Gaussian-Process-Based Emulators for Building Performance Simulation

Worker[1], Supervisor[2], Boss[1]
[1]Some University, Some Town, Some Country
[2]Another Institution, Some City, Some Country
worker@fake.email.com

Abstract

In this paper we present a novel emulator of a building simulator for the simulation-assisted design of high performance buildings. Our emulator is based on Gaussian-Process (GP) regression models. Such non-linear models are better suited than linear models as emulators because the simulator itself is a collection of non-linear models based on differential equations. We show that our proposed emulator is about 3 times more accurate than linear models in predicting the output of the simulator, achieving an average error of around 10-25 kWh/m^2 for prediction of energy outputs that are in the range of 10-800 kWh/m^2, compared to an error of around 50-100 kWh/m^2 obtained by using linear models. Our emulators also heavily reduce the computational burden for building designers who rely on simulators. For example, the emulator can first be trained with observations from the simulator using a wide variety of building designs and weather data. This pre-trained model can then be used by building designers for exploration of new designs by predicting the performance of new buildings very quickly (in just a few milliseconds). We expect our approach to be particularly useful for Uncertainty Analysis (UA), Sensitivity Analysis (SA), robust design, and optimisation.

Introduction

This paper presents unconventional emulators based on Gaussian processes (GPs) as a means of enhancing the simulation-assisted design of high performance buildings. The use of emulators to represent experimentally-observed relationships and complex or computationally expensive simulations has been proposed previously for Building Performance Simulation (BPS) (e.g., Hygh et al., 2012; Amiri, Mottahedi, and Asadi, 2015; Eisenhower et al., 2012). These proposals, however, have remained theoretical or academic. To the best of our knowledge, none of the prominent building simulation engines (e.g., ESP-r, EnergyPlus) or Graphical User Interface (GUI)-based programs (e.g., DesignBuilder, Sefaira) offers a regression model as a supplement/replacement for the *main* simulator. Experimentally-derived regression relations are used in sub-components of the simulators (e.g., Percentage People Dissatisfied (PPD) models), but the goal is not to obtain final outputs (e.g., indoor temperature) from 'raw' inputs (e.g., wall construction, building layout).

A typical BPS run (i.e., a building design with one set of weather data and operating conditions) may take from a few minutes for very simple models, to hours for reasonably detailed ones. In addition, the accuracy of the 'predicted' energy use of a design depends on the accuracy of the inputs, and whether the parameters of the simulator have been set appropriately. Even if all fixed inputs (like building properties) are known with perfect accuracy, and the tuning parameters of a simulator are set to some ideal values, the presence of random inputs like weather and human factors means that the simulator does not deliver satisfactory results. A standard approach is to use Monte Carlo (MC) analyses, but this usually requires a large number of BPS runs, significantly increasing the computation time. For such computational-intesive tasks, emulators can be much faster while being reasonably accurate.

In this paper, we show that the non-linear GP-based emulators is about 3 times more accurate than linear models in predicting the output of the simulator, achieving an average error of around 10-25 kWh/m^2 for prediction of energy outputs that are in the range of 10 to 800 kWh/m^2, compared to an error of around 50-100 kWh/m^2 obtained by using linear models. This is expected since the simulator itself is a collection of non-linear models based on differential equations for which non-linear models are better suited than linear ones.

Our emulators also heavily reduce the computational burden for building designers who rely on simulators. For example, the emulator can first be trained with observations from the simulator using a wide variety of building designs and weather data. This pre-trained model can then be used by building designers for exploration of new designs by predicting the performance of new buildings very quickly (in just a few milliseconds). The emulators respond almost instantaneously to a query, regardless of the complexity of the building design being probed.

Our work, and that described in the literature, both use BPS as the 'ground truth', which means that the emulators (regression models) were trained to faithfully reproduce the responses of the simulator. We do not, as yet, have well-developed case studies with measured data to compare the performance of emulators against simulators. While it is clear that neither simulators nor emulators can model reality *exactly*, emulators can be retrained when better data becomes available while simulators cannot. In this sense, emulators are both fast and flexible. The practice of calibrating a simulator using real data, which is a manual process whose conduct depends on the judgement of the simulator user, is not comparable to retraining a regression model.

Related Work

As we argued in the introduction, emulators are most useful in applications that rely on distributions of outputs obtained from hundreds of simulations using MC, e.g., UA, SA, and parametric design exploration (Rastogi, 2016; Hopfe, 2009; de Wit, 2001; Nault et al., 2015; Nault, 2016). Despite the fact that linear models are not suitable for modeling an inherently non-liear simulator, the literature is dominated by linear models, primarily because linear models are easier to train and interpret. In many non-linear techniques, particularly those used in machine learning, e.g., Artificial Neural Networks (ANNs), Gaussian processes, the relations between inputs and outputs are usually not clear. Therefore, the performance of the non-linear methods must be justified by substantially better performance. Recently, non-linear models have been proposed for predicting building energy use and other thermal quantities, e.g., ANNs, Support Vector Machines (SVMs) (Kalogirou, 2006; Zhao and Magoulès, 2012), and Gaussian process regression (Rastogi, 2016). Some recent work has also explored the use of GP regression for optimisation (Wood, Eames, and Challenor, 2015), optimal glazing design (Kim et al., 2013), and operational control (Yan et al., 2013).

Simulators and Emulators

Given a set of building-design parameters