R^2 = SSR (explained variation of *Y*) / SST (total variation of *Y*)

R^2 is a misleading statistic since it does not account for the number of degrees of freedom and increases as additional variables are included even if these variables have very little explicative power. A more desirable goodness-of-fit measure is the correct or adjusted R^2 computed as,

$$Adj\ R^2 = 1 - (1 - R^2)\frac{n-1}{n-k}$$

(Eqn. 6)

where n is the total number of observations, and k is the number of model parameters. A widely used estimate of the magnitude of the absolute error of the model is the root mean square error (RMSE), defined as

$$RMSE = \sqrt{\frac{SSE}{n-k}}$$

(Eqn. 7)

The RMSE is an absolute measure with the same units as the Y variable. A normalized measure is often more appropriate. Such a measure is the coefficient of variation of the RMSE (CVRMSE or simply CV) defined as

$$CV = \frac{RMSE}{\bar{Y}}$$

(Eqn. 8)

The F-statistic, which tests for significance of the overall regression model (goodness-of-fit), is defined as:

$$F = \frac{SSR}{SSE} \cdot \frac{n-k}{k-1}$$

(Eqn. 9)

4.3.3 Model Parsimony using Stepwise Regression

It is best to select the model that yields a reasonably high "goodness-of-fit" for the fewest model parameters .This is referred to as model parsimony (Reddy, 2011c). This approach helps reduce the multi collinearity problem due to correlated and potentially redundant regressors. Among the different methods that can be used to select a parsimonious model, stepwise regression is one of the most effective.

The stepwise regression procedure combines elements of both backward elimination and forward selection (Dielman, 2001). It begins with forward selection by examining the list of all possible regressors in simple regressions and choosing the one with the largest partial *F* statistic. The next most highly correlated predictor to the response is identified, given the current variable already in the regression equation. This variable is then allowed to enter the equation and the parameters re-estimated along with the goodness-of-fit. Any parameter that is not statistically significant is removed from the equation. This process continues until no more variables "enter" or "leave" the regression equation. The stepwise technique helps to identify some important variables but doesn't necessarily produce the best regression equation (Dielman, 2001). The final decision on model section requires the judgment of the model builder, and on mechanistic insights into the problem.

4.4 Classification and Regression Trees (CART)

Predictors like linear or polynomial regression are global models, where a single predictive formula is supposed to hold over the entire data space. When the data has features which interact in complicated, nonlinear ways or when the solution space has regions with abrupt ridges and discontinuities, assembling a single global model may not be satisfactory. Some of the non-parametric smoothers try to fit models locally and then merge them together, but such models can be hard to interpret. An alternative approach to nonlinear regression is to sub-divide, or partition, the space into smaller regions, where the interactions are more manageable. These regions can then be partitioned further into smaller sub-divisions in a recursive manner, as in hierarchical clustering, until finally the regions can be fit with simple models (constant or linear). The global model thus has two parts: one is the recursive partition; the other is a simple local model for each cell of the partition.

Recursive partitioning is a stage wise process that sequentially breaks the data up into smaller and smaller pieces. This is initiated by a two-step search method. First, for each split (value) of a given predictor, a sum of squares of the response is computed within each of the two splits and added. Their sum will be equal to or less than the original sum of squares for the response variable. The “best” split for each predictor is defined as the split that reduces the sum of squares the most. Second, with the best split of each predictor determined, the best split overall is determined using the same sum of squares criteria. By selecting the best split overall, the best predictor by the sum of squares criteria is implicitly chosen. The result is a recursive partitioning of the data that can be represented within a basis function framework. The basis functions are indicator variables defined by the best splits (Berk, 2008) . This two-step search procedure is easily generalized so that the response variable can be categorical or numeric, and in its most visible implementation, the recursive partitioning is called Classification and Regression Trees (CART). CART was developed by Leo Breiman (Breiman et al., 1984) over three decades ago and remains a popular data analysis tool.

4.4.1 Regression Trees

Regression Trees are a subset of CART and are used to predict a continuous response. Consider a regression problem with continuous response Y and inputs X_1 and X_2, each taking values in the unit interval. The CART algorithm selects a variable and corresponding split-point to achieve the best fit and splits the space into two regions, and models the response by the mean of Y in each region. Then one or both of these regions are split into two more regions, and this process is continued, until some stopping rule is applied. For example in Figure 20 the first split is at $X_1 = t_1$. Then the region $X_1 \leq t_1$ is split at $X_2 = t_2$ and the region $X_1 > t_1$ is split at $X_1 = t_3$ and so on. The result of this process is a partition into the five regions $R_1 \ldots R_5$.

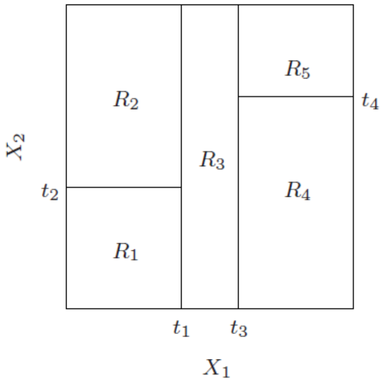


Figure 20 : Recursive Binary Partitions (Hastie et al., 2009)

The corresponding regression model predicts Y with a constant c_m in region R_m that is,

$$\hat{f}(X) = \sum_{m=1}^{5} c_m \, I \, \{(X_1, X_2) \in R_m\}$$

(Eqn. 10)

Where, $$c_m = \text{avg}(y_i | x_i \in R_m)$$

This is called a piecewise constant model. This same model can be represented by the binary tree on the left panel in Figure 21 . A key advantage of the recursive binary tree is its interpretability. The feature space partition is fully described by a single tree .The full dataset sits at the top of the tree in the root node. Observations satisfying the condition at each junction or branch node are assigned to the left branch, and the others to the right branch. The terminal nodes or leaves of the tree correspond to the regions $R_1 \ldots R_5$. The right panel of Figure 21 is a perspective plot of the regression surface from this model (Hastie et al., 2009).

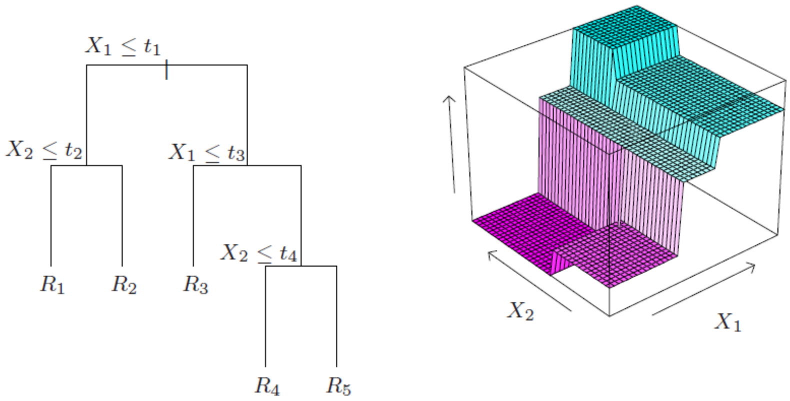


Figure 21 : Binary Regression Tree Diagram and Regression Surface Plot (Hastie et al., 2009)

4.4.2 Regression tree growing Algorithm

The sum of squared errors for a tree *T* is

$$S = \sum_{c \in leaves\,(T)} \sum_{i \in c} (y_i - m_c)^2$$

(Eqn. 11)

where,

$$m_c = \frac{1}{n_c} \sum_{i \in c} y_i$$

is the prediction of leaf *c* with n_c data points.

The basic tree growing algorithm is as follows:

Step 1: Start with a single node containing all points. Calculate m_c and S

Step 2: If all the points in the node have the same value for all the input variables, stop. Otherwise, search over all binary splits of all variables for the one which will reduce S as much as possible. If the largest decrease in S would be less than some threshold δ, or one of the resulting nodes would contain less than q points, stop. Otherwise, take that split, creating two new nodes.

Step 3: In each new node, go back to step 1.

The most critical aspect in the tree growing algorithm is the stopping criteria. Selecting the right size tree is a matter of balancing the bias-variance tradeoff. Larger trees fit the data closely with fewer data points in terminal nodes implying a high variance model while shallower trees will be affected by model bias. The most widely used strategy to constrain the size of a tree is called "pruning". The pruning process removes undesirable branches by combining nodes that do not reduce heterogeneity sufficiently for the extra complexity added. The process starts at the terminal nodes and works back up the tree until all of the remaining nodes are satisfactory. K-fold cross validation (generally k =10) is used to prune extra nodes that do not help improve the generalization error.

4.4.3 Variable Importance with CART

Estimates of predictor importance for a regression tree are calculated by summing changes in the mean squared error (MSE) due to splits on every predictor and dividing the sum by the number of branch nodes. This sum is generally taken over the best splits found at each branch node unless the tree is grown with surrogate splits, in which case the sum is taken over all splits at each branch node including surrogate splits. A detailed explanation of surrogate splits and variable importance is provided by (Breiman et al., 1984). At each node, MSE is estimated as node error weighted by the node probability. The probability of a node is computed as the proportion of observations from the original data that satisfy the conditions for that node. Variable importance associated with a split is computed as the difference between MSE for the parent node and the total MSE for the two children.

4.4.4 Instability of Individual Trees

One major problem with trees is their high variance. Often a small change in the data can result in a very different series of splits, making interpretation somewhat precarious. The major reason for this instability is the hierarchical nature of the process: the effect of an error in the top split is propagated down to all of the splits below it. One can alleviate this to some degree by trying to use a more stable split criterion, but the inherent instability is not removed. It is the price to be paid for estimating a simple, tree-based structure from the data. Random Forest, an ensemble of trees, utilizes *Bagging* (explained below) to average predictions across many trees to reduce this variance.

4.5 Random Forests [RF] – An Ensemble of Binary Trees

Since CART was introduced, it has been widely used in statistical data analysis. It has many good properties; it handles all types of data in regression and classification problems and deals with missing values effectively. It is appropriate to use in high dimensional and large data sets since it is highly resistant to irrelevant feature variables and computationally efficient. It also provides some insights into which variables are important and where, by virtue of providing a visual tree. But CART often has a higher error rate than other methods such as SVMs or Boosting and is unstable in the sense that if the training data is changed a little bit, it can change a lot. Ensemble methods were introduced to improve weak and unstable predictors such as CART.

Leo Breiman attempted to improve methods such as CART and pointed out that unstable predictors can be stabilized by making many predictions using multiple weak learners that together constitute an ensemble learner (Breiman, 1998). Bagging (Bootstrap Aggregating) is an ensemble method refined from that idea. It generates multiple trees by making bootstrap replicates of the original data and using them as new training data sets to construct trees. Because about 2/3 of the original data are used to construct each tree, the performance of each tree is relatively worse than a tree built with the original training data set. But by averaging predictions across those trees, the variance of the final ensemble gets smaller and often results in significant accuracy improvement (Breiman, 1996) . The computational cost of Bagging, however, is high and the performance of Bagging is often worse than other machine methods such as SVMs and Boosting, especially when the dimension of the feature space is large. Additionally, since bagging involves all the predictors for each tree split it can be adversely affected by correlated predictors or a few dominant predictors.

The success and some drawbacks of Bagging rapidly inspired a huge amount of work on various different ensemble methods to improve CART. Several researchers suggested the random selection of a subset of features for each tree split or even randomly choosing a split from k best splits (Bae, 2008). (Breiman, 2001) further investigated the ideas of Bagging and random feature selection and developed a new algorithm for classification and regression; Random Forests (RF). RF works by building an ensemble of decision trees on bootstrapped samples wherein each tree split is chosen from a limited set of randomly selected features. Since it includes many trees, this ensemble is called a forest.

Since RF was suggested by Breiman, it has received much attention due to its remarkable empirical success. Breiman showed that the accuracy of RF is as good as or sometimes better than that of SVMs (Breiman, 2001). One of the reasons why RF is so effective for complex response functions is that it capitalizes on very flexible fitting procedures that can respond to highly local features of the data. Such flexibility is desirable because it can substantially reduce the bias in the fitted values compared to the fitted values from parametric regression .The flexibility in RF comes in part from individual trees that can find nonlinear relationships and interactions. Another source of the flexibility is large trees that are not precluded from having very small sample sizes in their terminal nodes. RF consciously address over-fitting by using OOB observations (explained below) to construct the fitted values and measures of fit and by averaging over trees. Yet another source of flexibility is the random sampling of predictors. This strategy allows predictors that work well, but only for a very few observations, the opportunity to participate. This also reduces competition between correlated predictors, and given a large enough number of trees each gets a chance to contribute. This two part strategy – flexible fitting functions and averaging over OOB observations is highly effective and has the potential to break the bias-variance tradeoff (Berk, 2008).

4.5.1 RF Algorithm

Let $D_n = \{(\mathbf{X}_i, Y_i) : i = 1, \dots\dots, n\}$ $where\, \mathbf{X}_i = \left(X_i^{(1)}, \dots\dots,\ X_i^{(d)}\right) \in \mathbf{R}^d, Y_i \in \mathbf{R}$ be the i.i.d. training data set. Then the Random Forest algorithm suggested by Breiman is constructed as follows (Bae, 2008):

Step 1: Draw *K* independent bootstrap samples $B_k, k = 1, \dots., K$ from D_n, where $|B_i| = n$. Note that each B_k consists of n samples chosen randomly from D_n with replacement and |A| is the number of elements in set A.

Step 2: For each $B_k, k = 1, \dots., K$, grow a tree with following rules.

2.1 At each node, randomly select a subset of *F* variables from *d* variables, where $F \leq d$ is a tuning parameter in the Random Forests algorithm.

2.2 At each node, find the best split (feature variable and split point) among the *F* variables chosen at 2.1.

2.3 Grow trees to a maximum depth without pruning. That is, grow trees until each terminal node contains no more than 5 training data observations in regression and until each terminal node contains data with same class in classification.

2.4 Let $f\ (x\ , D_n\ , \boldsymbol{\theta}_k)$ be the resulting tree predictor where x is a set of feature variables, $\boldsymbol{\theta}_k$ is a randomly chosen variable consisting of subsets of feature variables, split points at each node and B_k . Thus $\boldsymbol{\theta}_j\ , j = 1, \dots, K$ are identical independent distributed random variables

Step 3: Define the final Random Forests predictor $f\ (x\ , D_n)$ as

$$f\ (x\ , D_n)\ \ =\ \frac{1}{K}\sum_{k=1}^{K} f\ (x\ , D_n\ , \boldsymbol{\theta}_k)$$

(Eqn. 12)

4.5.2 Out of Bag Observations and forecasting error

In random forests, there is no need for cross-validation or a separate test set to get an unbiased estimate of the forecasting or test error. When sampling randomly from a set of observations to generate a bootstrap training sample for a single tree an average of 36.8% of the observations are not used for building that individual tree. These observations are considered “out of the bag” or OOB for that tree. The accuracy of a random forest’s prediction can be estimated from these OOB data as

$$\mathbf{OOBMSE} = \boldsymbol{\frac{1}{n}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} (\boldsymbol{y_i} - \overline{\boldsymbol{\hat{y}_{\imath\,\mathrm{OOB}}}})^{\mathbf{2}}$$

(Eqn. 13)

Where $\overline{\boldsymbol{\hat{y}_{\imath\,\mathrm{OOB}}}}$ denotes the average prediction for the *i*th observation from all trees for which this observation has been OOB, *n* is the data size.

4.6 Predicator importance with RF

In many statistical learning applications the goal is not only to achieve high prediction accuracy but also to understand the underlying mechanism, or in other words explore how inputs are related to outputs. Finding relevant variables may be one of the ways to understand this. RF provides two approaches to assess predictor importance.

4.6.1 Contribution to Model Fit

One approach to measuring predictor importance is to record the decrease in fitting measure (ex. Gini Index) each time a given variable is used to define a split. The sum of these reductions for a given tree is a measure of importance for the variable when the tree is built. For RF one can average this measure of importance over the set of trees. However, reductions in the fitting criteria ignore the forecasting skill of a model since the fit measures are computed with the training data and not the test data (OOB Data). If one cannot forecast well it means that the model cannot usefully reproduce the empirical world. Moreover it can be difficult to translate contributions to fit statistics into practical terms.

4.6.2 Contributions to Forecasting Skill

(Breiman, 2001) has suggested another approach based on the reduction of predictive accuracy when a predictor is randomly shuffled. The shuffling makes that predictor on the average unrelated to the response and all other predictors. In contrast to fit statistics, forecasting skill has direct implications for actual decisions and can be translated into practical terms. The measure of variable importance is based on the difference between predictive performance of the ensemble on the original data set and the performance on the *modified* data set in which an algorithm randomly permutes values of the observed attribute between examples (Figure 22). By measuring the performance before and after the described modification for each tree in the forest, the algorithm combines these differences into an importance estimate. In the RF framework, the most widely used score of importance of a given variable is the increase in the mean error of a tree (mean square error for regression and misclassification rate for classification) in the forest when the observed values of this variable are randomly permuted in the OOB samples (Genuer, Poggi, & Tuleau-Malot, 2010).

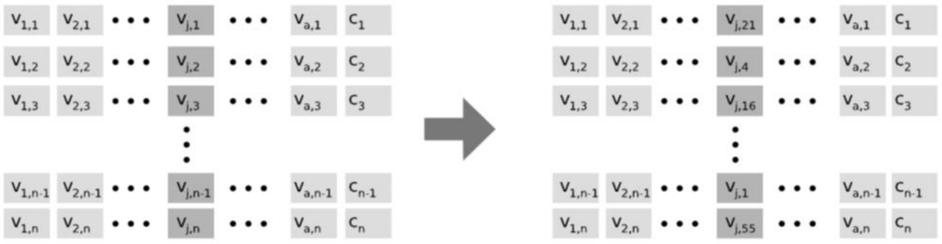

$v_{1,1}$ $v_{2,1}$ ••• $v_{j,1}$ ••• $v_{a,1}$ c_1
$v_{1,2}$ $v_{2,2}$ ••• $v_{j,2}$ ••• $v_{a,2}$ c_2
$v_{1,3}$ $v_{2,3}$ ••• $v_{j,3}$ ••• $v_{a,3}$ c_3
$v_{1,n-1}$ $v_{2,n-1}$ ••• $v_{j,n-1}$ ••• $v_{a,n-1}$ c_{n-1}
$v_{1,n}$ $v_{2,n}$ ••• $v_{j,n}$ ••• $v_{a,n}$ c_n
$v_{1,1}$ $v_{2,1}$ ••• $v_{j,21}$ ••• $v_{a,1}$ c_1
$v_{1,2}$ $v_{2,2}$ ••• $v_{j,4}$ ••• $v_{a,2}$ c_2
$v_{1,3}$ $v_{2,3}$ ••• $v_{j,16}$ ••• $v_{a,3}$ c_3
$v_{1,n-1}$ $v_{2,n-1}$ ••• $v_{j,1}$ ••• $v_{a,n-1}$ c_{n-1}
$v_{1,n}$ $v_{2,n}$ ••• $v_{j,55}$ ••• $v_{a,n}$ c_n

Figure 22 : Randomly permuting values of the attribute v_j

Permutation-based MSE reduction has been adopted as the state-of-the-art approach for variable ranking by various authors (Grömping, 2009; Ishwaran, Kogalur, Blackstone, & Lauer, 2008). It is determined as follows: For tree *t*, the OOB mean squared error is calculated as the average of the squared deviations of OOB responses from their respective predictions:

$$\mathbf{OOBMSE}_{t} = \frac{1}{n_{\mathrm{OOB}}\,,t} \sum_{i=1:i\,\epsilon\,OOB_t}^{n} (y_i - \hat{y}_{i,t})^2$$

(Eqn. 14)

where the ˆ indicates predictions, OOB$_t$ = { *i* : observation *i* is OOB for tree *t*}, that is, summation is done over OOB observations only, and n_{OOB},t is the number of OOB observations in tree *t*. If regressor $\boldsymbol{X_j}$ does not have predictive value for the response, it should not make a difference if the values for X_j are randomly permuted in the OOB data before the predictions are generated. Thus,

$$\mathbf{OOBMSE}_{\boldsymbol{t}}\,(\boldsymbol{X_j}\text{ permuted}) = \frac{\mathbf{1}}{\boldsymbol{n}_{\mathbf{OOB}}\,,\boldsymbol{t}} \sum_{\boldsymbol{i=1:i\,\epsilon\,OOB_t}}^{\boldsymbol{n}} (\boldsymbol{y_i} - \ \boldsymbol{\hat{y}_{i,t}}(\boldsymbol{X_j}\text{ permuted}))^{\mathbf{2}}$$

(Eqn. 15)

should not be substantially larger than OOBMSE$_t$. For each variable X_j in each tree *t*, the difference [OOBMSE$_t$ (*Xj* permuted) – OOBMSE$_t$] is calculated based on one random permutation of the variable’s out-of-bag data for the tree. This difference is 0 for a variable that happens to be not involved in any split of tree *t*. The MSE reduction according to regressor X_j for the complete forest is obtained as the average over all *ntree* trees of these differences. Variable Importance of $\boldsymbol{X_j}$ is then equal to:

$$\mathbf{VI}\,(\mathbf{X}_{\boldsymbol{j}}) = \frac{1}{ntree} \sum_{t=1}^{ntree} (\,\mathbf{OOBMSE}_{\boldsymbol{t}}\,(\boldsymbol{X_j}\text{ permuted}) - \ \mathbf{OOBMSE}_{\boldsymbol{t}}\,)$$

(Eqn. 16)

One can standardize the above equation by computing its standard deviation over the *ntree* trees. The result can then be interpreted as a z-score so that importance measures are now all on the same scale (Berk, 2008). It is sometimes possible for forecasting accuracy to improve slightly when a variable is shuffled because of the randomness introduced. A negative measure of forecasting importance follows ,which can be treated as no decline in accuracy or can simply be ignored (Berk, 2008).

4.7 RF Tuning parameters

Despite the complexity of the RF algorithm and the large number of potential tuning parameters, most of the usual defaults work well in practice. The tuning parameters most likely to require some manipulation are the following:

4.7.1 Node size

Unlike in CART, the number of observations in the terminal nodes of each tree in RF can be very small. Software packages like Matlab and R use the default of 5 for regression and 1 for classification .The goal is to grow trees with as little bias as possible. The high variance of individual trees that would result can be tolerated because of the averaging over a large number of such trees.

4.7.2 Number of Trees

The number of trees should be chosen based on the cost of computation. In practice 500 trees are often a good compromise and appear commonly in research. One benefit of a large number of trees is that each predictor will have an ample opportunity to contribute, even if very few are drawn for each split.

4.7.3 Number of Predictors Sampled

Most statistical software applications (R, Matlab) by default take the square root of the total number of variables for classification, and one third the total number for regression. Breiman suggested starting with the defaults and then trying a few more or less. In practice large differences in performance are rarely found and selecting a few predictors each time seem to be adequate provided the number of trees is in the order of 500 or so.

5: DESCRIPTION OF ILLUSTRATED CASE STUDY BUILDING

5.1 Prototype Building Selected

The US Dept. of Energy’s (DOE) Building Technologies Program, working with DOE's National Labs, developed models for 16 commercial building types in 16 locations representing all U.S. climate zones. These 16 building types cover about 70% of the commercial buildings in the United States (NREL, PNNL, & US DoE, 2011). From this list, the medium office prototype was selected as the baseline simulation model for this study. By virtue of its size the medium office was expected to be affected by both envelope and internal loads and hence would allow a wider mix of design variables to be evaluated as compared to the smaller or larger commercial building types which tend to be either envelope or internal load dominated .

The baseline medium office prototype building is a theoretical building modeled with characteristics typical of buildings of this size and use. The building is a 53,600 ft^2 (4,980 m^2) three-story building. The building is rectangular shaped, 164 ft. (50 m) by 109 ft. (33 m) with an aspect ratio of 1.5. The HVAC system consists of Packaged Units with a gas furnace for primary heating. Delivery is via Variable Air Volume terminals which also have electric reheat coils. Building components regulated by ASHRAE Standard 90.1-2004 are assumed to meet the minimum prescriptive requirements of that standard. Components not regulated by Standard 90.1 are assumed to be designed as is standard practice for a medium office building. Standard practice is determined from various sources including a review of the Commercial Buildings Energy Consumption Survey

(CBECS) and the input of various design and construction industry professionals (Thornton, Wang, Lane, Rosenberg, & Liu, 2009).

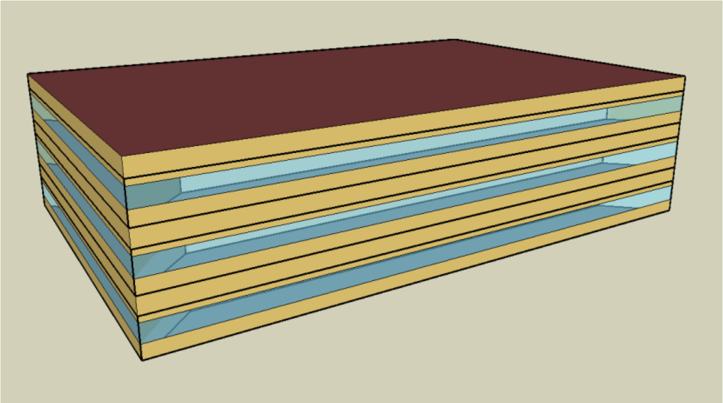


Figure 23 : Axonometric View of the Medium Office Prototype Building

See Appendix A for a score card that summarizes the building descriptions, system characteristics, thermal zones, internal loads, schedules, and other key modeling input information. This can be downloaded as a spreadsheet from DOE's Energy Efficiency and Renewable Energy news site - http://www.energycodes.gov/development/commercial/90.1_models . For an exhaustive review of the medium office prototype building features and energy modeling guidelines refer to a technical report published by PNNL (Thornton et al., 2009)

5.2 Simulation Parameters

5.2.1 Weather File

For this study TMY2 weather data for Oklahoma City, which is categorized as Climate Zone 3A (warm-humid), was used to run simulations on the medium office prototype using eQuest version 3.65. This location was chosen because buildings situated here require both heating and cooling over the year.

5.2.2 Building Simulation Inputs – Independent Variables

Careful selection of input parameters is important for obtaining meaningful results. (Lam & Hui, 1996) performed sensitivity analysis on 60 input parameters relevant for the energy performance of a 40 story office building in Hong Kong. They categorized parameters with significant influence on energy use and demand into three major groups:

Building Load Parameters - occupant density, lighting load and equipment load are the most important. Other significant parameters include design variables of the window system and building envelope.

HVAC System Parameters- summertime thermostat set point, supply fan efficiency and fan static pressure

HVAC Plant Parameters- coefficient of performance (COP) of chillers, chilled water supply temperature, chilled water design temperature difference and chilled water pump impeller efficiency.

(Reddy, Maor, & Panjapornpon, 2007) provide a list of heuristically identified influential parameters that have simple and clear correspondence to specific inputs to the DOE-2 simulation program. Based on the above guidelines a list of 15 independent design variables (Table 1) representing all three major categories (building - system - plant) of interest were chosen for investigation. All the rest of the energy modeling parameters were set to the DoE mid-size commercial prototype description or kept as eQuest defaults where necessary. See appendix B for the complete DoE2 definitions of the selected variables. All the selected variables take numeric values.

Category	Parameters	Abbreviations	Units
Internal Load Variables	Lighting Power Density	LPD	W/ft2
	Eqip Power Density	EPD	W/ft2
Envelope Load Variables	Wall Construction R-Value	Wall-R	h-ft2-°F/Btu
	Roof R-Value	Roof-R	h-ft2-°F/Btu
	Glass U-Value	Win-U	Btu/h-ft2-°F
	Shading Coefficient	SC	Fraction
	Infiltration_AC	Infil-AC	AC-h
	Window Height	Win-Ht	ft
System Variables	Supply Fan Pressure	Fan-Pres	in. of WG
	Min Flow Ratio	Min-FlowR	Fraction
	Min Outdoor Air	Min-OA	Fraction
	Min Cooling Supply Temp	Min-CoolT	°F
	Max Heating Supply Temp	Max-HeatT	°F
Plant Variables	Furnace Eff	Furn-Eff	Fraction
	Cooling EIR	Cool-EIR	Fraction

Table 1 : List of Independent Variables

5.2.3 Simulation Outputs (Response)

The simulation outputs used in this study are annual **Energy Use Index (EUI)** with units of **kBtu/sqft/yr**. and annual **peak electric demand (PED) in kW/yr**. Annual electric consumption figures generated by eQuest in MWh (1000 X kWh) and annual gas consumption figures in MBTUs (1,000,000 x Btu) were converted to like units and added to generate the total annual energy use. This figure was divided by the total area of the building (53,600 sqft.) to get the EUI in kBtu/sqft/yr.

Annual PED for the whole building was reported directly by eQuest as the sum of the 12 monthly coincident peak demand values. Coincident peak demand for each energy end-use is captured at the time the whole building experiences its peak demand. In this case the end use coincident demands were combined to represent a single building total. Electric demand is simulated at an hourly time step by eQuest.

5.3 Applications to Selected Design Feature Sets

Two different feature (design variable) sets were selected to test and contrast the different analytical techniques discussed in Chapter 4. The small feature set is a group of 5 variables selected from the list of 15 simulation inputs presented earlier. The large feature set consists of all 15 variables.

The purpose of picking the small feature set was to test whether traditional techniques like OLS Linear Regression would provide superior results over an inductive statistical learning technique like CART, when dealing with fewer variables. Conversely, a nonlinear technique like RF was expected to be more effective for modeling the higher dimensional feature set that is much sparser and hence technically not a good candidate for fitting a single global linear regression model.

While most design problems tend to involve at least a dozen or more variables, the number of significant variables might subsequently be reduced to no more than 6-7. This is due to the limitation of both conventional visualization techniques as well as human perception limits. Accurate identification of important variables, from a prediction perspective, allows the fitting of simpler models and also simplifies visualization/reporting requirements. For the small feature set this was not critical; however , for the large feature set variable ranking techniques were explored since the chances of redundant variables being included was higher.

For the small feature set with 5 variables a carpet plot matrix of $\left[\binom{5}{2} = 10\right]$ subplots can depict all possible variable pairs. However such traditional 2D or even 3D visualization techniques such as response surface plots are inadequate for joint visualization of larger variable sets. The large feature set required the evaluation of high dimensional data visualization techniques from the domain of visual data mining .One such technique has been incorporated into a GUI presented in section 6.2.4.

Lastly, with the small feature set an exhaustive combination of variables could be simulated for analysis. However, with the larger feature set this was not feasible from a simulation run time perspective and appropriate experimental design techniques were explored to sample fewer but representative runs from the solution space.

5.3.1 Small Feature Set – 5 Design Variables

From the list of 15 simulation inputs introduced earlier, a subset of five variables having 5 discreet numeric levels each (Table 2) were selected as independent inputs to the baseline energy model. An exhaustive combination of $\left(5^{\mathrm{k}} = 3125\right)_{\mathrm{k}=5}$ runs was generated for simulation. The simulated responses are EUI in kBtu/sqft/yr. and PED in kW/yr. Approximately 20% of the data was randomly set aside for model testing. Two contrasting modeling techniques, OLS Regression and Regression Tree (CART) were employed on the data to determine important variables and create models for real time prediction without having to re-run the simulation engine.

A variety of OLS regression models were successively built, involving linear, quadratic and cross terms. Stepwise regression was then employed to retain important variables and identify a parsimonious model. Alternatively, individual regression tree models (for different terminal node sizes) were trained on the simulated data set .The visual tree outputs generated were used to assess variable importance, study variable interactions and make predictions.

Regressors		Levels				
		1	2	3	4	5
Factors	LPD	1.5	1.325	1.15	0.975	0.8
	SC	0.7	0.575	0.45	0.325	0.2
	Ewall R	27	22.2	17.4	12.6	7.8
	Win U	1	0.815	0.63	0.445	0.26
	Win H	7.575	6.06	4.545	3.03	1.515

Table 2: Small Feature Set (5 Levels Each)

With the intent of reducing simulation run time, Central Composite Design was employed to generate $(2^k + 2k + 1 \ = 43\ )_{k=5} \ = 43$ variable combinations. However, only OLS Regression models were built on this reduced dataset since the runs were deemed too few to effectively train a statistical learning model like CART. These OLS models have been compared with the ones generated using the exhaustive variable combination and the results are discussed later in this is study.

5.3.2 Large Feature Set – 15 Design Variables

All 15 variables were utilized for this analysis. A minimum of three levels for each variable (Table 3) were retained to capture any quadratic behavior. However, even with three levels an exhaustive combination of the 15 variables would lead to $3^{15} \sim 14\ x\ 10^{6}$ combinations; an impractical number of simulations. An experimental design technique was essential to select fewer runs while ensuring stratified (representative) sampling of the variable space. **Latin Hypercube Sampling (LHS)** was used to generate a relatively sparse 15000 variable combinations for simulation. See Section 4.2.2 for an overview of LHS. The simulated responses are EUI in kBtu/sqft/yr. and PED in kW/yr.

Regressors		Levels		
		Low	Mid	High
Factors	LPD	0.80	1.40	2.00
	EPD	0.80	1.00	1.20
	Wall-R	7.80	17.40	27.00
	Roof-R	15.00	22.50	30.00
	Win-U	0.25	0.74	1.22
	SC	0.16	0.55	0.93
	Infil-AC	0.20	0.60	1.00
	Win-Ht	1.52	4.55	7.58
	Fan-Pres	1.50	2.75	4.00
	Min-FlowR	0.30	0.65	1.00
	Min-OA	0.10	0.30	0.50
	Min-CoolT	50.0	57.5	65.0
	Max-HeatT	85.0	102.5	120.0
	Furn-Eff	1.25	1.40	1.54
	Cool-EIR	0.359	0.405	0.450

Table 3 : Large Feature Set (15 Variables)

A RF ensemble of 500 regression trees was then utilized to generate variable rankings for both responses. It should be noted that the choice of range for each variable is influential for predictor ranking using RF. For example, if a designer overly restricts the range of variability of an otherwise influential parameter, that variable can turn out to be insignificant. Once the best predictive variables were ordered according to importance, several OLS regression models were built on the top ranked 5-8 variables. Two best OLS models (one for each response) were selected based on $Adj\ R^2$ and CV.

The variable ranking and model building process discussed above has been automated using a software application specially developed for the purpose of this study. The GUI of this software application has been designed to allow users to vary inputs to the selected OLS models for real time predictions, while simultaneously setting constraints on the responses in order to perform multicriterion satisficing what-if scenarios. This is described in detail in the following chapter.

6: RESULTS OBTAINED WITH THE CASE STUDY BUILDING

6.1 Small Design Feature Set

The small feature set is a matrix of dimension 3125 x 5 (Simulation Runs x Predictors). See section 5.3.1 for an overview of the analysis performed. 625 rows of data were randomly set aside for model testing (Test Data) while the remaining 2500 (Training Data) were utilized to build the OLS and CART models. A separate dataset of 43 simulation runs, derived using CCD, was used to for building OLS models only. The same Test Data was used to validate these models as well.

6.1.1 Simulated Building Energy Response

The simulated responses are EUI in kBtu/sqft/yr. and PED in kW/yr. Figure 24 shows the frequency distribution of the simulated EUI across all 3125 runs. The EUI falls in the expected range of 40-60 for this type of building. The Y –Axis represents the number of runs for each EUI bin. Figure 25 shows the corresponding distribution for PED. There is a strong linear correlation between the two responses as can be observed in Figure 26.

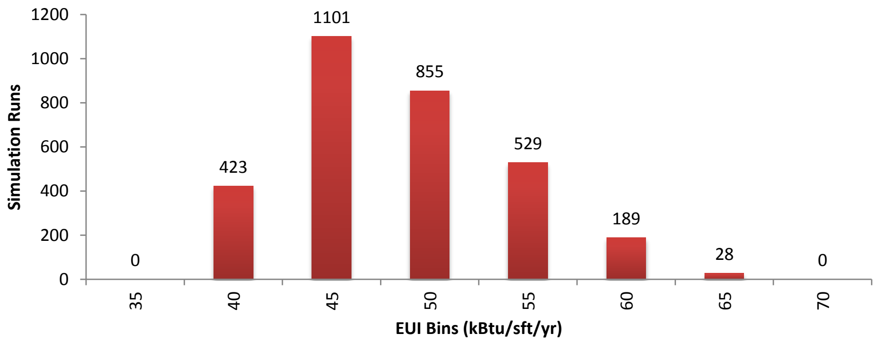


Figure 24 : Frequency Histogram of Simulated EUI

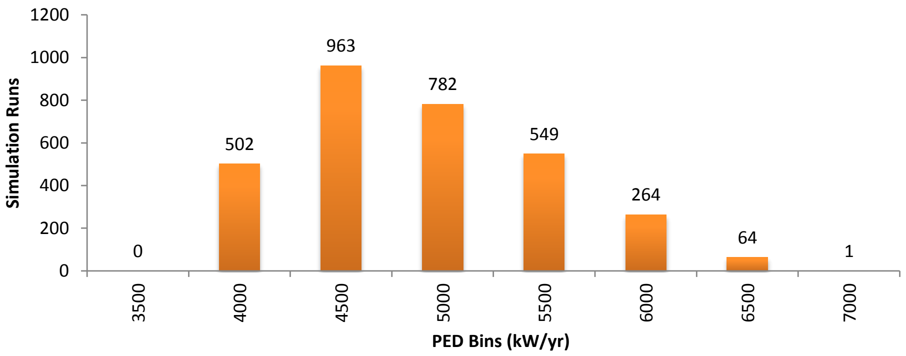


Figure 25 : Frequency Histogram of Simulated PED

	EUI	PED
	kBtu/sft/yr	kW/yr
Max	63	6513
Median	45	4554
Mean	46	4640
Min	36	3529

Table 4 : Simulated Response Descriptive Statistics (5 Vars)

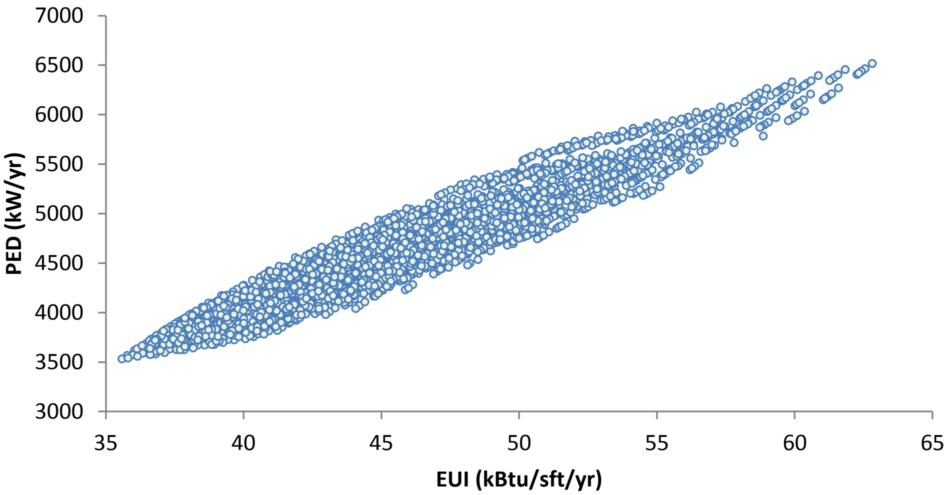


Figure 26 : Simulated Response Scatter Plot

6.1.2 Linear Regression – Exhaustive Variable Combinations

Several OLS regression models were built for both EUI and PED prediction using linear, interaction, and quadratic terms. The models were validated using the Test Data. The results are presented in Table 5 and Table 6 for EUI and PED respectively. Stepwise regression was utilized for model parsimony. However, stepwise regression was not able to conclusively identify important predictors and in all cases retained all variables from the full model. In a simplified strategy the variables were ranked by regressing them individually (Models 1-5). WinH appeared to be the dominant predictor for EUI followed by SC and WinU. For PED only WinH and WinU were important. Based on this, models involving the top 2 and 3 variables (Models 6-9) were built. For EUI prediction, Model 9, involving WinH, SC and WinU was found to have good overall fit (R^2 ~ 92.5%) as well as predictive ability (test CV of 3.29%) .

Model	Mdl Name	Adj R2	RMSE	CV	# of Model Terms	Predictors	Test Data Validation	
							RMSE	CV
1	WallR	0.40%	5.45	11.87%	2	1		
2	LPD	6.66%	5.27	11.49%	2	1		
3	WinU	10.40%	5.17	11.26%	2	1		
4	SC	12.97%	5.09	11.10%	2	1		
5	WinH	62.91%	3.32	7.25%	2	1		
6	2Var-Full	79.47%	2.47	5.39%	6	2	2.40	5.20%
7	2Var-Stepwise	79.48%	2.47	5.39%	5	2	2.40	5.19%
8	3Var-Full	92.46%	1.50	3.27%	10	3	1.52	3.29%
9	3Var-Stepwise	92.46%	1.50	3.27%	8	3	1.52	3.29%
10	AllVar-Linear	93.60%	1.38	3.01%	6	5	1.38	2.98%
11	Linear-Stepwise	93.60%	1.38	3.01%	6	5	1.38	2.98%
12	Interactions	99.46%	0.40	0.87%	16	5	0.42	0.90%
13	Interac-Stepwise	99.46%	0.40	0.87%	14	5	0.42	0.90%
14	PureQuadratic	93.76%	1.36	2.97%	11	5	1.38	2.98%
15	Full	99.60%	0.34	0.75%	21	5	0.36	0.77%
16	Full-Stepwise	99.60%	0.34	0.75%	20	5	0.36	0.77%

Table 5 : Regression Model Results for EUI Prediction

For PED prediction, the 3 variable model had excellent fit (R^2 ~ 97%) and a test CV of only 2.3%. Models built using all 5 variables (Models 10-16) were found to be progressively better with the addition of higher order terms and the full variable models had an $R^2 > 99\%$ and CV < 1%.

Model	Mdl Name	Adj R2	RMSE	CV	# of Model Terms	Predictors	Test Data Validation	
1	WallR	1.03%	612.28	13.21%	2	1		
2	LPD	1.32%	611.38	13.19%	2	1		
3	SC	4.01%	603.00	13.01%	2	1		
4	WinU	22.97%	540.16	11.66%	2	1		
5	WinH	63.73%	370.68	8.00%	2	1	**RMSE**	**CV**
6	2Var-Full	90.91%	185.52	4.00%	6	2	198.98	4.26%
7	2Var-Stepwise	90.92%	185.48	4.00%	5	2	198.82	4.26%
8	3Var-Full	97.03%	106.14	2.29%	10	3	108.29	2.32%
9	3Var-Stepwise	97.03%	106.12	2.29%	9	3	108.17	2.32%
10	AllVar-Linear	93.50%	156.90	3.39%	6	5	152.89	3.28%
11	Linear-Stepwise	93.50%	156.90	3.39%	6	5	152.89	3.28%
12	Interactions	99.27%	52.64	1.14%	16	5	54.35	1.16%
13	Interac-Stepwise	99.27%	52.62	1.14%	14	5	54.19	1.16%
14	PureQuadratic	93.86%	152.46	3.29%	11	5	151.47	3.25%
15	Full	99.60%	38.80	0.84%	21	5	40.77	0.87%
16	Full-Stepwise	99.60%	38.78	0.84%	18	5	40.67	0.87%

Table 6 : Regression Model Results for PED Prediction

6.1.3 Linear Regression – Central Composite Design

A similar approach as described in section (7.1.2) was adopted for the CCD dataset and multiple models were developed. The models were validated using the same Test Data. The results are presented in Table 7 and Table 8 for EUI and PED respectively. Here too models involving the top 3 predictors displayed acceptable fit and predictive abilities. Similar to the results found with the exhaustive dataset, models built on all 5 variables were found to be progressively more accurate with the addition of higher order terms and the full models had an $R^2 > 99\%$ and CV < 2%.

Model	Mdl Name	Adj R2	RMSE	CV	# of Model Terms	Predictors	Test Data Validation	
1	WallR	-2.07%	3.34	7.29%	2	1		
2	LPD	4.16%	3.24	7.07%	2	1		
3	WinU	6.73%	3.20	6.97%	2	1		
4	SC	14.27%	3.07	6.68%	2	1		
5	WinH	65.46%	1.95	4.24%	2	1	**RMSE**	**CV**
6	2Var-Full	81.62%	1.42	3.10%	6	2	2.41	5.22%
7	2Var-Stepwise	81.73%	1.42	3.09%	3	2	2.66	5.75%
8	3Var-Full	91.30%	0.98	2.13%	10	3	1.56	3.37%
9	3Var-Stepwise	91.97%	0.94	2.05%	5	3	1.68	3.64%
10	AllVar-Linear	98.12%	0.45	0.99%	6	5	1.39	3.01%
11	Linear-Stepwise	98.12%	0.45	0.99%	6	5	1.39	3.01%
12	Interactions	99.88%	0.12	0.25%	16	5	0.57	1.23%
13	Interac-Stepwise	99.89%	0.11	0.24%	9	5	0.57	1.24%
14	PureQuadratic	97.91%	0.48	1.04%	11	5	1.40	3.02%
15	Full	99.97%	0.06	0.13%	21	5	0.55	1.18%
16	Full-Stepwise	99.97%	0.06	0.12%	13	5	0.53	1.15%

Table 7 : Regression Model Results for EUI Prediction (CCD)

Model	Mdl Name	Adj R2	RMSE	CV	# of Model Terms	Predictors	Test Data Validation	
1	WallR	-1.51%	364.49	7.88%	2	1		
2	LPD	-1.34%	364.18	7.87%	2	1		
3	SC	2.20%	357.75	7.73%	2	1		
4	WinU	21.71%	320.10	6.92%	2	1		
5	WinH	67.56%	206.05	4.45%	2	1	**RMSE**	**CV**
6	2Var-Full	69.96%	198.28	4.29%	6	2	293.26	6.29%
7	2Var-Stepwise	71.50%	193.11	4.18%	3	2	270.20	5.79%
8	3Var-Full	97.36%	58.81	1.27%	10	3	155.32	3.33%
9	3Var-Stepwise	97.32%	59.18	1.28%	6	3	189.67	4.07%
10	AllVar-Linear	98.19%	48.65	1.05%	6	5	153.75	3.30%
11	Linear-Stepwise	98.19%	48.65	1.05%	6	5	153.75	3.30%
12	Interactions	99.62%	22.17	0.48%	16	5	67.58	1.45%
13	Interac-Stepwise	99.65%	21.43	0.46%	9	5	70.11	1.50%
14	PureQuadratic	98.18%	48.75	1.05%	11	5	151.53	3.25%
15	Full	99.94%	8.95	0.19%	21	5	57.80	1.24%
16	Full-Stepwise	99.94%	8.62	0.19%	15	5	57.52	1.23%

Table 8 : Regression Model Results for PED Prediction (CCD)

6.1.4 Regression Tree Models – All Predictors

Along the lines of model parsimony for linear regression, a tradeoff between predictive accuracy and visual interpretability is also necessary to select a regression tree model. The depth of the tree is a function of the minimum leaf size, i.e. the minimum number of data observations allowed for each terminal node. Figure 27 depicts the effect of minimum leaf size on the number of Tree splits (measure of complexity) and Test Data CV% (predictive accuracy) for both EUI and PED prediction. A smaller leaf size implies finer partitioning of the data and hence a better fitting model (more accuracy), but this is achieved only by increasing the number of splits in the tree (more complexity). For example increasing the min leaf size from 10 to 100 (X-Axis) reduced the tree splits by approximately a factor of 10; however, the prediction CV% more than doubled.

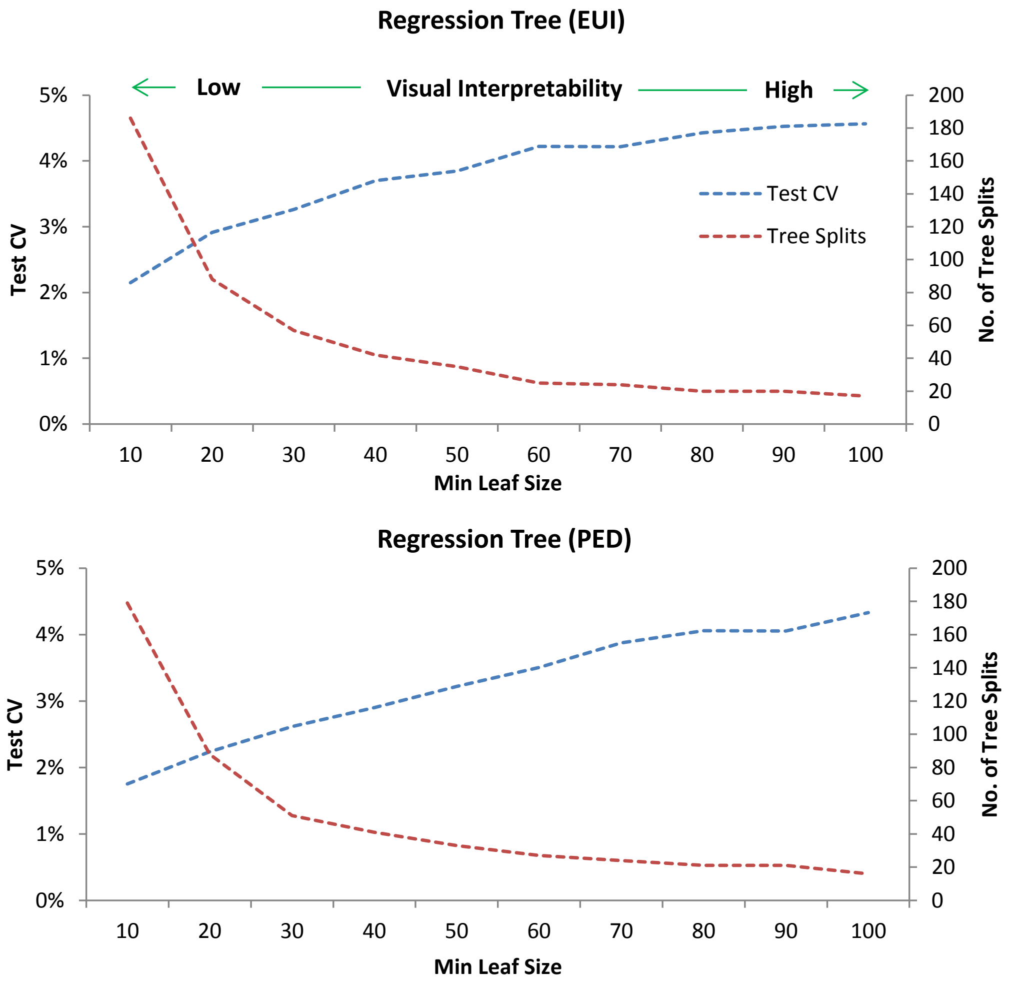


Figure 27 : Tradeoff between Complexity and Accuracy of Regression Trees

Based on the results shown in Figure 27 a minimum leaf size of 100 was deemed appropriate for the given dataset in order to generate trees for both responses. Figure 28 depicts the EUI prediction tree which has 14 splits (levels), and is thus visually simple. The tree was tested using the separate Test Data and a CV of 4.69% was achieved. A lower CV could be attained by decreasing the min leaf criteria; however as pointed out earlier the tree will grow deeper and be difficult to interpret. Figure 29 depicts the PED prediction tree which has 15 splits (levels) and a test CV of 4.35%. It should be noted that both trees were pruned to be within 1 standard error of the minimum cost tree. This is standard practice used to reduce the generalization error through cross validation. Refer to section 4.4.2 for an overview of pruning. A choice of tree size will depend on the intended use of the model. If only prediction is required then a deep tree may be acceptable, however, if the tree is meant to be a visual aid as well, then a compromise has to be reached between predictive accuracy and visual interpretability.

A visual inspection of the trees reveals an inherent variable ranking derived from the order in which the variables appear in the tree. In both cases WinH is clearly the dominant predictor. Although the training data contained all 5 predictors, not all variables were used for building the trees. For example, in the EUI tree (Figure 28) WallR is not used at all while in the PED tree (Figure 29) LPD has been eliminated. Thus, the tree model is helpful as a visual aid for identifying important predictors.

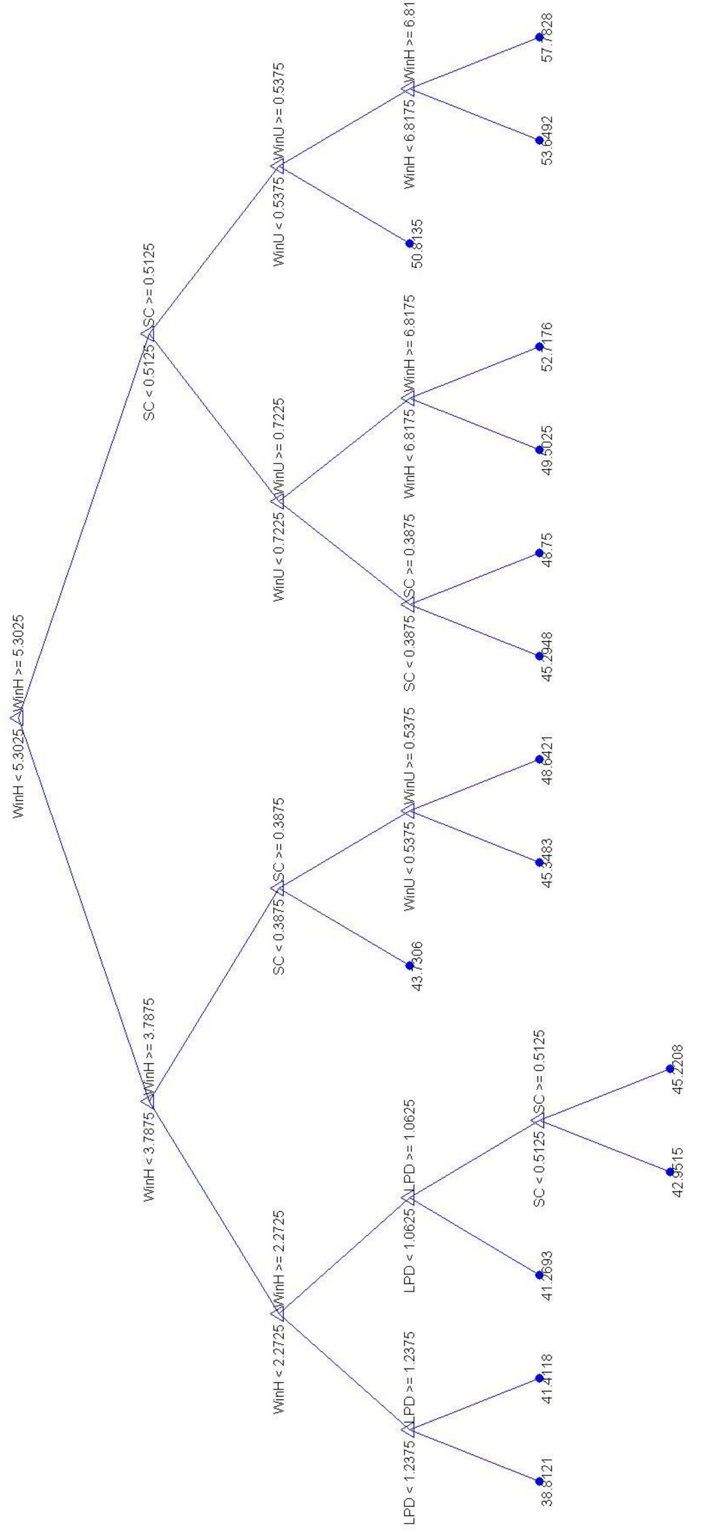


Figure 28 : EUI Prediction Tree_5 Variables (Test CV 4.69%)

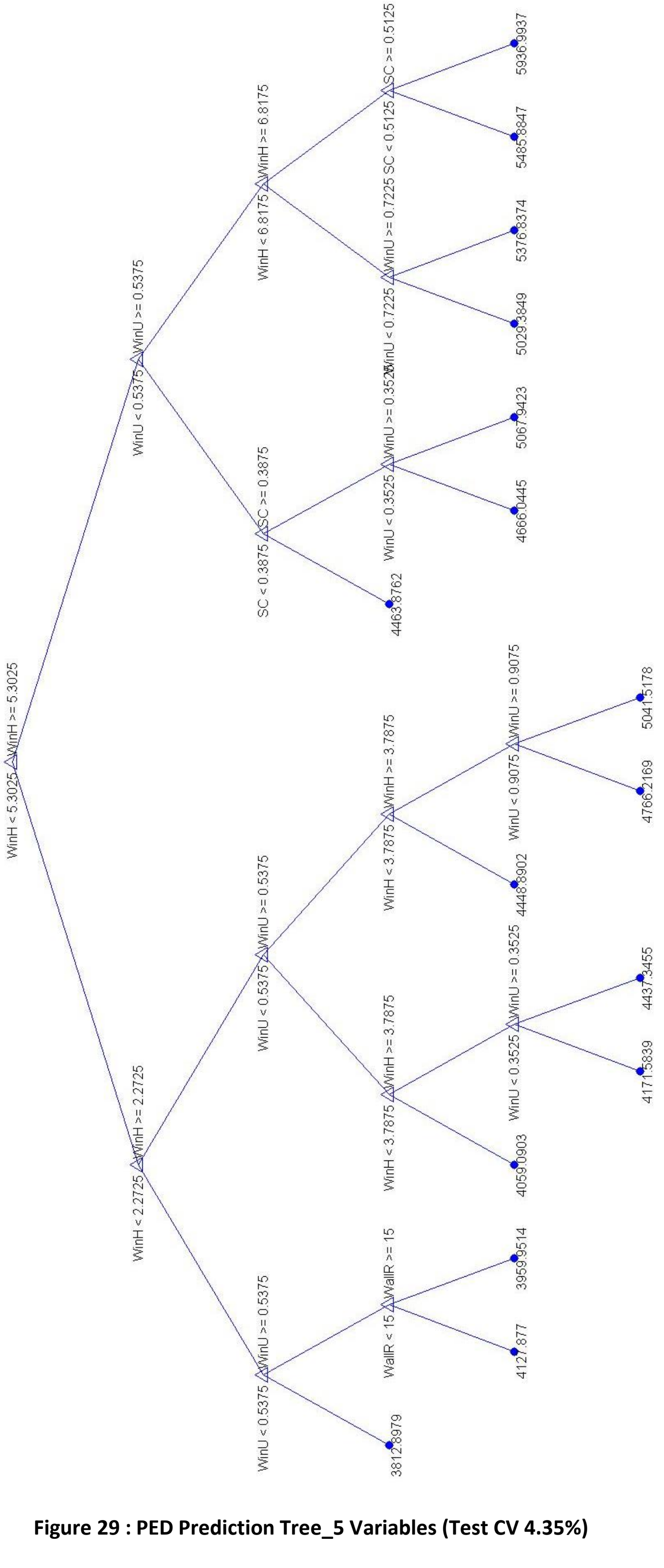


Figure 29 : PED Prediction Tree_5 Variables (Test CV 4.35%)

6.1.5 Regression Tree Models – Best Predictors

As discussed in the previous section, variable importance can be visually determined from a tree; however, it is not always a reliable technique since multiple variables may occupy the same level on different nodes. Alternatively CART provides a more rigorous method for computing variable importance based on a reduction of mean squared error due to splits on each variable. See section 4.4.3 for an explanation of the ranking process.

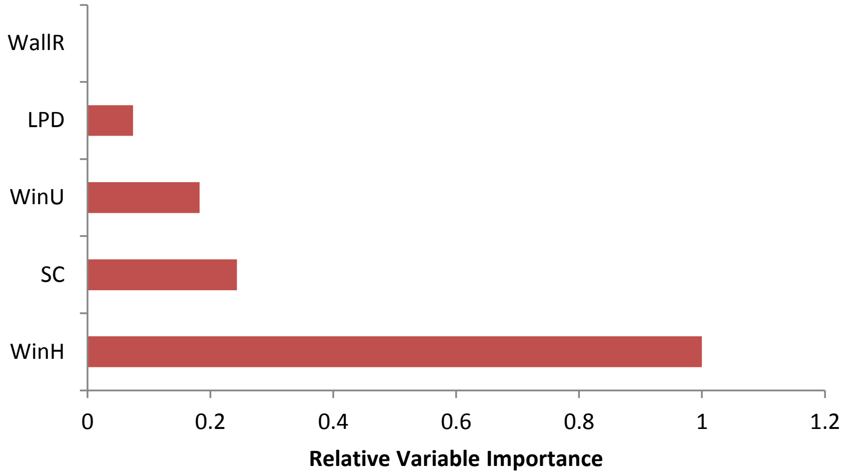


Figure 30 : CART Predictor Ranking for EUI

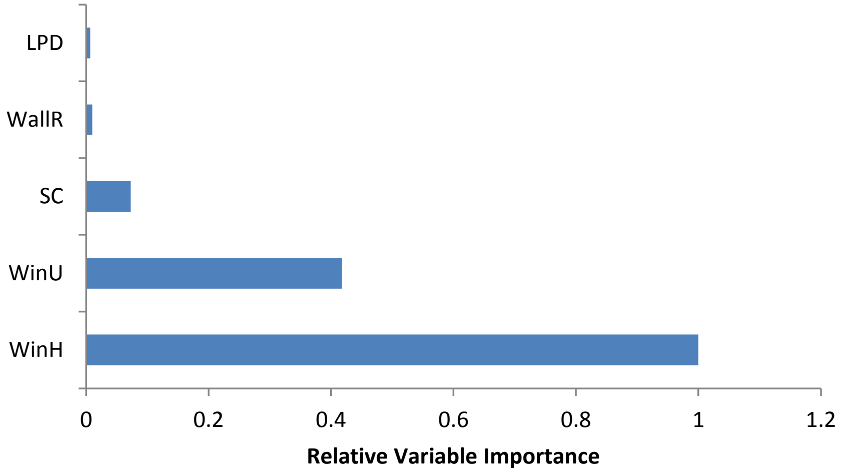


Figure 31 : CART Predictor Ranking for PED

Figure 30 and Figure 31 show the relative predictor rankings (scaled to the maximum) for EUI and PED respectively. WinH is clearly the most important for both responses. In the case of PED the only other variable that appears to have any influence is WinU. The other three have little to no impact. For EUI, both SC and WinU have similar standing, although compared to WinH they are relatively much less important. The predictor ranking provided an opportunity for further simplification of the tree model by retaining only the top 2-3 variables, provided a Test CV < 5% could be achieved.

Figure 32 is the EUI prediction tree built only on the three variables WinH, WinU and SC. Compared to the tree (Figure 28) trained on all 5 variables this tree has the same number of levels and slightly poorer Test CV (4.78%). This suggests that there is not much additional benefit to using this tree model. For PED the 3 variable tree model (Figure 33) has one less level than the 5-variable model (Figure 29). The test CV is again slightly poorer at 4.41% compared to 4.35% for the 5-variable tree. A second PED model trained only on WinH and WinU data is shown in Figure 34. This tree has four levels less than the 5-variable model and the Test CV is ~ 5%. This one might be a reasonable substitute for the 5-varaible model due to its simplicity and acceptable predictive ability.

By tweaking the min leaf size, applying pruning criteria and using variable importance ranking, the decision maker can come up with several tree alternatives and select the one that best meets the dual criteria of predictive ability and visual interpretability.

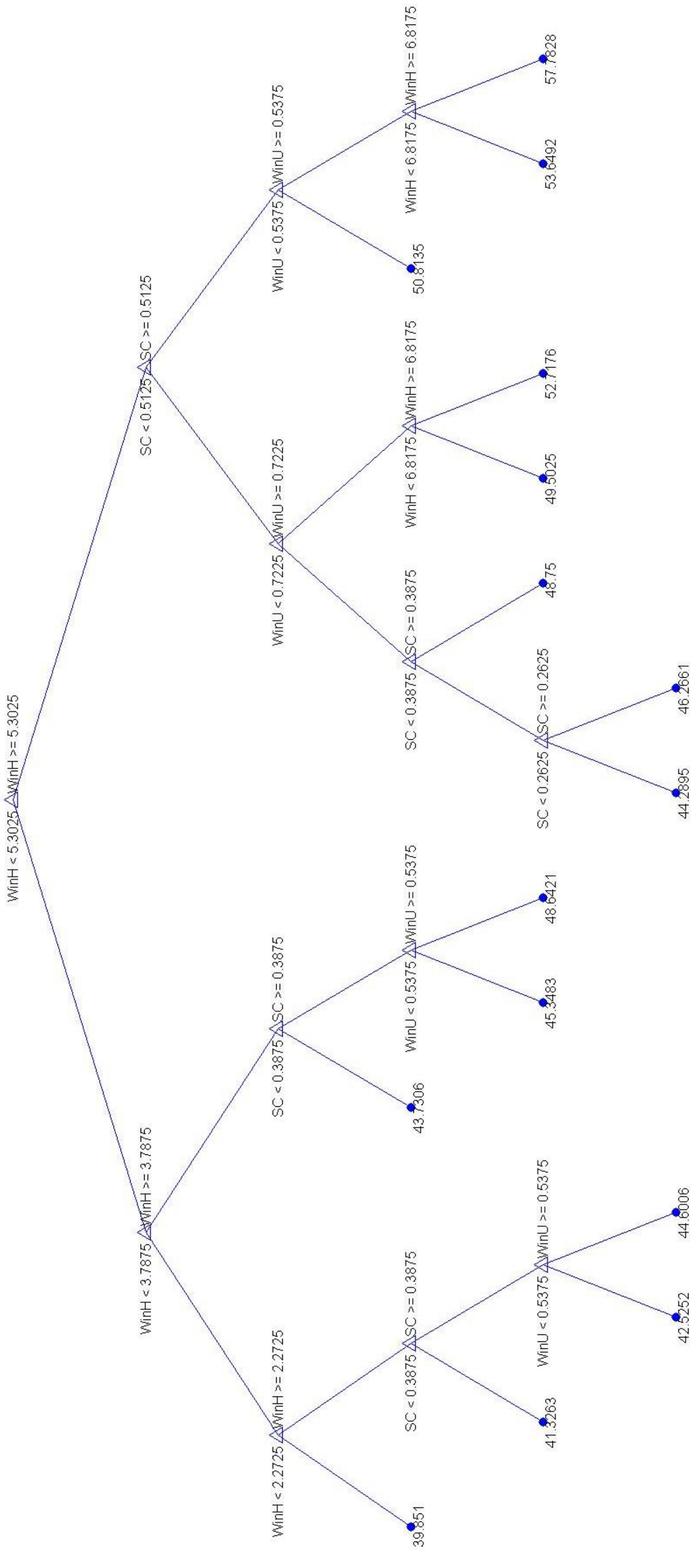


Figure 32 : EUI Prediction Tree_3 Variables (Test CV 4.78%)

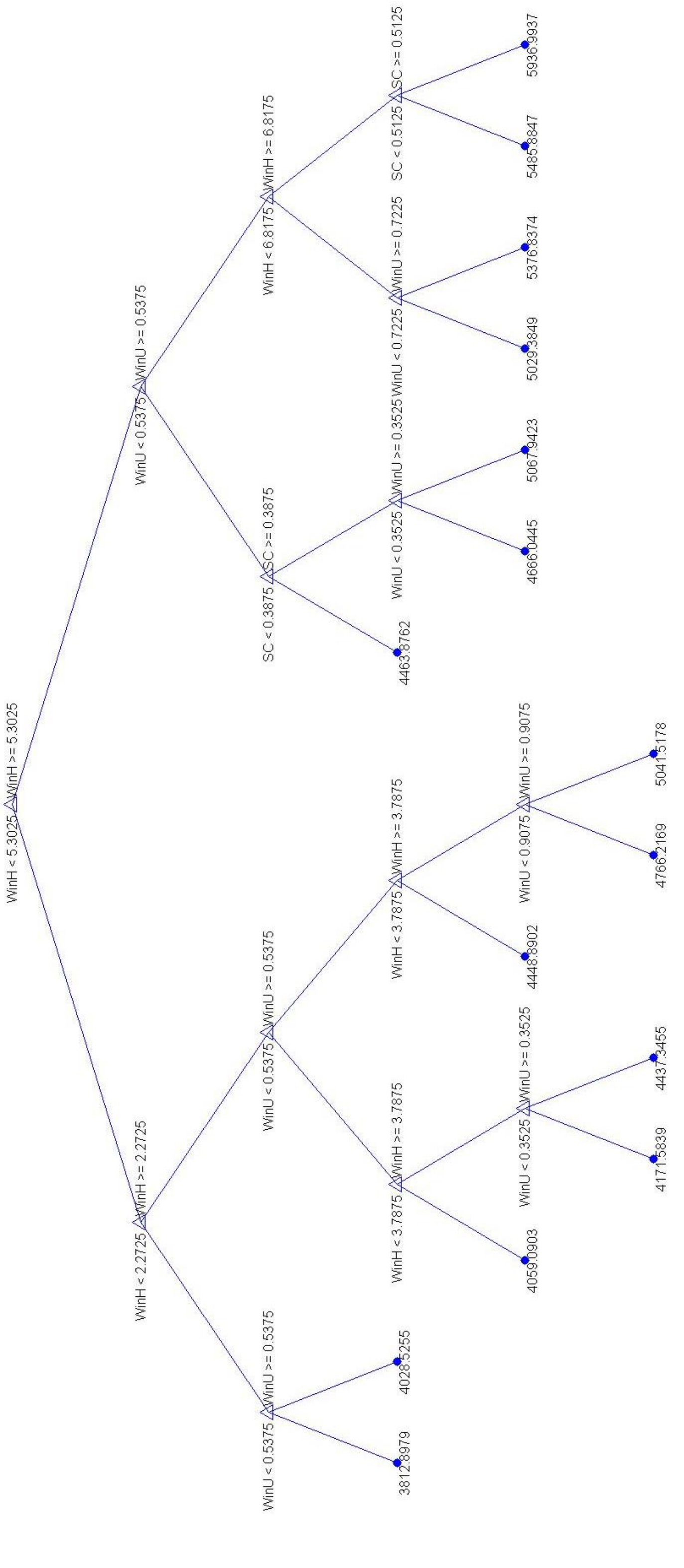


Figure 33 : PED Prediction Tree_3 Variables (Test CV 4.41%)

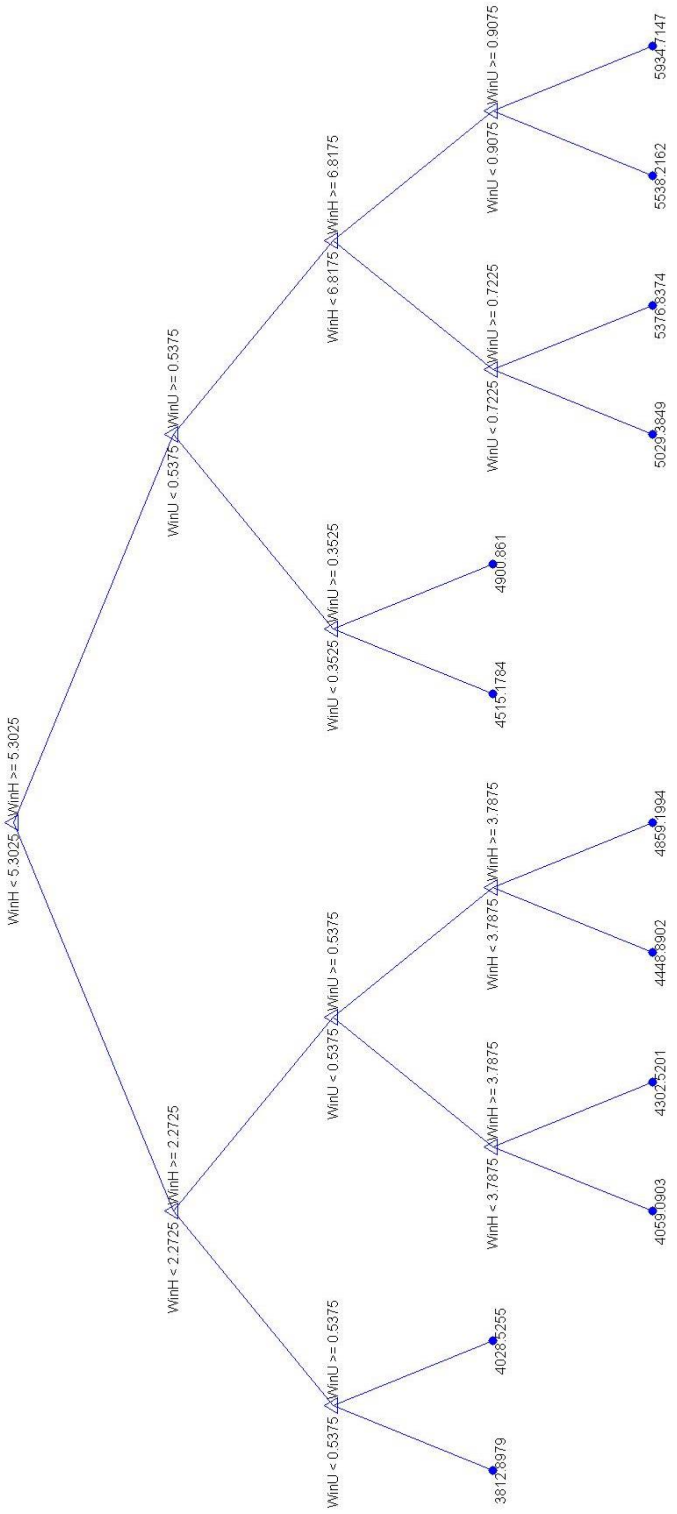


Figure 34 : PED Prediction Tree_2 Variables (Test CV 5%)

6.1.6 Final Model Selection

The results obtained from analyzing the small feature set suggest that if high prediction accuracy and explicit analytical models are required, then OLS regression is the better choice. However, if visual interpretability is more desirable and a reasonable reduction in predictive ability can be tolerated (as is often the case with early stage or schematic design synthesis) then CART may be a better alternative. CART is flexible enough to handle categorical independent variables, categorical dependent variables (classification problems) and is also well suited to handle a non-continuous solution space.

CCD based run generation was found to be very promising due to the excellent predictive ability of the OLS models built on the CCD dataset, which contains less < 2% of the variable combinations in the exhaustive dataset. This translates into a very significant reduction in simulation run time. Hence for OLS regression modeling, CCD or other such experimental design techniques should be explored wherever possible instead of exhaustive enumeration. However, CART being a statistical induction technique which fits local models to variations in the data, the CCD dataset was too sparse to train a useful decision tree and the entire exhaustive data was used instead.

While it was feasible to simulate the exhaustive combinations of the small feature set, such an approach is impractical for larger data sets where combinations might run into tens of thousands or even millions. Experimental design techniques such as Latin Hypercube Sampling would be required to generate a feasible number of representative simulation runs.

6.2 Large Design Feature Set

The large feature data set is a matrix of dimension 15000 x 15 (Simulation Runs x Predictors). The 15000 rows were obtained using Latin Hypercube Sampling from a sample space of over 3^{15} possible combinations. See section 5.3.2 for an overview of the analysis performed on the large feature set.

6.2.1 Simulated Building Energy Responses

The simulated responses are EUI in kBtu/sqft/yr. and PED in kW/yr. Figure 35 shows the frequency distribution of the simulated EUI across all 15000 runs. The Y –Axis represents the number of runs for each EUI bin. The distribution has a positive skew with a mean of 54 kBtu/sqft/yr.

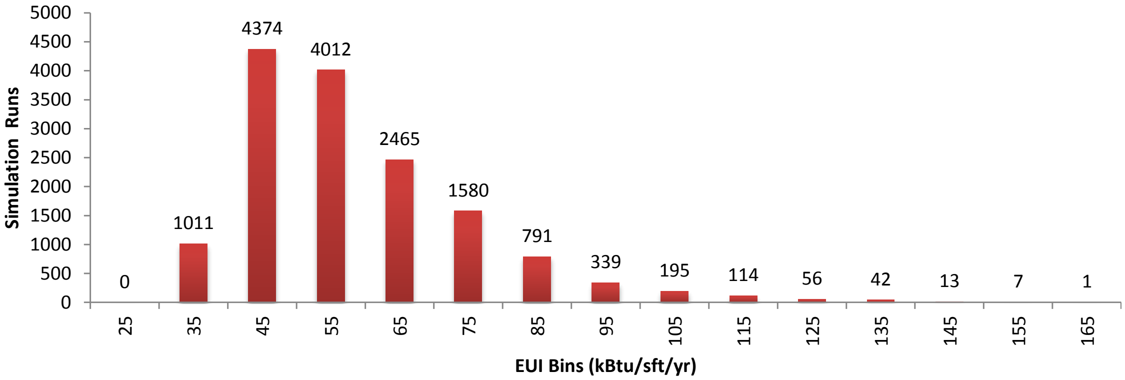


Figure 35 : Frequency Histogram of Simulated EUI

	EUI	PED
	kBtu/sft/yr	kW/yr
Max	161	17652
Median	50	5331
Mean	54	5739
Min	25	2393

Table 9: Simulated Response Descriptive Statistics

Figure 36 shows the frequency distribution of the simulated PED across all 15000 runs. The Y –Axis represents the number of runs for each PED bin. The distribution has a positive skew with a mean of 5739 kW/yr. Figure 37 clearly depicts a strong linear correlation between the two responses for electric demand and consumption.

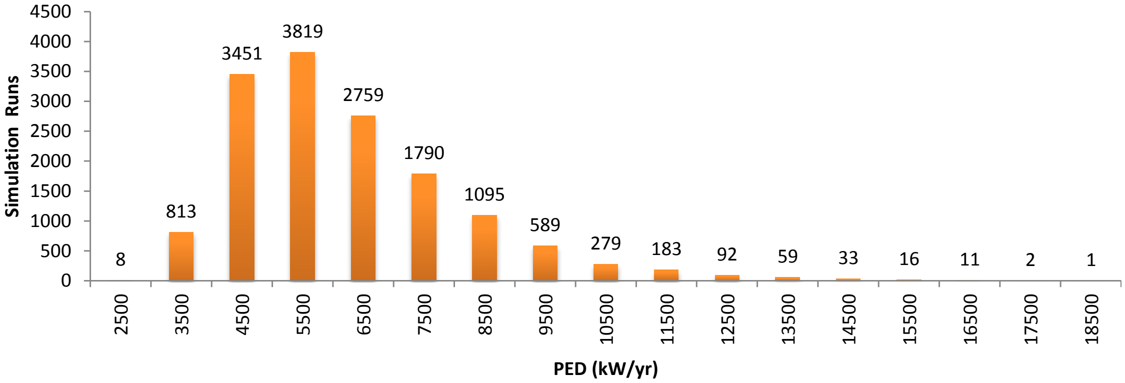


Figure 36 : Frequency Histogram of Simulated PED

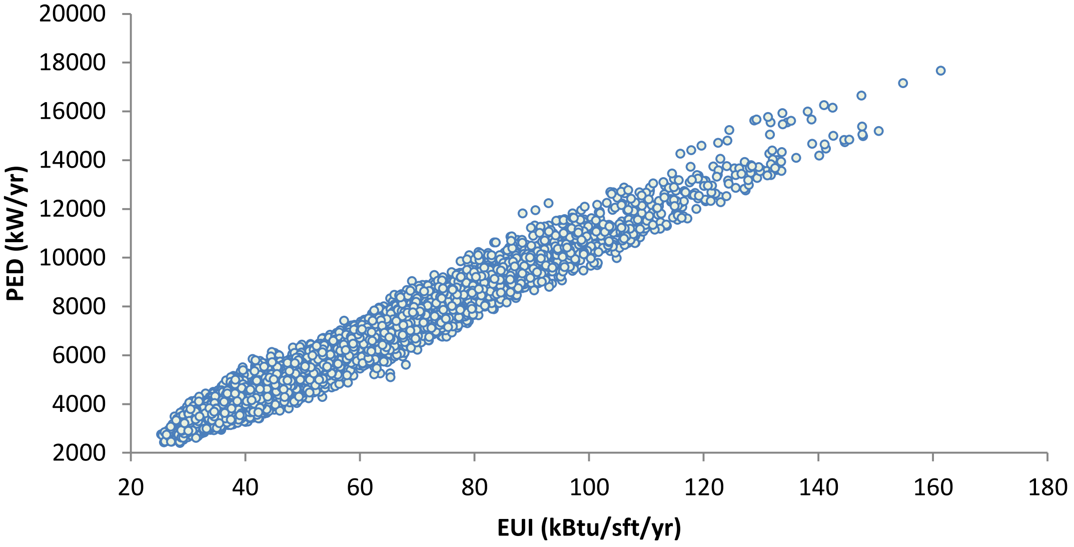


Figure 37 : Simulated Response Scatter Plot (15 Vars)

6.2.2 Feature Selection results using Random Forest

A random forest ensemble of 500 regression trees was built on the large feature dataset and a CV of 6.52% was found for EUI prediction. Despite the fact that the RF ensemble is also a very capable predictive model, the purpose of using RF at this stage was solely to take advantage of its robust feature selection ability. See section 4.6.2 for a detailed overview of the feature selection process using RF. The strength of this technique lies in the fact that it ranks variables based on their predictive ability using OOB (Out-of-Bag) test data. An overall measure of predictor importance is standardized for each variable across all the trees in the ensemble and is provided as a Z-score. A Z-score > 3 may be used to select the most important predictors; however, a more appropriate representation would be to scale the variable importance measures relative to the maximum observed value as shown in Figure 38. In this case the Var Imp measure of Min-FlowR is unity, so the second most influential variable Min-CoolT has a relative measure of approximately 0.8 and so on.

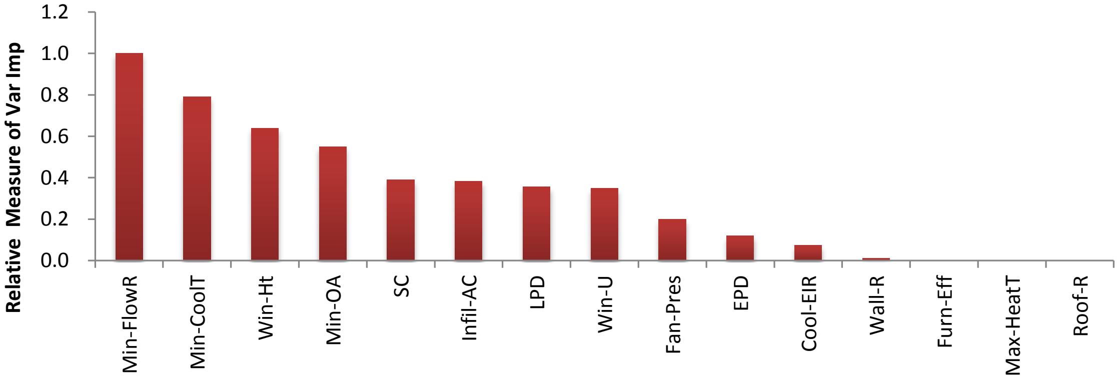


Figure 38 : Relative Measure of Variable Importance (EUI)

Another RF ensemble of 500 regression trees was trained for PED prediction, and a test CV of 6.13% was found. Figure 39 depicts relative importance of the PED predictors by scaling the individual Z-scores to the value for Min-FlowR.

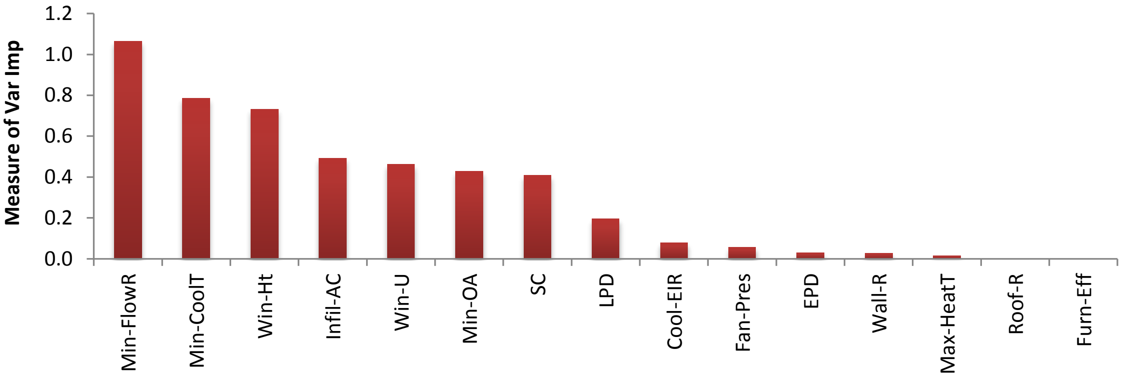


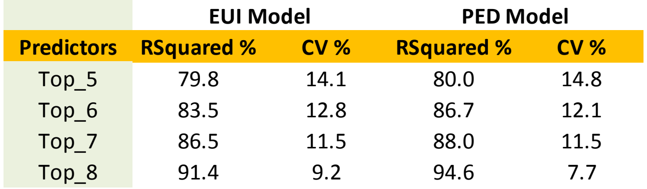
Figure 39 : Relative Measure of Variable Importance (PED)

We find that for both responses the top three most important predictors (Min-FlowR, Min-CoolT and Win-Ht.) are the same. The rest are fairly close in terms of their ranking and there is no instance of any variable with a wide disparity in ranking between the two responses. This may, however, not be the case if other response criteria are selected. While the physical significance of the top ranked predictors may be ascertained in some cases, complex underlying interactions and relationships may make it difficult to directly associate variable ranking with the actual (or expected) physical influence of the variable. There may also be some ambiguity for adjacently ranked variables with very close measures of importance. Hence the interpretation of variable importance will often also require the domain insight of the designer.

6.2.3 OLS Predictive Models

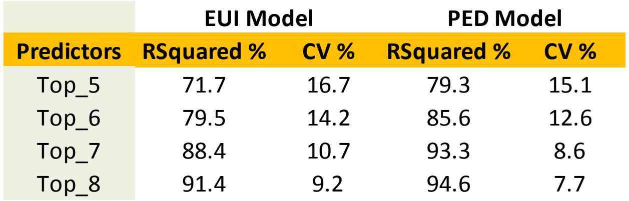
Once the important predictors were identified , pairs of OLS models (one for each response) were fit to successively larger subsets (starting with 5) of the top ranked variables for EUI (Figure 38) and PED prediction (Figure 39), till established model performance criteria were achieved (CV < 10 % and $R^2 > 90\%$).

Table 10 and Table 11 present pertinent statistics for the OLS models built on the top ranked variables related to EUI and PED respectively. Only the models involving the Top-8 variables were found to meet the pre-defined criteria and were selected for the next phase. Incidentally, the Top 8 predictors for both responses are the same. Table 12 lists the coefficients and terms of the two selected OLS regression models.



	EUI Model		PED Model	
Predictors	RSquared %	CV %	RSquared %	CV %
Top_5	79.8	14.1	80.0	14.8
Top_6	83.5	12.8	86.7	12.1
Top_7	86.5	11.5	88.0	11.5
Top_8	91.4	9.2	94.6	7.7

Table 10 : OLS Models fit to Top Ranked Variables related EUI



	EUI Model		PED Model	
Predictors	RSquared %	CV %	RSquared %	CV %
Top_5	71.7	16.7	79.3	15.1
Top_6	79.5	14.2	85.6	12.6
Top_7	88.4	10.7	93.3	8.6
Top_8	91.4	9.2	94.6	7.7

Table 11 : OLS Models fit to Top Ranked Variables related to PED

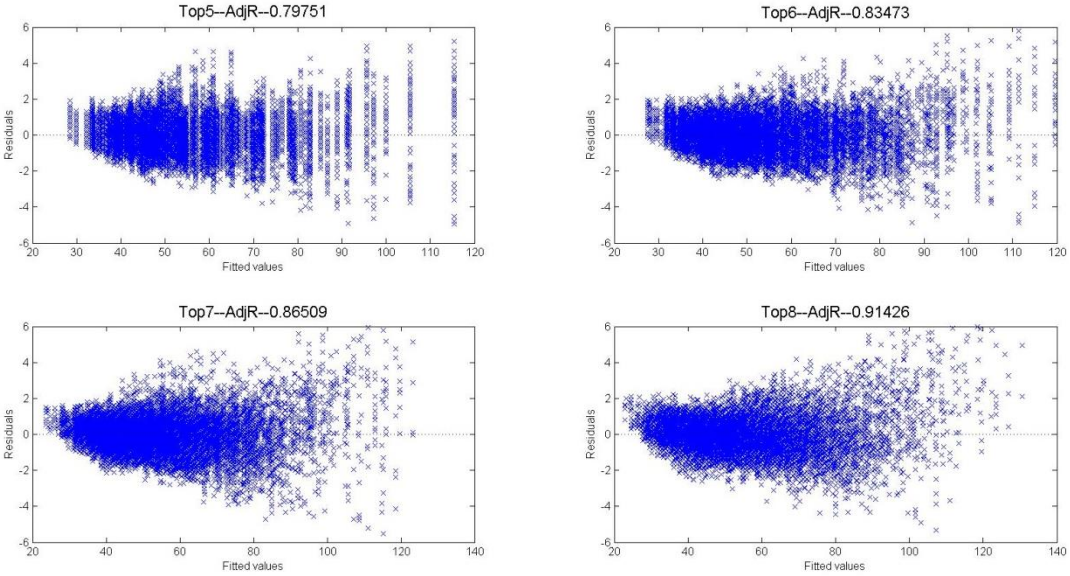


Figure 40 : Residual Plots of EUI prediction models built on Top (5-8) Variables

The standardized residual plots in Figure 40 and Figure 41 suggest some degree of hetroskedasticity at higher values of EUI and PED. However it may be reasoned that prediction accuracy at such high values is not critical. It is also clear that there is no noticeable improvement in residual behavior with the increase in variables from 5 to 8.

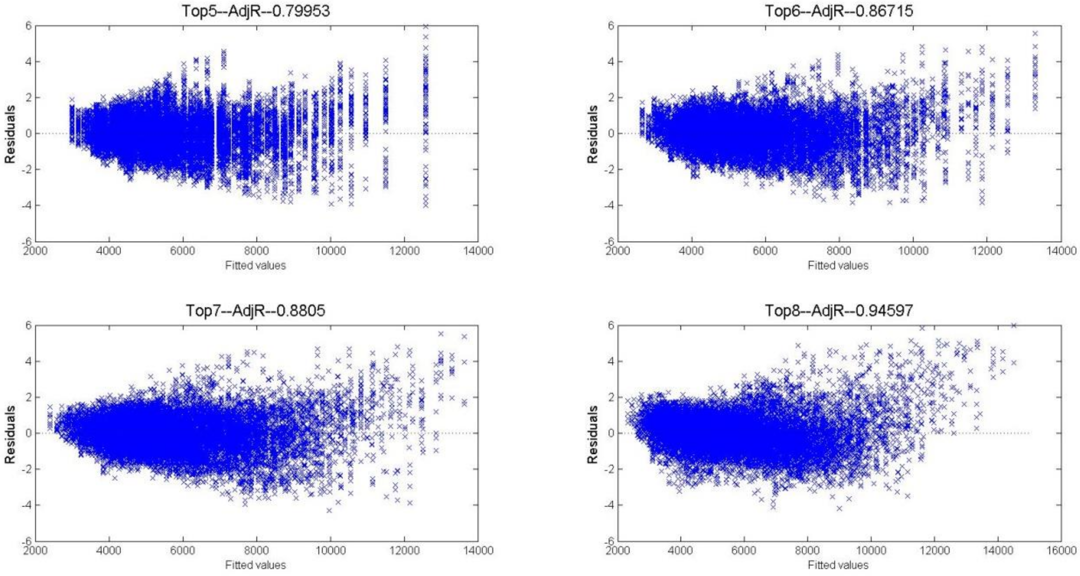


Figure 41 : Residual Plots of PED prediction models built on Top (5-8) Variables

EUI Model	
Terms	**Coefficients**
Intercept	471.87
Min-FlowR	-128.24
Min-CoolT	-13.70
Win-Ht	-8.67
Min-OA	-61.91
SC	-92.83
Infil-AC	16.21
LPD	0.25
Win-U	3.60
Min-FlowR x Min-CoolT	1.61
Min-FlowR x Win-Ht	2.70
Min-FlowR x SC	13.13
Min-FlowR x Infil-AC	13.59
Min-FlowR x LPD	-0.72
Min-FlowR x Win-U	8.56
Min-CoolT x Win-Ht	0.11
Min-CoolT x Min-OA	1.02
Min-CoolT x SC	1.27
Min-CoolT x Infil-AC	-0.31
Min-CoolT x LPD	0.09
Min-CoolT x Win-U	-0.07
Win-Ht x Min-OA	2.79
Win-Ht x SC	3.00
Win-Ht x Infil-AC	-0.15
Win-Ht x LPD	-0.09
Win-Ht x Win-U	1.62
Min-OA x SC	17.50
Min-OA x Infil-AC	6.69
Min-OA x Win-U	5.16
SC x Infil-AC	-3.65
SC x Win-U	-2.96
Infil-AC x LPD	-0.54
Infil-AC x Win-U	-0.86
Min-FlowR x Min-FlowR	27.16
Min-CoolT x Min-CoolT	0.11
SC x SC	6.42
Infil-AC x Infil-AC	3.50
LPD x LPD	0.57
Win-U x Win-U	-2.54

PED Model	
Terms	**Coefficients**
(Intercept)	48148.73
Min-FlowR	-12101.34
Min-CoolT	-1406.84
Win-Ht	-935.71
Min-OA	-7270.63
SC	-9110.29
Infil-AC	1863.84
LPD	-464.45
Win-U	393.77
Min-FlowR x Min-CoolT	149.75
Min-FlowR x Win-Ht	302.07
Min-FlowR x Min-OA	2875.15
Min-FlowR x SC	1572.91
Min-FlowR x Infil-AC	1365.61
Min-FlowR x LPD	-80.91
Min-FlowR x Win-U	722.68
Min-CoolT x Win-Ht	11.64
Min-CoolT x Min-OA	112.81
Min-CoolT x SC	117.53
Min-CoolT x Infil-AC	-24.87
Min-CoolT x LPD	11.54
Win-Ht x Min-OA	219.63
Win-Ht x SC	328.72
Win-Ht x Infil-AC	-22.35
Win-Ht x Win-U	200.74
Min-OA x SC	1784.60
Min-OA x Infil-AC	1024.31
Min-OA x LPD	134.88
Min-OA x Win-U	128.21
SC x Infil-AC	-149.10
SC x Win-U	-207.69
Infil-AC x Win-U	-258.56
LPD x Win-U	50.25
Min-FlowR x Min-FlowR	2274.78
Min-CoolT x Min-CoolT	10.95
Min-OA x Min-OA	-2274.27
SC x SC	629.97
Infil-AC x Infil-AC	177.38
LPD x LPD	77.06
Win-U x Win-U	-351.63

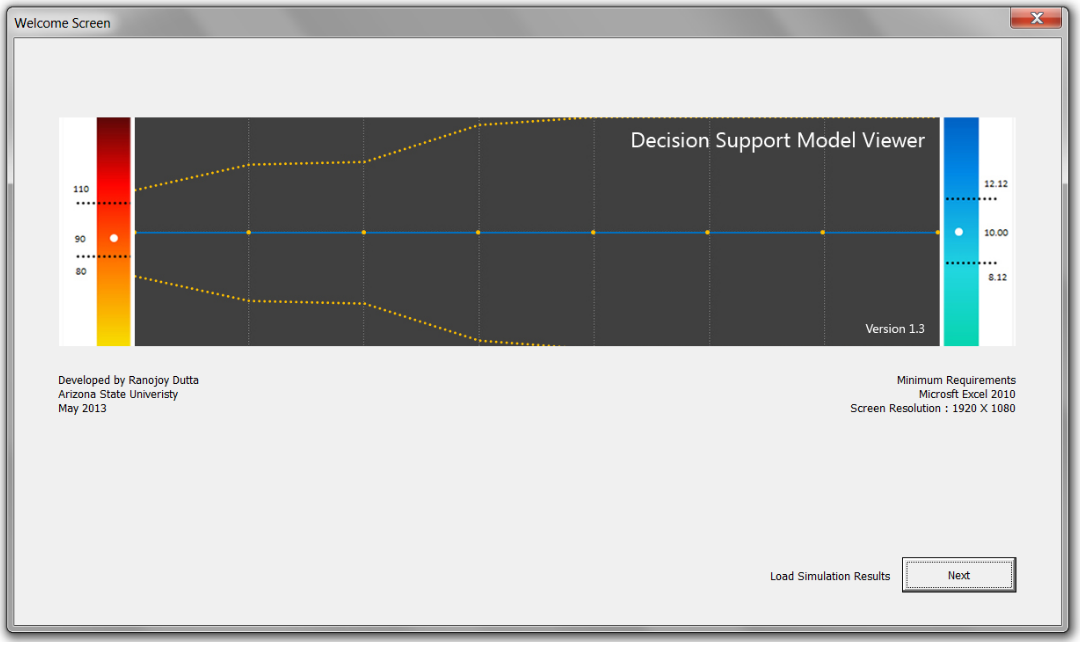
Table 12 : OLS Model Coefficients

6.2.4 Selecting Final Solution(s) using the DSMV Interface

Once the regression models have been selected the user can make real time predictions using the Decision Support Model Viewer (DSMV) Interface version 1.3. The DSMV has been developed to facilitate the interactive visualization stage of the VADSM framework proposed by this research. It has been programmed in VBA (Visual Basic for Applications) using Microsoft® Excel as the host application. The DSMV is independent of any particular energy simulation program and only requires the following inputs in .xlsx or .csv format

- Independent Variable Names , Units and Levels (Only Numeric Variables)
- Dependent Variables Names and Units
- Table of Simulation Inputs [X](Rows represent runs and Columns represent variables)
- Simulated Response(s) for each simulation input row in [X]

The DSMV has been designed to gradually present more complex data as the user proceeds though a wizard consisting of several user forms. The following section presents DSMV screen captures and provides descriptions of associated information/options using an illustrative example.



Decision Support Model Viewer

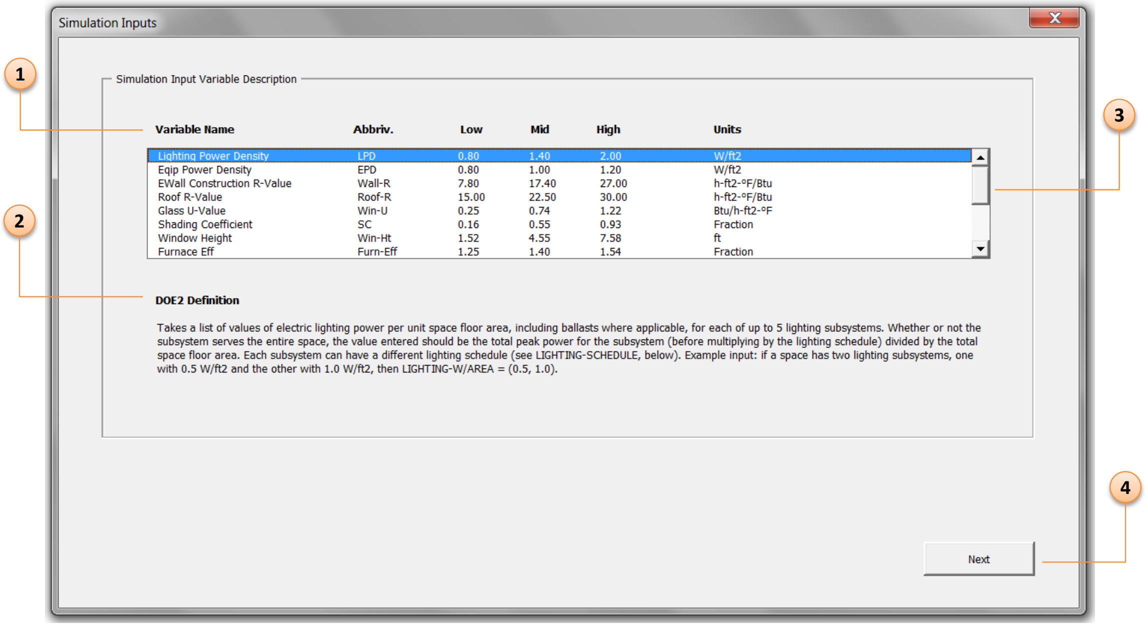
Version 1.3

Developed by Ranojoy Dutta
Arizona State Univeristy
May 2013

Minimum Requirements
Microsft Excel 2010
Screen Resolution : 1920 X 1080

Figure 42 : DSMV Welcome Screen

Upon loading the DSMV wizard a user is presented with the **welcome screen** that offers a single button to load simulation results and load the Simulation Inputs screen.



Variable Name	Abbriv.	Low	Mid	High	Units
Lighting Power Density	LPD	0.80	1.40	2.00	W/ft2
Eqip Power Density	EPD	0.80	1.00	1.20	W/ft2
EWall Construction R-Value	Wall-R	7.80	17.40	27.00	h-ft2-°F/Btu
Roof R-Value	Roof-R	15.00	22.50	30.00	h-ft2-°F/Btu
Glass U-Value	Win-U	0.25	0.74	1.22	Btu/h-ft2-°F
Shading Coefficient	SC	0.16	0.55	0.93	Fraction
Window Height	Win-Ht	1.52	4.55	7.58	ft
Furnace Eff	Furn-Eff	1.25	1.40	1.54	Fraction

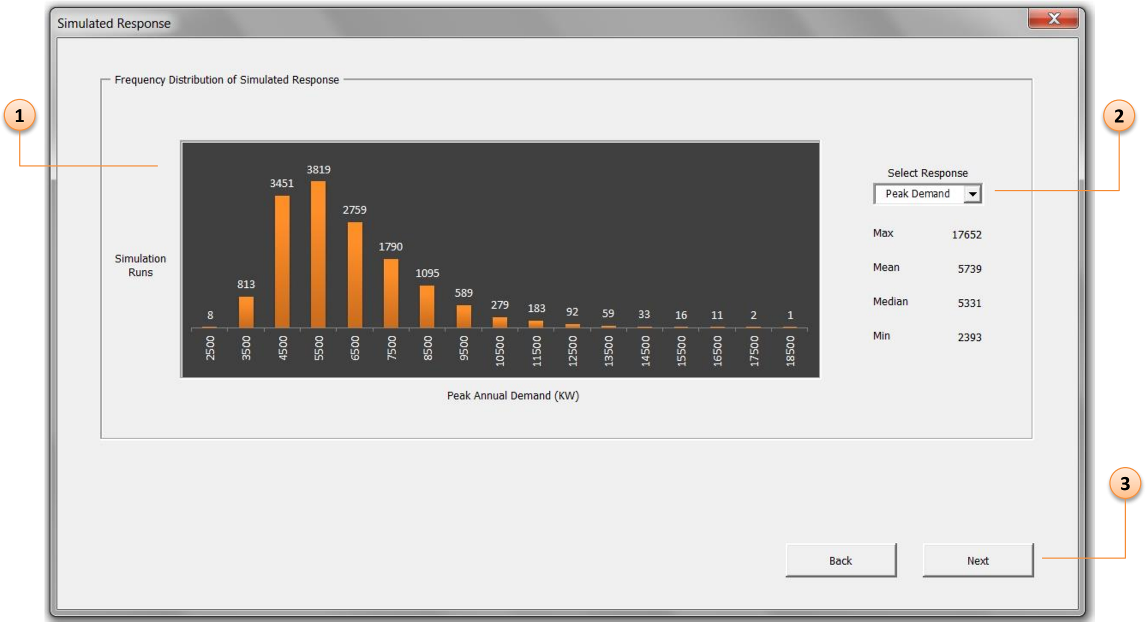
DOE2 Definition

Takes a list of values of electric lighting power per unit space floor area, including ballasts where applicable, for each of up to 5 lighting subsystems. Whether or not the subsystem serves the entire space, the value entered should be the total peak power for the subsystem (before multiplying by the lighting schedule) divided by the total space floor area. Each subsystem can have a different lighting schedule (see LIGHTING-SCHEDULE, below). Example input: if a space has two lighting subsystems, one with 0.5 W/ft2 and the other with 1.0 W/ft2, then LIGHTING-W/AREA = (0.5, 1.0).

Figure 43 : DSMV Simulation Inputs

The **Simulation Inputs** screen provides the following information and options.

1. Table of Input Variable names, abbreviations, levels and units. In this case the display shows the three levels for each design variable.
2. DOE2 definitions of each variable (can be replaced by user descriptions or notes).
3. Scroll Bar to view additional variables.
4. Load the next screen.



Select Response: Peak Demand

Max 17652

Mean 5739

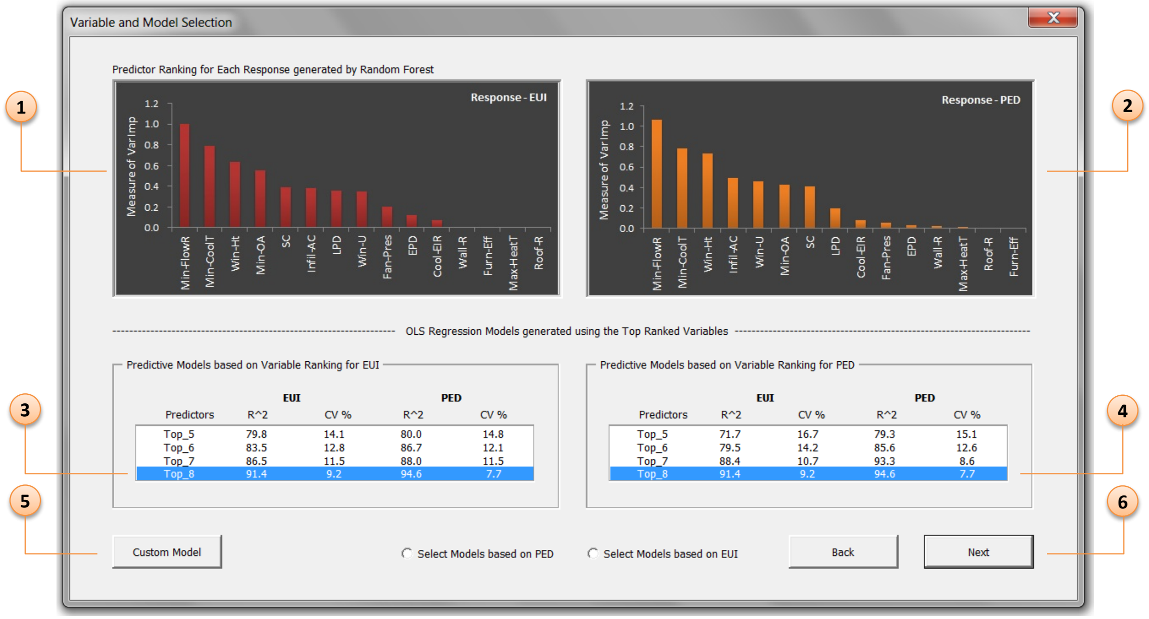
Median 5331

Min 2393

Figure 44 : DSMV Simulated Response Distribution

The **Simulated Response** screen provides the following information and options

1. A frequency distribution of the simulated response. In this graph Peak Electric Demand in kW is depicted. The X-Axis represents kW bins and the Y-axis shows the number of simulated runs. This information will provide the user with guidelines for setting practical constraints on the response.
2. This dropdown allows the user to view additional response variable(s) and their distribution.
3. Load the next screen.



Predictor Ranking for Each Response generated by Random Forest

OLS Regression Models generated using the Top Ranked Variables

Predictive Models based on Variable Ranking for EUI

	EUI		PED	
Predictors	R^2	CV %	R^2	CV %
Top_5	79.8	14.1	80.0	14.8
Top_6	83.5	12.8	86.7	12.1
Top_7	86.5	11.5	88.0	11.5
Top_8	91.4	9.2	94.6	7.7

Predictive Models based on Variable Ranking for PED

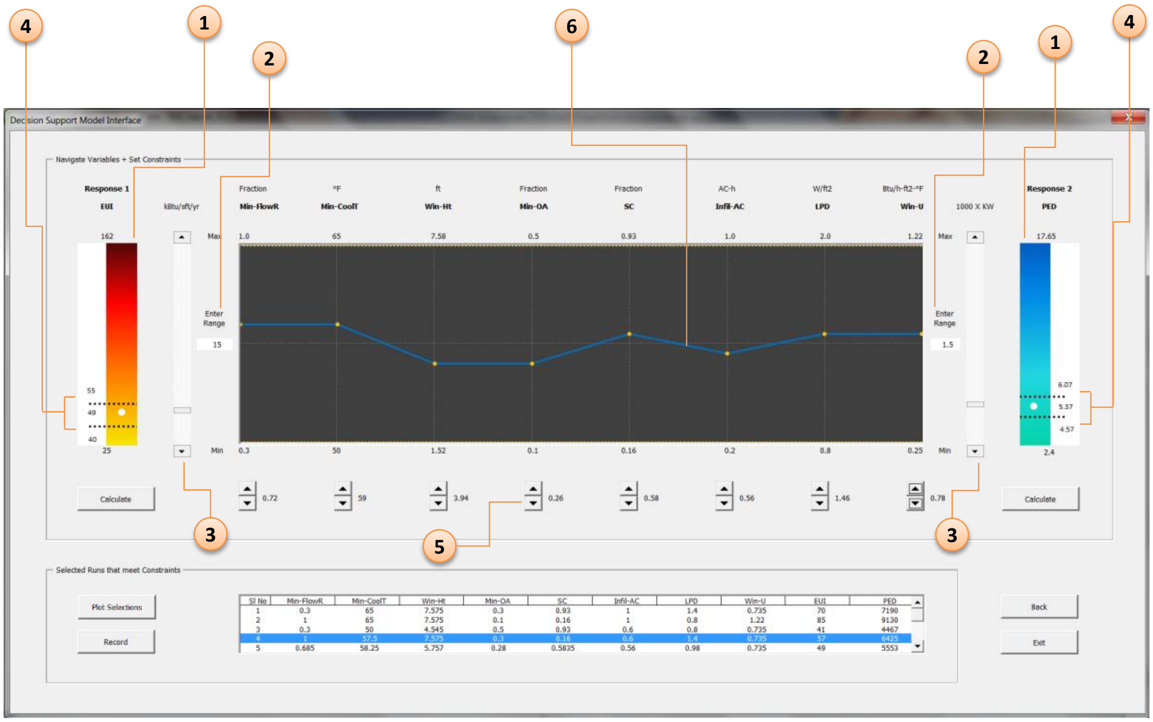
	EUI		PED	
Predictors	R^2	CV %	R^2	CV %
Top_5	71.7	16.7	79.3	15.1
Top_6	79.5	14.2	85.6	12.6
Top_7	88.4	10.7	93.3	8.6
Top_8	91.4	9.2	94.6	7.7

Figure 45 : DSMV Variable and Model Selection

The **Variable and Model Selection** screen provides the following information and options

1. Variable Importance Ranking determined by the Random Forest algorithm for Response 1 (EUI). The variable scores are sorted along the X-Axis in decreasing importance from left to right.
2. Variable Importance Ranking for Response 2 (PED).
3. Pairs of Regression models (one for each response) built on top ranked 5, 6, 7 and 8 variables as per Response 1. R^2 and CV are provided as selection parameters.
4. Pairs of Regression models built on top 5, 6, 7 and 8 variables as per Response 2.

   Note: In this example the models built on the Top-8 variables were selected based on R^2 and CV. See Table 12 for the complete list of regression coefficients.
5. Alternatively, a user may build models with a custom variable combination.
6. Load the next screen.



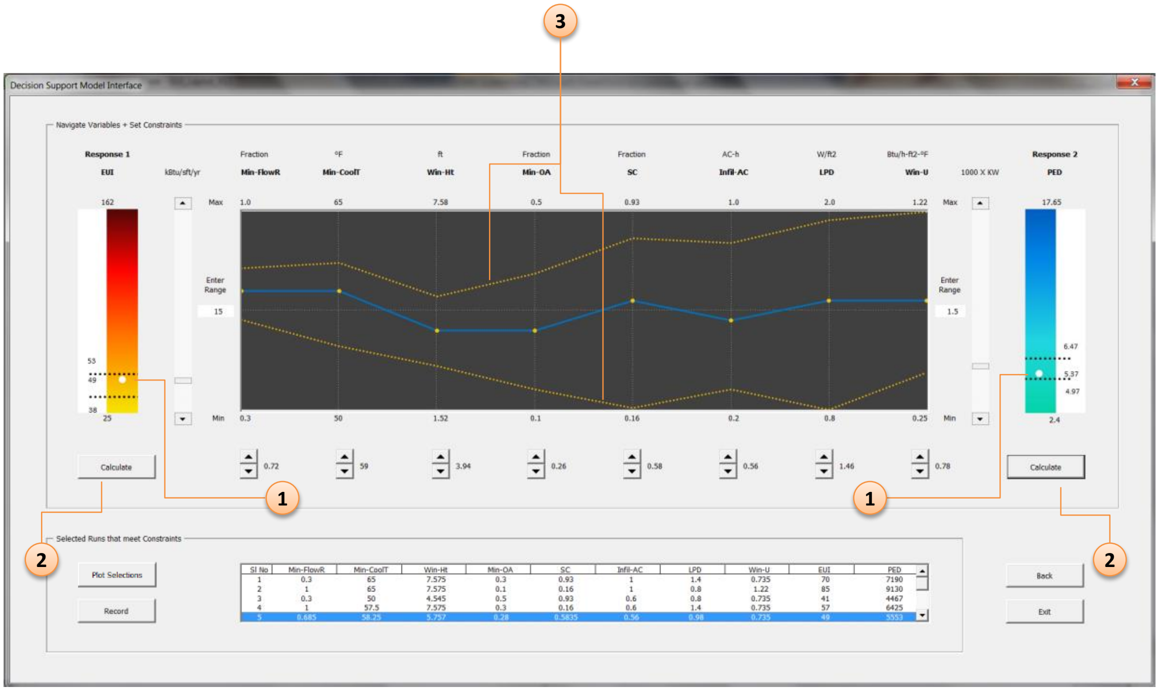
Sl No	Min-FlowR	Min-CoolT	Win-Ht	Min-OA	SC	Infil-AC	LPD	Win-U	EUI	PED
1	0.3	65	7.575	0.3	0.93	1	1.4	0.735	70	7190
2	1	65	7.575	0.1	0.16	1	0.8	1.22	85	9130
3	0.3	50	4.545	0.5	0.93	0.6	0.8	0.735	41	4467
4	1	57.5	7.575	0.3	0.16	0.6	1.4	0.735	57	6425
5	0.685	58.25	5.757	0.28	0.5835	0.56	0.98	0.735	49	5553

Figure 46 : DSMV Decision Support Model Interface Part 1

The **Decision Support Model Interface** screen provides the following information

1. Graphical representation of simulated response range.
2. Enter user defined width of constraint region for the Response.
3. Scroll bar to position the constraint region within the simulated range.
4. User defined Response constraint bands.
5. Spin button controls to adjust variable values between provided limits.
6. Polyline representing a single user selected variable combination (design option) as input to the regression models.

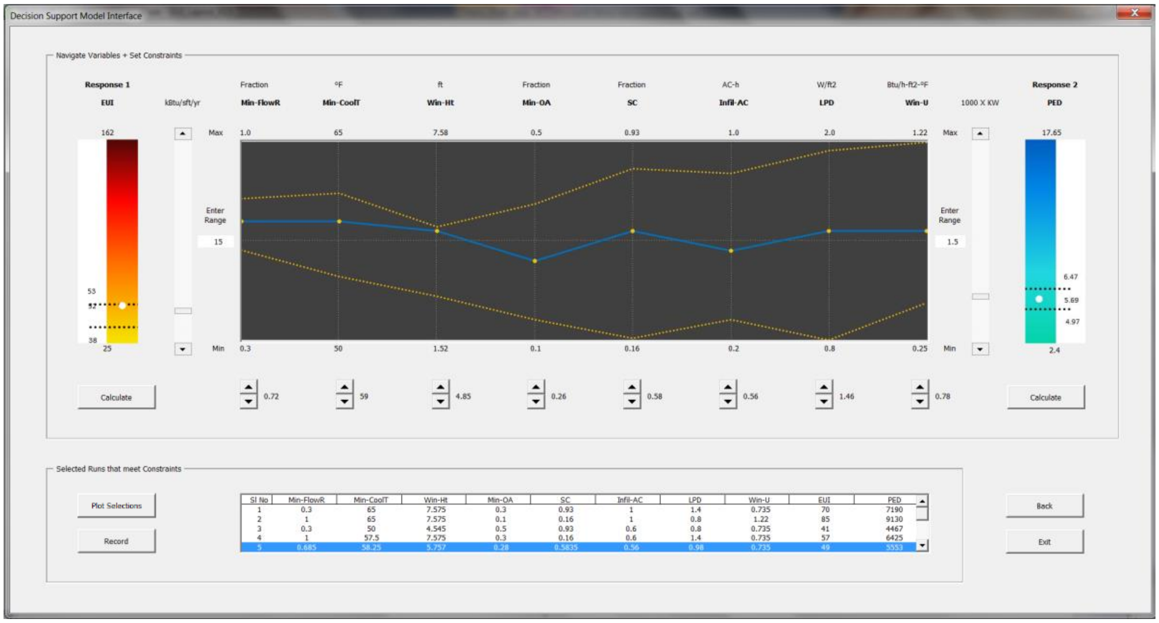
Note: Only the Top 8 variables have been selected for visualization and are presented in order of importance from left to right



Sl No	Min-FlowR	Min-CoolT	Win-Ht	Min-OA	SC	Infil-AC	LPD	Win-U	EUI	PED
1	0.3	65	7.575	0.3	0.93	1	1.4	0.735	70	7190
2	1	65	7.575	0.1	0.16	1	0.8	1.22	85	9130
3	0.3	50	4.545	0.5	0.93	0.6	0.8	0.735	41	4467
4	1	57.5	7.575	0.3	0.16	0.6	1.4	0.735	57	6425
5	0.685	58.25	5.757	0.28	0.5835	0.56	0.98	0.735	49	5553

Figure 47 : DSMV Decision Support Model Interface Part 2

1. The white dot represents the predicted response(s) based on the user selected variable combination (blue polyline). **Note**: This dot has to remain within the constraint bands in order to satisfy the response criteria.
2. Once the response criteria have been fixed this button calculates Min-Max Ranges for each variable by successively feeding the response constraints and the selected values of all the other 7 predictors as constants, into the second order regression equation and solving a resulting quadratic equation for each variable. These Min-Max points are connected by the yellow dotted lines (Label 3) as a visual range of movement for each variable. **Note:** Solving the quadratic equations may result in more than one solution within the predefined variable range; however, in this interface only the first one to satisfy the constraints is selected by default.

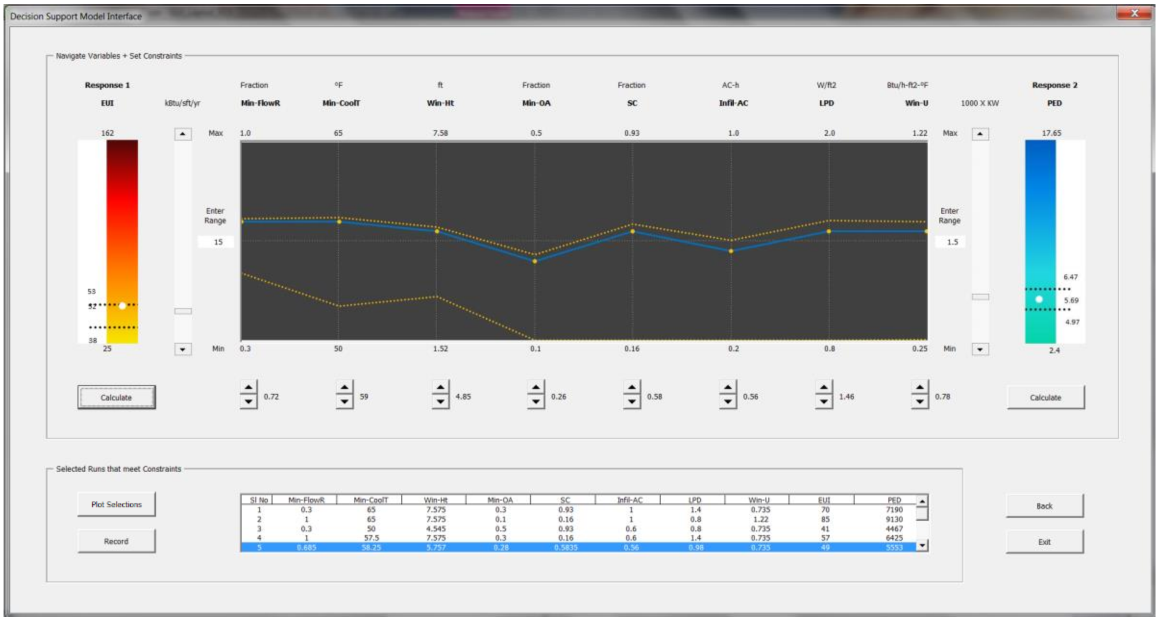


Sl No	Min-FlowR	Min-CoolT	Win-Ht	Min-OA	SC	Infil-AC	LPD	Win-U	EUI	PED
1	0.3	65	7.575	0.3	0.93	1	1.4	0.735	70	7190
2	1	65	7.575	0.1	0.16	1	0.8	1.22	85	9130
3	0.3	50	4.545	0.5	0.93	0.6	0.8	0.735	41	4467
4	1	57.5	7.575	0.3	0.16	0.6	1.4	0.735	57	6425
5	0.685	58.25	5.757	0.28	0.5835	0.56	0.98	0.735	49	5553

Figure 48 : DSMV Decision Support Model Interface Part 3

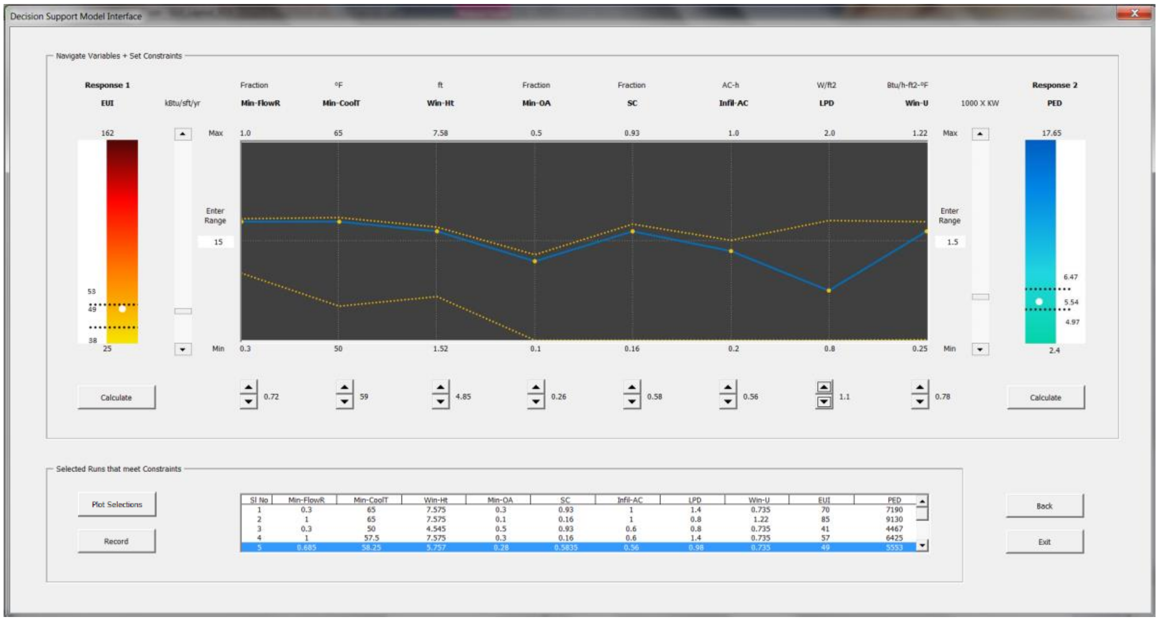
Assume that the requirement is to increase Win-Ht. from 3.945ft and Min-OA fraction from 0.26 (Figure 47) to 4.85ft and 0.32 respectively, while meeting the response criteria of (38< EUI<53 kBtu/sqft) and (4.97<PED<6.47 MW). When Win-Ht. is increased to 4.85 ft. (Figure 48) predicted EUI hits the upper constraint of approx. 53 kBtu/sqft./yr. (white dot on the red bar) so we re-calculate the available variable ranges for the given constraints. The new upper bounds suggest that none of the variables can be further increased (Figure 49) for the given response criteria.

At this point the only alternative is to adjust another variable in order to lower the predicted EUI. In this particular example LPD is reduced from 1.46 to 1.1 W/ft2 (Figure 50) since an increase in Win-Ht. can be expected to provide more natural lighting.



Sl No	Min-FlowR	Min-CoolT	Win-Ht	Min-OA	SC	Infil-AC	LPD	Win-U	EUI	PED
1	0.3	65	7.575	0.3	0.93	1	1.4	0.735	70	7190
2	1	65	7.575	0.1	0.16	1	0.8	1.22	85	9130
3	0.3	50	4.545	0.5	0.93	0.6	0.8	0.735	41	4467
4	1	57.5	7.575	0.3	0.16	0.6	1.4	0.735	57	6425
5	0.685	58.25	5.757	0.28	0.5835	0.56	0.98	0.735	49	5553

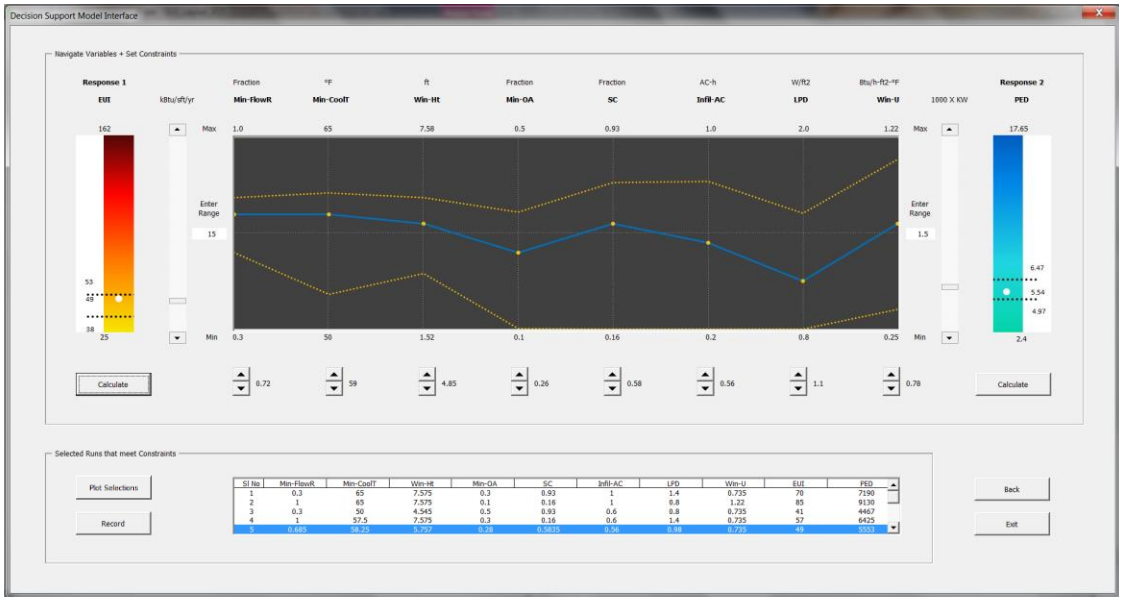
Figure 49 : DSMV Decision Support Model Interface Part 4



Sl No	Min-FlowR	Min-CoolT	Win-Ht	Min-OA	SC	Infil-AC	LPD	Win-U	EUI	PED
1	0.3	65	7.575	0.3	0.93	1	1.4	0.735	70	7190
2	1	65	7.575	0.1	0.16	1	0.8	1.22	85	9130
3	0.3	50	4.545	0.5	0.93	0.6	0.8	0.735	41	4467
4	1	57.5	7.575	0.3	0.16	0.6	1.4	0.735	57	6425
5	0.685	58.25	5.757	0.28	0.5835	0.56	0.98	0.735	49	5553

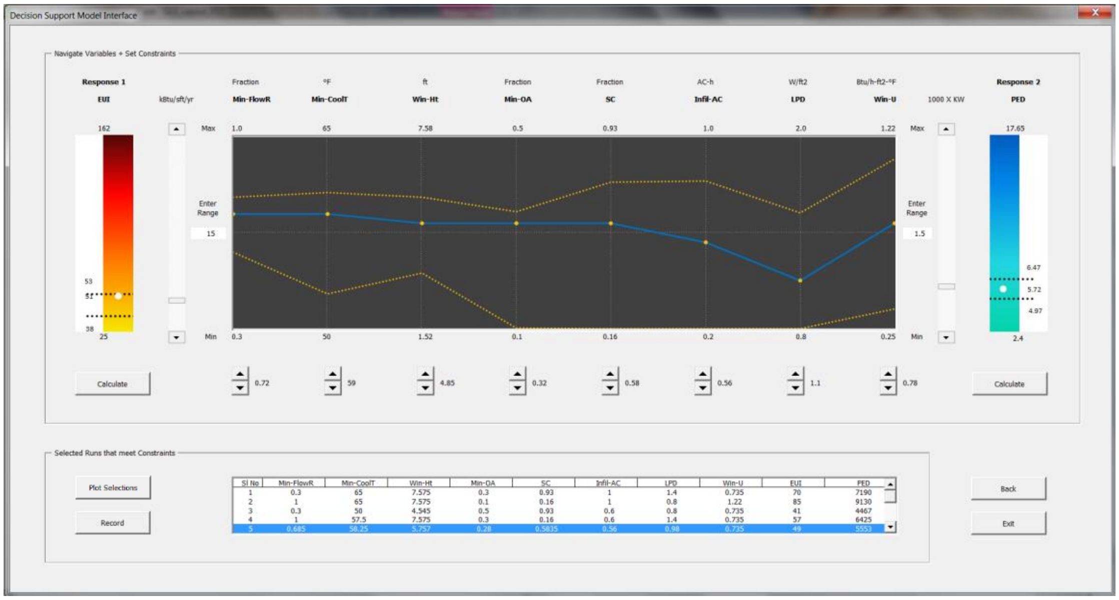
Figure 50 : DSMV Decision Support Model Interface Part 5

After a reduction in LPD the EUI drops to 49 kBtu/sqft./yr. The variable ranges are recalculated and now there is more flexibility in adjusting the other variables (Figure 51) while still meeting the response criteria. The Min-OA fraction is now increased to 0.32 and the final design solution still meets the response criteria (Figure 52).



Sl No	Min-FlowR	Min-CoolT	Win-Ht	Min-OA	SC	Infil-AC	LPD	Win-U	EUI	PED
1	0.3	65	7.575	0.3	0.93	1	1.4	0.735	70	7190
2	1	65	7.575	0.1	0.16	1	0.8	1.22	85	9130
3	0.3	50	4.545	0.5	0.93	0.6	0.8	0.735	41	4467
4	1	57.5	7.575	0.3	0.16	0.6	1.4	0.735	57	6425
5	0.685	58.25	5.757	0.28	0.5835	0.56	0.98	0.735	49	5553

Figure 51 : DSMV Decision Support Model Interface Part 6



Sl No	Min-FlowR	Min-CoolT	Win-Ht	Min-OA	SC	Infil-AC	LPD	Win-U	EUI	PED
1	0.3	65	7.575	0.3	0.93	1	1.4	0.735	70	7190
2	1	65	7.575	0.1	0.16	1	0.8	1.22	85	9130
3	0.3	50	4.545	0.5	0.93	0.6	0.8	0.735	41	4467
4	1	57.5	7.575	0.3	0.16	0.6	1.4	0.735	57	6425
5	0.685	58.25	5.757	0.28	0.5835	0.56	0.98	0.735	49	5553

Figure 52 : DSMV Decision Support Model Interface Part 7

This solution can now be recorded, and if required, the entire process can repeated for evaluating other feasible design options. After the desired number of iterations the final solutions (red numbered circles) can be compared as shown in Figure 53 by plotting them using the two design criteria (EUI & PED) as X and Y coordinates respectively . In this example, 6 alternative design options were generated in the manner described above and plotted within the DSMV interface using the - Plot Selections – button.

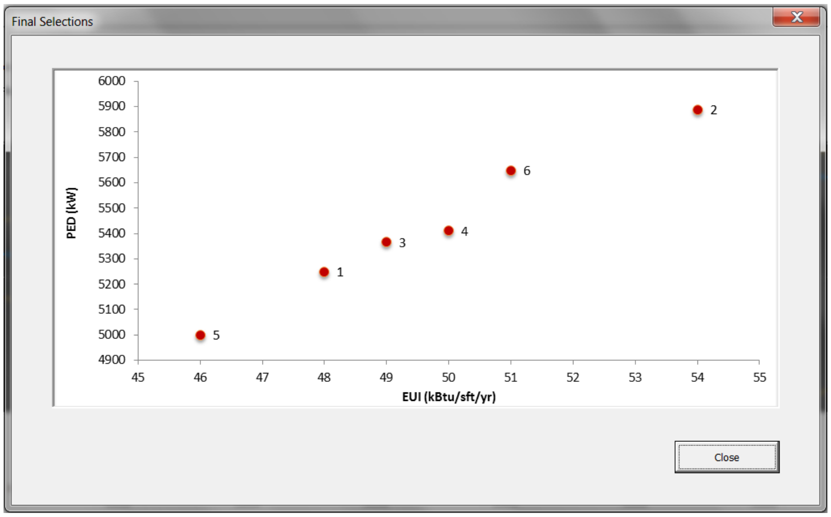


Figure 53 : Final Solution Comparison Chart

In this case the two criteria (EUI & PED) happen to be strongly correlated (Figure 37) hence the linear trend in Figure 53 . Consequently in this case there is a single solution that has both lowest EUI and PED values. Such a solution however, may not be attainable if inversely related design criteria are being considered. The final selections plot in that case would be useful in identifying more than one non-dominated solution; allowing the designer to evaluate tradeoffs offered by each.

7: SUMMARY AND FUTURE WORK

7.1 Summary

High performance (low energy) building design is a difficult multi-criteria decision making (MCDM) problem that requires careful analysis of numerous possible design variables by generating large number of simulation runs using detailed energy simulation programs. This task can only be tackled by computers, which have the advantage of computational speed, parallel processing, and accuracy. In MCDM problems, the searching of a single optimal solution is of little value, since the objectives are often competitive. So a purely optimization based technique is inadequate, instead, an interactive procedure involving the decision maker is required to determine near-optimal Satisficing (Satisfy + Suffice) solutions. The need to address multi-criteria requirements makes it more valuable for a designer to know the "latitude" or "degrees of freedom" he has in changing certain design variables while achieving satisfactory levels of energy performance. Currently such a design framework is lacking and hence this thesis proposes an alternative methodology [VADSM] for low energy building design evaluation. VADSM supports the two key elements of the MCDM process, namely search and decision making, using data mining techniques and interactive visualizations respectively. A custom software interface [DSMV] has been developed for enabling interactive and dynamic decision making.

Two different design feature sets were selected to evaluate and contrast different analytical techniques incorporated in VADSM. The small feature set was made up of 5 design variables while the larger set contained 15 variables. The simulated responses in both cases were building EUI in KBtu/sqft/yr. and PED in kW/yr. The purpose of picking the small feature set was to test whether traditional techniques like OLS Linear Regression would provide superior results when dealing with fewer variables, over an inductive statistical learning technique like CART. Conversely, a nonlinear technique like RF was expected to be more effective for feature selection and modeling the higher dimensional feature set that is much sparser and hence technically not a good candidate for fitting a single global linear regression model.

The results obtained from analyzing the small feature set suggested, that if high prediction accuracy and explicit analytical models are required then, OLS regression is the better choice. However if visual interpretability is more desirable and a reasonable reduction in predictive ability can be tolerated, then CART is a viable alternative, especially in the case of categorical variables and/or a non-continuous (jagged) solution space. CART is also well suited for classification problems where the output is categorical. Additionally, CART also provides clear variable ranking (both visual and statistical) that can used to eliminate redundant ones and create simpler predictive models.

For the small design feature set OLS Models built on the CCD runs had excellent predictive ability when compared to similar OLS models built on exhaustive variable combinations. This implies a significant potential for reduction in simulation run time. Hence for OLS regression modeling, CCD or other such experimental design techniques should be explored wherever possible instead of exhaustive enumeration. However, CART being a statistical induction technique, which fits local models to variations in data, the CCD approach may not yield sufficient data to train a useful decision tree.

With the small feature set an exhaustive combination of variables could be simulated for analysis. However, with the larger feature set this was not feasible since the time need to simulate the hundreds of thousands of runs would be prohibitive. So sampling methods such as LHRS were used to sample fewer, but representative runs from the entire solution space. The selection of a specific number of representative simulations to run for a given set of independent variable combinations can be addressed as a tradeoff between generating adequate training data for statistical models and reducing computer run time. Experimental design is a well-established technique that helps in this regard and specific techniques applicable for computer experiments should be further explored.

Separate RF ensembles of 500 trees each were built for both design responses. RF was chosen due to its robust variable ranking technique. The strength of this technique lies in the fact that it ranks variables based on their predictive ability using OOB (Out-of-Bag) test data. Once the important predictors were identified pairs of OLS models (one for each response) were fit to successively larger subsets (starting with 5) of the top ranked variables for EUI and PED prediction , till established model performance criteria was met (CV < 10 % and R^2 > 90%). Although the RF models themselves had acceptable predictive ability they are very difficult to visualize. Hence after feature selection simpler OLS regression models were chosen instead due to their analytically explicit equations that could be visually incorporated in the DSMV application.

Model selection parameters such R^2 , CV, RF ensemble size should be reflective of the objectives of the designer in evaluating alternatives. For preliminary design a relatively lower R^2 might suffice, however, for detailed design evaluation higher prediction accuracy would be more appropriate.

7.2 Future work

7.2.1 Methodological Improvements

This study was limited to 15 key independent design variables relevant to a mid-size commercial building; however, high performance building design can often involve much larger design feature sets in the range of 50-100 variables. Thus a natural extension of this research would be to explore the analysis of such large feature sets involved in the design of more complex buildings. Clearly, appropriate feature selection techniques would be critical in such cases to identify the most relevant variables and reduce the complexity of any prediction models. Similarly, the number of objective functions (design criteria) could be increased in order to evaluate additional relevant design criteria such as primary and secondary HVAC components, life cycle costing, CO2 emissions, day lighting, comfort etc. A strategy of applying user defined weights to each criterion can help in prioritizing the relative impacts of design decisions.

Additionally the scope of the DSMV application can be expanded to include categorical predictors like HVAC system types, control strategies, material types etc. Feedback from different types of users such as architects, energy engineers, environmental designers etc. would be helpful for improving the workflow and interface design

While the present research explored the application of VADSM to new building design only, the proposed methodology is also well suited for evaluating energy efficiency retrofits or identifying and improving operational deficiencies in existing buildings, provided a well-calibrated energy model with a set of relevant independent variables and ranges is available.

7.2.2 Software Improvements

Integration of the DSMV application as a module or add-in with an existing energy simulation tool will reduce potential interoperability errors and allow for faster iterations. Moving beyond energy there is a wide range of other performance criteria such as comfort, economics, safety, environmental impact etc. Links to such simulation tools will provide additional information to further enhance the decision-making process. Future research in this area should also explore the benefits of the rapid and parallelized computing power of cloud-based infrastructure to run large batch simulations in near real time. A more sophisticated interface that can inform the designer of simultaneous changes to multiple objectives while allowing them to adjust the level of detail appropriate to the specific task would be a future improvement. Advances in the fields of high dimensional data visualization and GUI design could provide further innovative data visualization solutions.

REFERENCES

Addison, M. S. (1988). *A multiple criteria satisficing methodology for the design of energy-efficient buildings.* (Unpublished Master of Environmental Planning). Arizona State University, Tempe.

Agarwal, R., Mannial, H., Srikant, R., Toivonen, H., & Verkamo, I. (1996). Fast discovery of association rules. *Advances in knowledge discovery and data mining* (pp. 307-328). Menlo Park, CA, USA: American Association for Artificial Intelligence.

Bae, C. (2008). *Analyzing random forests.* (Unpublished Doctor of Philosophy). UNIVERSITY OF CALIFORNIA, BERKELEY, Berkeley.

Berger, P. D., & Maurer, R. E. (2002). *Experimental design: With applications in management, engineering, and the sciences*. Belmont, CA: Duxbury/Thomson Learning.

Berk, R. A. (2008). *Statistical learning from a regression perspective*. New York, NY: Springer Verlag.

Bertini, E., Tatu, A., & Keim, D. (2011). Quality metrics in high-dimensional data visualization: An overview and systematization. *Visualization and Computer Graphics, IEEE Transactions On, 17*(12), 2203-2212. doi:10.1109/TVCG.2011.229

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123-140. doi:10.1007/BF00058655

Breiman, L. (1998). Arcing classifiers. *The Annals of Statistics, 26*(3), 801-824.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. doi:10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Belmont, Calif.: Wadsworth International Group.

Caldas, L. G., & Norford, L. K. (2003). Genetic algorithms for optimization of building envelopes and the design and control of HVAC systems. *Journal of Solar Energy Engineering, 125*(3), 343-351. doi:10.1115/1.1591803

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20,* 273.

Crawley, D. B., Hand, J. W., Kummert, M., & Griffith, B. T. (2008). Contrasting the capabilities of building energy performance simulation programs. *Building and Environment, 43*(4), 661-673. doi:10.1016/j.buildenv.2006.10.027

D'Cruz, N. A., & Radford, A. D. (1987). A multicriteria model for building performance and design. *Building and Environment, 22*(3), 167-179. doi:http://dx.doi.org.ezproxy1.lib.asu.edu/10.1016/0360-1323(87)90005-9

Diakaki, C., Grigoroudis, E., & Kolokotsa, D. (2008). Towards a multi-objective optimization approach for improving energy efficiency in buildings. *Energy and Buildings, 40*(9), 1747-1754. doi:http://dx.doi.org.ezproxy1.lib.asu.edu/10.1016/j.enbuild.2008.03.002

Dielman, T. E. (2001). *Applied regression analysis for business and economics* (3rd ed.). Pacific Grove, CA: Duxbury/Thomson Learning.

Dong, B., Cao, C., & Lee, S. E. (2005). Applying support vector machines to predict building energy consumption in tropical region. *Energy and Buildings, 37*(5), 545-553. doi:10.1016/j.enbuild.2004.09.009

Fayyad, U. (1997). Data mining and knowledge discovery in databases: Implications for scientific databases. *Scientific and Statistical Database Management, 1997. Proceedings., Ninth International Conference On,* 2-11. doi:10.1109/SSDM.1997.621141

Fayyad, U. M., Grinstein, G. G., Wierse, A., & NetLibrary, I. (2002). *Information visualization in data mining and knowledge discovery*. San Francisco: MK/Morgan Kaufmann Publishers.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). Advances in knowledge discovery and data mining. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.), (pp. 1-34). Menlo Park, CA, USA: American Association for Artificial Intelligence.

Foucquier, A., Robert, S., Suard, F., Stéphan, L., & Jay, A. (2013). State of the art in building modelling and energy performances prediction: A review. *Renewable and Sustainable Energy Reviews, 23*(0), 272-288. doi:10.1016/j.rser.2013.03.004

Genuer, R., Poggi, J., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters, 31*(14), 2225-2236. doi:10.1016/j.patrec.2010.03.014

Gero, J. S., D'Cruz, N., & Radford, A. D. (1983). Energy in context: A multicriteria model for building design. *Building and Environment, 18*(3), 99-107. doi:http://dx.doi.org.ezproxy1.lib.asu.edu/10.1016/0360-1323(83)90001-X

Grinstein, G., Hoffmann, P., Pickett, R. (2002). Benchmark development for the evaluation of visualization for data mining. *Information visualization in data mining and knowledge discovery* (pp. 129). San Francisco, Calif.: Morgan Kaufmann.

Grömping, U. (2009). Variable importance assessment in regression: Linear regression versus random forest. *The American Statistician, 63*(4), 308-319. doi:10.1198/tast.2009.08199

Haberl, J., Sparks, R., & Culp, C. (1996). Exploring new techniques for displaying complex building energy consumption data. *Energy and Buildings, 24*(1), 27-38. doi:10.1016/0378-7788(95)00959-0

Haberl, J. S., Bronson, D., Hinchey, S., & O'Neal, D. (1993). Graphical tools to help calibrate the DOE-2 simulation program. *ASHRAE Journal, Jan*

Haberl, J., & Abbas, M. (1998). Development of graphical indices for displaying large scale building energy data sets<br />   <br /> . *Journal of Solar Energy Engineering, 120*(3), 156.

Han, J., & Kamber, M. (2001). *Data mining: Concepts and techniques*. San Francisco: Morgan Kaufmann Publishers.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York: Springer.

Helton, J. C., & Davis, F. J. (2003). Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems. *Reliability Engineering & System Safety, 81*(1), 23-69. doi:http://dx.doi.org.ezproxy1.lib.asu.edu/10.1016/S0951-8320(03)00058-9

Hoffmann, P., & Grinstein, G. (2002). A survey of visualizations for high dimensional data mining. *Information visualization in data mining and knowledge discovery* (pp. 47). San Francisco, Calif.: Morgan Kaufmann.

Ishwaran, H., Kogalur, U. B., Blackstone, E. H., & Lauer, M. S. (2008). Random survival forests. *The Annals of Applied Statistics, 2*(3), 841-860.

Kalogirou, S. A. (2000). Applications of artificial neural-networks for energy systems. *Applied Energy, 67*(1–2), 17-35. doi:10.1016/S0306-2619(00)00005-2

Kawashima, M., Dorgan, C. E., & Mitchell, J. W. (1995). Hourly thermal load prediction for the next 24 hours by ARIMA, EWMA, LR and an artificial neural network. *ASHRAE Trans., 101*(Part 1), 186-200.

Keim, D. A. (2001). VISUAL EXPLORATION of LARGE DATA SETS. *Communications of the ACM, 44*(8)

Keim, D. A., & Andrienko, G. (2008). <br />Visual analytics: Definition, process, and challenges. *Information visualization: Human-centered issues and perspectives* (pp. 154). Berlin ; New York: Springer.

Keim, D. A., Kohlhammer, J., Ellis, G., & Mansmann, F. (Eds.). (2010). *Mastering the information Age Solving problems with Visual analytics*. Germany: Eurographics.

Keim, D. A. (2000). Designing pixel-oriented visualization techniques: Theory and applications. *Visualization and Computer Graphics, IEEE Transactions On, 6*(1), 59-78. doi:10.1109/2945.841121

Keim, D. A., Mansmann, F., Schneidewind, J., & Ziegler, H. (2006). Challenges in visual data analysis. *Information Visualization, 2006. IV 2006. Tenth International Conference On,* 9-16. doi:10.1109/IV.2006.31

Keim, D. (1995). *Visual support for query specification and data mining*. Aachen: Verlag Shaker.

Kim, H., Stumpf, A., & Kim, W. (2011). Analysis of an energy efficient building design through data mining approach. *Automation in Construction, 20*(1), 37-43. doi:10.1016/j.autcon.2010.07.006

Kleinbaum, D. G. (2008). *Applied regression analysis and other multivariable methods* (4th ed.). Belmont, Calif.: Thomson Brooks/Cole Publishing.

Krarti, M., Kreider, J. F., Cohen, D., & Curtiss, P. (1998). Estimation of energy savings for building retrofits using neural networks. *Journal of Solar Energy Engineering, 120*(3), 211-216. doi:10.1115/1.2888071

Lam, J. C., & Hui, S. C. M. (1996). Sensitivity analysis of energy performance of office buildings. *Building and Environment, 31*(1), 27-39. doi:10.1016/0360-1323(95)00031-3

LBNL. (2006). Building design advisor homepage. Retrieved 4/20, 2013, from http://gaia.lbl.gov/BDA/bdainfo.htm

Lee, W., House, J. M., & Kyong, N. (2004). Subsystem level fault diagnosis of a building's air-handling unit using general regression neural networks. *Applied Energy, 77*(2), 153-170. doi:10.1016/S0306-2619(03)00107-7

Li, Q., Meng, Q., Cai, J., Yoshino, H., & Mochida, A. (2009). Applying support vector machine to predict hourly cooling load in the building. *Applied Energy, 86*(10), 2249-2256. doi:10.1016/j.apenergy.2008.11.035

Lundin, M., Andersson, S., & Östin, R. (2004). Development and validation of a method aimed at estimating building performance parameters. *Energy and Buildings, 36*(9), 905-914. doi:10.1016/j.enbuild.2004.02.005

Mckay, M. D., Beckman, R. J., & Conover, W. J. (2000). A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics, 42*(1, Special 40th Anniversary Issue), 55-61.

Mihalisin, T. (2002). Multidimensional education : Visual and algorithmic data mining domains and symbiosis. *Information visualization in data mining and knowledge discovery* (pp. 305). San Francisco, Calif.: Morgan Kaufmann.

Morbitzer, C. A. (2003). *<br />Towards the integration of simulation into the building design process<br />* . (Unpublished Doctor of Philosophy). University of Strathclyde,

Ngatchou, P., Anahita Zarei, & El-Sharkawi, M. A. (2005). Pareto multi objective optimization. *Intelligent Systems Application to Power Systems, 2005. Proceedings of the 13th International Conference On,* 84-91. doi:10.1109/ISAP.2005.1599245

NREL, PNNL, & US DoE. (2011). *U.S. department of energy commercial reference building models of the national building stock.* (No. NREL/TP-5500-46861). Colorado: NREL.

Olofsson, T., & Andersson, S. (2001). Long-term energy demand predictions based on short-term measured data. *Energy and Buildings, 33*(2), 85-91. doi:10.1016/S0378-7788(00)00068-2

Pak Chung Wong, & Thomas, J. (2004). Visual analytics. *Computer Graphics and Applications, IEEE, 24*(5), 20-21. doi:10.1109/MCG.2004.39

Papamichael, K. (1999). Application of information technologies in building design decisions. *Building Research & Information, 27*(1), 20-34.

Parsaye, K., & Chignell, M. (1993). *Intelligent database tools & applications: Hyperinformation access, data quality, visualization, automatic discovery*. New York: Wiley.

Peterson, A. R. (2009). *Visual data mining: Using parallel coordinate plots with k-means clustering and color to find correlations in a multidimensional dataset.* (Unpublished Master of Science). Kutztown Univeristy of Pennsylvania, Kutztown.

Reddy, T. A. (2011a). Calibration of white box models. *Applied data analysis and modeling for energy engineers and scientists* (pp. 333) Springer.

Reddy, T. A. (2011b). Design of experiments. *Applied data analysis and modeling for energy engineers and scientists* (pp. 201) Springer.

Reddy, T. A. (2011c). Estimation of linear model parameters using least squares. *Applied data analysis and modeling for energy engineers and scientists* (pp. 141) Springer.

Reddy, T. A., Maor, I., & Panjapornpon, C. (2007). Calibrating detailed building energy simulation programs with measured data--part II: Application to three case study office buildings (RP-1051). *HVAC&R Research, 13*(2), 243-265.

Siirtola, H., & Räihä, K. (2006). Interacting with parallel coordinates. *Interacting with Computers, 18*(6), 1278-1309. doi:10.1016/j.intcom.2006.03.006

Simon, H. A. (1957). *Models of man:Social and rational; mathematical essays on rational human behavior in society setting*. New York: Wiley.

Tan, P., Steinbach, M., & Kumar, V. (2005). *Introduction to data mining* (1st ed.). Boston: Pearson Addison Wesley.

Thomas, J. J., & Cook, K. A. (2006). A visual analytics agenda. *Computer Graphics and Applications, IEEE, 26*(1), 10-13. doi:10.1109/MCG.2006.5

Thornton, B., Wang, W., Lane, M., Rosenberg, M., & Liu, B. (2009). *50% energy savings design technology packages for medium office buildings.* (Technical Support Document No. PNNL-19004). Richland, Washington: Pacific Northwest National Laboratory.

Tominski, C., Abello, J., & Schumann, H. (2009). CGV—An interactive graph visualization system. *Computers & Graphics, 33*(6), 660-678. doi:10.1016/j.cag.2009.06.002

Tsanas, A., & Xifara, A. (2012). Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools. *Energy and Buildings, 49*(0), 560-567. doi:10.1016/j.enbuild.2012.03.003

Tso, G. K. F., & Yau, K. K. W. (2007). Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks. *Energy, 32*(9), 1761-1768. doi:10.1016/j.energy.2006.11.010

Tufte, E. R. (2001). *The visual display of quantitative information* (2nd ed.). Cheshire, Conn.: Graphics Press.

Tukey, J. W. (1977). *Exploratory data analysis*. Reading, Ma.: Addison-Wesley.

Ware, C. (2000). *Information visualization: Perception for design*. San Francisco ; London: Morgan Kaufman.

Wong, S. L., Wan, K. K. W., & Lam, T. N. T. (2010). Artificial neural networks for energy analysis of office buildings with daylighting. *Applied Energy, 87*(2), 551-557. doi:10.1016/j.apenergy.2009.06.028

Wright, J., & Loosemore, H. (2001). The multi-criterion optimization of building thermal design and control. *Optimization, 2*, 873-880.

Yu, Z., Haghighat, F., Fung, B. C. M., & Yoshino, H. (2010). A decision tree method for building energy demand modeling. *Energy and Buildings, 42*(10), 1637-1646. doi:10.1016/j.enbuild.2010.04.006

Zhao, H., & Magoulès, F. (2012). A review on the prediction of building energy consumption. *Renewable and Sustainable Energy Reviews, 16*(6), 3586-3592. doi:10.1016/j.rser.2012.02.049

Zionts, S. (1979). MCDM: If not a roman numeral, then what? *Interfaces, 9*(4), 94-101.

APPENDIX A

DOE2 VARIABLE DEFINITIONS

LIGHTING-W/AREA (LPD)

Takes a list of values of electric lighting power per unit space floor area, including ballasts where applicable, for each of up to 5 lighting subsystems. Whether or not the subsystem serves the entire space, the value entered should be the total peak power for the subsystem (before multiplying by the lighting schedule) divided by the total space floor area. Each subsystem can have a different lighting schedule (see LIGHTING-SCHEDULE, below). Example input: if a space has two lighting subsystems, one with 0.5 W/ft2 and the other with 1.0 W/ft2, then LIGHTING-W/AREA = (0.5, 1.0).

EQUIPMENT-W/AREA (EPD)

Takes a list of values of maximum equipment power per unit floor area of the space for up to five types of electrical equipment. An alternative to EQUIPMENT-KW. If both EQUIPMENT-W/AREA and EQUIPMENT-KW are specified, the contributions are added. The program calculates the electrical power in watts for all of the equipment in a space as

Watts = (EQUIPMENT-KW * 1000
+ AREA * EQUIPMENT-W/AREA) * (EQUIP-SCHEDULE value)

for all equipment types.

The EQUIP-SENSIBLE and EQUIP-LATENT keywords give the fraction of heat gain for each equipment type that is sensible and latent, respectively. The total hourly heat gain from all of the equipment (for all of the equipment types) is, therefore:

Qwatts = Watts * (EQUIP-SENSIBLE + EQUIP-LATENT)

RESISTANCE (Wall-R/Roof-R)

The thermal resistance of the material.

GLASS-CONDUCTANCE (Win-U)

The conductance of the glazing, excluding the outside air film coefficient. The conductance given in glass manufacturers' data sheets usually includes the outside air film resistance for a wind speed of 7.5 mph (summer) or 15 mph (winter).

SHADING-COEF (SC)

When TYPE=SHADING-COEF is entered, the program first calculates the solar heat gain using transmission and absorption coefficients for a reference glazing (clear, 1/8" thick, single-pane, double-strength sheet glass). This solar heat gain is then multiplied by the value of SHADING-COEF to determine the resultant solar heat gain.

The shading coefficient depends, in general, not only on the type of glass but also on whether blinds, shades, draperies, etc., are used with the window. To simulate operable shading devices, assign a SHADING-SCHEDULE to a window (see the WINDOW command) and the resultant solar heat gain each hour will be multiplied by the schedule value.

For shading coefficient values of different glazing types with and without shading devices, see manufacturers' data sheets or the ASHRAE 1989 Handbook of Fundamentals, p. 27.26ff.

We strongly recommend that exterior WINDOWs in a sunspace be described with TYPE=GLASS-TYPE-CODE rather than SHADING-COEF. This allows the program to accurately calculate the hourly direct and diffuse radiation transmitted by the glazing. This is not possible with SHADING-COEF except for standard 1/8" clear glass.

HEIGHT (Win-Ht)

Height of the glazed part of the window.

FURNACE-HIR (Furn-Eff)

Ratio of fuel used by the furnace (including that used by the pilot light, if present) to the heating energy produced. In calculating this ratio, the fuel used and heating produced should be expressed in the same units.

COOLING-EIR (Cool-EIR)

The Electric Input Ratio (EIR), or 1/(Coefficient of Performance), for the cooling unit at ARI rated conditions. The program defines EIR to be the ratio of the electric energy input to the rated capacity, when both the energy input and rated capacity are expressed in the same units. This EIR is at ARI rated conditions, i.e., without correction for different temperature or part load.

Note: If you include fan electric energy consumption in your value of COOLING-EIR, then you should set SUPPLY-KW/FLOW to zero (and SUPPLY-STATIC, SUPPLY-EFF and SUPPLY-DELTA-T should be omitted). Otherwise, the supply fan electrical energy will be double counted. For commercial systems the default value of COOLING-EIR includes compressor and outdoor fan energy, but not indoor fan energy. Imbedding the fan energy into the COOLING-EIR is valid only if the fan is constant volume and INDOOR-FAN-MODE = INTERMITTENT; i.e. the fan cycles on/off with the compressor. If the fan runs continuously during occupied hours, or the fan is variable volume, then the fan energy cannot be included in the COOLING-EIR (or HEATING-EIR).

SUPPLY-STATIC (Fan-Pres)

Total static pressure of the supply fan at design flow rate. Pressure losses should include filters, coils, fan housing, and distribution system. Use either SUPPLY-STATIC and SUPPLY-EFF or SUPPLY-DELTA-T and SUPPLY-KW/FLOW.

MIN-FLOW-RATIO (Min-FlowR)

Minimum allowable zone air supply flow rate, expressed as a fraction of design flow rate. Applicable to variable-volume type systems only. This keyword also appears in the SYSTEM command, where it is a system level keyword that applies to all zones in the system. Here, it is a zone level keyword that applies only to this zone, allowing different MIN-FLOW-RATIOs for each zone. MIN-FLOW-RATIO can be scheduled using ZONE:MIN-FLOW-SCH.

If the sum of the MIN-FLOW-RATIOs of all the zones times the design flow rate is less than the specified outside air flow rate, there is implied 100 per cent outside air operation at, and possibly above, the zone MIN-FLOW-RATIO. In other words, it may be necessary for the system to operate at 100% outside air at very low airflows in order to satisfy the ventilation requirements.

If THERMOSTAT-TYPE = REVERSE-ACTION is not specified, zone MIN-FLOW-RATIO is also the flow rate fraction in the heating mode. The VAV box will modulate its airflow between the top and bottom of the cooling setpoint throttling range, and be at the minimum flow at all temperatures below the cooling throttling range. Care must be taken to specify a reasonable MIN-FLOW-RATIO in this case. Depending on the value of the MIN-FLOW-RATIO, the system may not have enough reheat capacity. Additionally, the introduction of a small amount of (low velocity) warm air at the ceiling level may cause temperature stratification problems in many buildings. To avoid this, the THERMOSTAT-TYPE should be REVERSE-ACTION, or an HMIN-FLOW-RATIO can be specified to establish a higher flow ratio during heating.

For dual-duct systems, MIN-FLOW-RATIO is the flow ratio at the outlet of the mixing box, and should be specified only if the box has a controller measuring air flow at the outlet. HMIN-FLOW-RATIO and CMIN-FLOW-RATIO specify the minimum air flows at the inlets to the mixing box (hot and cold decks, respectively). You should refer to the discussion of these keywords in the SYSTEM command for more information.

AIR-CHANGES/HR (Infil-AC)

The number of infiltration-caused air changes per hour at a reference wind speed of 10 mph (4.47 m/s) (see table under INF-METHOD). It has a correction for wind speed as shown by the following equation.

Infiltration in ach = (AIR-CHANGES/HR) * (wind-speed)/(reference wind speed)

MIN-OUTSIDE-AIR (Min OA)

The minimum outside air flow divided by the supply air flow during winter heating periods. You may alternatively, or additionally, specify outside air quantities at the zone level using the ZONE keywords OA-CHANGES, OA-FLOW/PER or OUTSIDE-AIR-FLOW. The default is calculated from zone loads and ZONE input.

If you enter MIN-OUTSIDE-AIR as well as the ZONE keywords OUTSIDE-AIR-FLOW, OA-CHANGES, OA-FLOW/PER or EXHAUST-FLOW, the ZONE values take precedence. If no zone-level values are specified, MIN-OUTSIDE-AIR will be used. If MIN-AIR-SCH is specified, MIN-OUTSIDE-AIR, or the corresponding ZONE values, should be entered.

The program will not allow MIN-OUTSIDE-AIR to be less than the sum of the EXHAUST-FLOWs for all zones divided by the sum of all supply flows for all zones. That is, the exhaust fan operation will override MIN-OUTSIDE-AIR if MIN-OUTSIDE-AIR is set too low.

The minimum outside air ratio reported on SV-A is based on the design calculated supply air flow and not on the value input for SUPPLY-FLOW, which overrides the design flow rate.

When evaporative cooling is in effect, the outside air dampers are 100% open. When outside air is able to cool the building without the aid of evaporative cooling, the outside and return air dampers modulate open.

MIN-SUPPLY-T (Min-CoolT)

For systems that can provide cooling, this is a required keyword that gives the minimum temperature of the air delivered to the zone. MIN-SUPPLY-T and COOL-DESIGN-T are used to size the capacity of the cooling coil and supply air flow rate. The supply air flow rates needed to satisfy the heating and cooling requirements are compared and the greater of the two quantities is used for the system air flow rate. Note that MIN-SUPPLY-T also controls the amount of moisture that can be removed by the cooling coils.

Note, that for those systems that are to maintain a constant cooling air discharge temperature (see keyword COOL-CONTROL), the control set point is determined by the value entered for COOL-SET-T rather than MIN-SUPPLY-T. In this case, the program uses MIN-SUPPLY-T to limit subcooling for dehumidification purposes (and to calculate the design air flow rate for cooling).

Note that MIN-SUPPLY-T is the design supply temperature *at the zone*, downstream of duct losses. COOL-SET-T, COOL-SET-SCH, COOL-RESET-SCH and the heating counterparts are all defined as *entering the duct*, upstream of duct losses. As such, they should be adjusted for the expected duct losses so that the hourly supply temperature at the zone is the desired temperature.

MAX-SUPPLY-T (Max-HeatT)

The maximum allowable temperature (i.e., maximum diffuser temperature) of the air delivered to the zones in a system. MAX-SUPPLY-T and DESIGN-HEAT-T are used to size the supply air flow rate and the capacity of the heating coil. MAX-SUPPLY-T, which should be greater that DESIGN-HEAT-T, is also used as an upper limit for supply air temperature control.

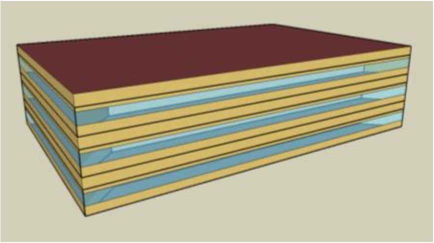
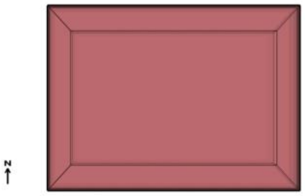
This entry is mandatory for certain types of systems (e.g., RESYS, MZS, DDS, SZCI, UVT, UHT, HP, HVSYS, FC, IU, PSZ, PMZS, PTAC) and optional for other types of systems (SZRH, VAVS, RHFS, CBVAV, PVAVS). If no entry is made, the program will use the sum of MIN-SUPPLY-T and REHEAT-DELTA-T.

Note that MAX-SUPPLY-T is the design supply temperature at the zone, downstream of duct losses. HEAT-SET-T, HEAT-SET-SCH, HEAT-RESET-SCH and the cooling counterparts are all defined as entering the duct, upstream of duct losses. As such, they should be adjusted for the expected duct losses so that the hourly supply temperature at the zone is the desired temperature.

Source: DOE2 Online Help File eQuest version 3.64

APPENDIX B

ASHRAE 90.1 PROTOTYPE BUILDING MODELING SPECIFICATIONS

	Item	Descriptions	Data Source
Program			
	Vintage	**NEW CONSTRUCTION**	
	Location (Representing 8 Climate Zones)	Zone 1A: Miami (very hot, humid) Zone 1B: Riyadh, Saudi Arabia (very hot, dry) Zone 2A: Houston (hot, humid) Zone 2B: Phoenix (hot, dry) Zone 3A: Memphis (warm, humid) Zone 3B: El Paso (warm, dry) Zone 3C: San Francisco (warm,marine) Zone 4A: Baltimore (mild, humid) Zone 4B: Albuquerque (mild, dry) Zone 4C: Salem (mild, marine) Zone 5A: Chicago (cold, humid) Zone 5B: Boise (cold, dry) Zone 5C: Vancouver, BC (cold, marine) Zone 6A: Burlington (cold, humid) Zone 6B: Helena (cold, dry) Zone 7: Duluth (very cold) Zone 8: Fairbanks (subarctic)	Selection of representative climates based on Briggs' paper. See Reference.
	Available fuel types	gas, electricity	
	Building Type (Principal Building	**OFFICE**	
	Building Prototype	**Medium Office**	
Form			
	Total Floor Area (sq feet)	53,600 (163.8 ft x 109.2 ft)	
	Building shape		
	Aspect Ratio	1.5	
	Number of Floors	3	
	Window Fraction (Window-to-Wall Ratio)	33% (Window Dimensions: 163.8 ft x 4.29 ft on the long side of facade 109.2 ft x 4.29 ft on the short side of the façade)	2003 CBECS Data and PNNL's CBECS Study 2007.
	Window Locations	even distribution among all four sides	
	Shading Geometry	none	
	Azimuth	non-directional	
	Thermal Zoning	Perimeter zone depth: 15 ft. Each floor has four perimeter zones and one core zone. Percentages of floor area: Perimeter 40%, Core 60%  	
	Floor to floor height (feet)	13	
	Floor to ceiling height (feet)	9 (4 ft above-ceiling plenum)	
	Glazing sill height (feet)	3.35 ft (top of the window is 7.64 ft high with 4.29 ft high glass)	

Architecture		
Exterior walls		
Construction	Steel-Frame Walls (2X4 16IN OC) 0.4 in. Stucco+5/8 in. gypsum board + wall Insulation+5/8 in.	Construction type: 2003 CBECS Data and PNNL's CBECS Study 2007. Exterior wall layers: default 90.1 layering
U-factor (Btu / h * ft^2 * °F) and/or R-value (h * ft^2 * °F / Btu)	ASHRAE 90.1 Requirements Nonresidential; Walls, Above-Grade, Steel-Framed	ASHRAE 90.1
Dimensions	based on floor area and aspect ratio	
Tilts and orientations	vertical	
Roof		
Construction	Built-up Roof: Roof membrane+Roof insulation+metal decking	Construction type: 2003 CBECS Data and PNNL's CBECS Study 2007. Roof layers: default 90.1 layering
U-factor (Btu / h * ft^2 * °F) and/or R-value (h * ft^2 * °F / Btu)	ASHRAE 90.1 Requirements Nonresidential; Roofs, Insulation entirely above deck	ASHRAE 90.1
Dimensions	based on floor area and aspect ratio	
Tilts and orientations	horizontal	
Window		
Dimensions	based on window fraction, location, glazing sill height, floor area and aspect ratio	
Glass-Type and frame	Hypothetical window with the exact U-factor and SHGC shown below	
U-factor (Btu / h * ft^2 * °F)	ASHRAE 90.1 Requirements Nonresidential; Vertical Glazing, 31.1-40%, U_fixed	ASHRAE 90.1
SHGC (all)		
Visible transmittance	Hypothetical window with the exact U-factor and SHGC shown above	
Operable area	0	Ducker Fenestration Market Data provided by the 90.1 envelope subcommittee
Skylight		
Dimensions	Not Modeled	
Glass-Type and frame	NA	
U-factor (Btu / h * ft^2 * °F)		
SHGC (all)		
Visible transmittance		
Foundation		
Foundation Type	Slab-on-grade floors (unheated)	
Construction	8" concrete slab poured directly on to the earth	
Thermal properties for ground level floor U-factor (Btu / h * ft2 * °F) and/or R-value (h * ft2 * °F / Btu)	ASHRAE 90.1 Requirements Nonresidential; Slab-on-Grade Floors, unheated	ASHRAE 90.1
Thermal properties for basement	NA	
Dimensions	based on floor area and aspect ratio	
Interior Partitions		
Construction	2 x 4 uninsulated stud wall	
Dimensions	based on floor plan and floor-to-floor height	
Internal Mass	6 inches standard wood (16.6 lb/ft²)	
Air Barrier System		
Infiltration	Peak: 0.2016 cfm/sf of above grade exterior wall surface area (when fans turn off) Off Peak: 25% of peak infiltration rate (when fans turn on)	Reference: PNNL-18898: Infiltration Modeling Guidelines for Commercial Building Energy Analysis.

HVAC			
	System Type		
	Heating type	Gas furnace inside the packaged air conditioning unit	2003 CBECS Data, PNNL's CBECS Study 2006, and 90.1 Mechanical Subcommittee input.
	Cooling type	Packaged air conditioning unit	
	Distribution and terminal units	VAV terminal box with damper and electric reheating coil Zone control type: minimum supply air at 30% of the zone design peak supply air.	
	HVAC Sizing		
	Air Conditioning	autosized to design day	
	Heating	autosized to design day	
	HVAC Efficiency		
	Air Conditioning	Various by climate location and design cooling capacity ASHRAE 90.1 Requirements Minimum equipment efficiency for Air Conditioners and Condensing Units	ASHRAE 90.1
	Heating	Various by climate location and design heating capacity ASHRAE 90.1 Requirements Minimum equipment efficiency for Warm Air Furnaces	ASHRAE 90.1
	HVAC Control		
	Thermostat Setpoint	75°F Cooling/70°F Heating	
	Thermostat Setback	80°F Cooling/60°F Heating	
	Supply air temperature	Maximum 104F, Minimum 55F	
	Chilled water supply temperatures	NA	
	Hot water supply temperatures	NA	
	Economizers	Various by climate location and cooling capacity Control type: differential dry bulb	ASHRAE 90.1
	Ventilation	ASHRAE Ventilation Standard 62.1 See under **Outdoor Air**.	ASHRAE Ventilation Standard 62.1
	Demand Control Ventilation	ASHRAE 90.1 Requirements	ASHRAE 90.1
	Energy Recovery	ASHRAE 90.1 Requirements	ASHRAE 90.1
	Supply Fan		
	Fan schedules	See under **Schedules**	
	Supply Fan Total Efficiency (%)	60% to 62% depending on the fan motor size	ASHRAE 90.1 requirements for motor efficiency and fan power limitation
	Supply Fan Pressure Drop	Various depending on the fan supply air cfm	
	Pump		
	Pump Type	NA	
	Rated Pump Head	NA	
	Pump Power	autosized	
	Cooling Tower		
	Cooling Tower Type	NA	
	Cooling Tower Efficiency	NA	
	Service Water		
	SWH type	Storage Tank	
	Fuel type	Natural Gas	
	Thermal efficiency (%)	ASHRAE 90.1 Requirements Water Heating Equipment, Gas storage water heaters, >75,000 Btu/h input	ASHRAE 90.1
	Tank Volume (gal)	260	
	Water temperature setpoint	120F	
	Water consumption	See under **Schedules**	

Internal Loads & Schedules			
	Lighting		
	Average power density (W/ft^2)	ASHRAE 90.1 Lighting Power Densities Using the Building Area Method	ASHRAE 90.1
	Schedule	See under **Schedules**	
	Daylighting Controls	ASHRAE 90.1 Requirements	
	Occupancy Sensors	ASHRAE 90.1 Requirements	
	Plug load		
	Average power density (W/ft^2)	See under **Zone Summary**	User's Manual for ASHRAE Standard 90.1-2004 (Appendix G)
	Schedule	See under **Schedules**	
	Occupancy		
	Average people	See under **Zone Summary**	User's Manual for ASHRAE Standard 90.1-2004 (Appendix G)
	Schedule	See under **Schedules**	
Misc.			
	Elevator		
	Quantity	2	Reference: DOE Commercial Reference Building Models of the National Building Stock
	Motor type	hydraulic	
	Peak Motor Power (W/elevator)	16,055	
	Heat Gain to Building	Interior	
	Peak Fan/lights Power (W/elevator)	161.9	90.1 Mechanical Subcommittee, Elevator Working Group
	Motor and fan/lights Schedules	See under **Schedules**	DOE Commercial Reference Building TSD and models (V1.3_5.0) and Addendum DF to 90.1-2007
	Exterior Lighting		
	Peak Power (W)	14,385	ASHRAE 90.1
	Schedule	See under **Schedules**	

References

Briggs, R.S., R.G. Lucas, and Z.T. Taylor. 2003. Climate Classification for Building Energy Codes and Standards: Part 2—Zone Definitions, Maps, and Comparisons. ASHRAE Transactions 109(2).

PNNL's CBECS Study. 2007. *Analysis of Building Envelope Construction in 2003 CBECS Buildings.* Dave Winiarski, Mark Halverson, and Wei Jiang. Pacific Northwest National Laboratory. March 2007.

PNNL's CBECS Study. 2006. *Review of Pre- and Post-1980 Buildings in CBECS – HVAC Equipment.* Dave Winiarski, Wei Jiang and Mark Halverson. Pacific Northwest National Laboratory. December 2006.

Gowri K, DW Winiarski, and RE Jarnagin. 2009. Infiltration modeling guidelines for commercial building energy analysis . PNNL-18898, Pacific Northwest National Laboratory, Richland, WA.
http://www.pnl.gov/main/publications/external/technical_reports/PNNL-18898.pdf

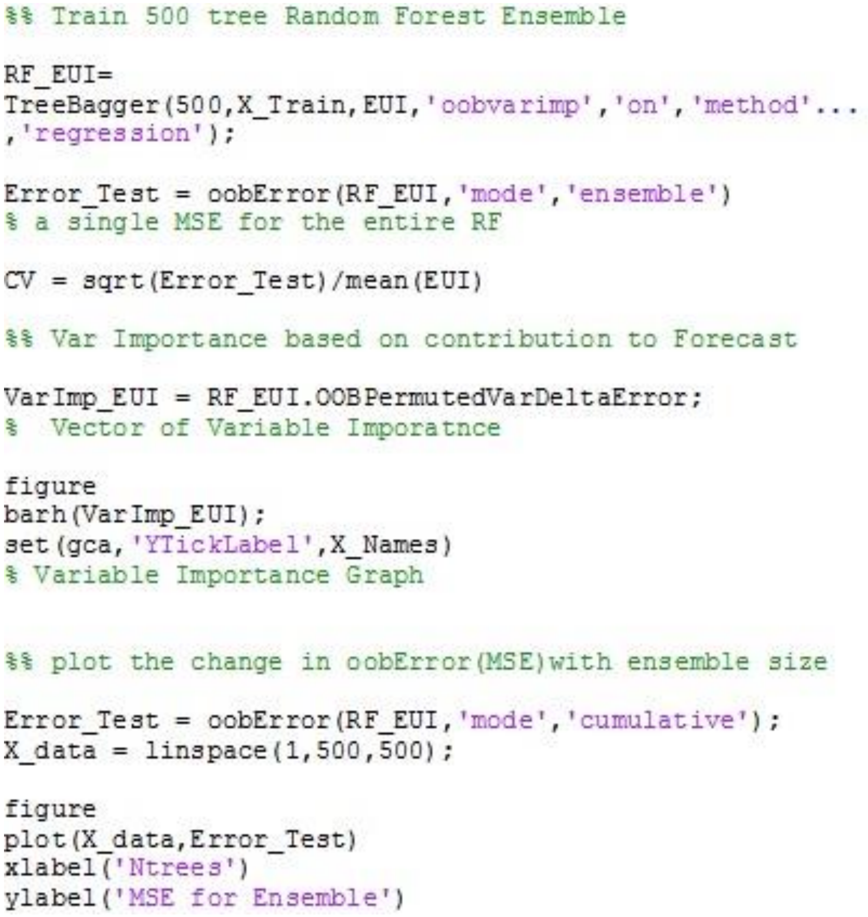
Source: Pacific Northwest National Laboratory, updated on 04-30-2011

APPENDIX C

MATLAB FUNCTIONS FOR RANDOM FOREST GENERATION

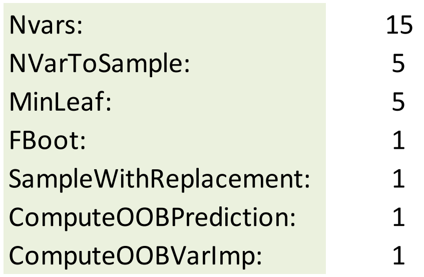
Matlab Class: TreeBagger

Sample Code



```
%% Train 500 tree Random Forest Ensemble

RF_EUI=
TreeBagger(500,X_Train,EUI,'oobvarimp','on','method'...
,'regression');

Error_Test = oobError(RF_EUI,'mode','ensemble')
% a single MSE for the entire RF

CV = sqrt(Error_Test)/mean(EUI)

%% Var Importance based on contribution to Forecast

VarImp_EUI = RF_EUI.OOBPermutedVarDeltaError;
%  Vector of Variable Imporatnce

figure
barh(VarImp_EUI);
set(gca,'YTickLabel',X_Names)
% Variable Importance Graph


%% plot the change in oobError(MSE)with ensemble size

Error_Test = oobError(RF_EUI,'mode','cumulative');
X_data = linspace(1,500,500);

figure
plot(X_data,Error_Test)
xlabel('Ntrees')
ylabel('MSE for Ensemble')
```

Tree Bagger Default Settings



Nvars:	15
NVarToSample:	5
MinLeaf:	5
FBoot:	1
SampleWithReplacement:	1
ComputeOOBPrediction:	1
ComputeOOBVarImp:	1

Reference: http://www.mathworks.com/help/stats/treebagger.html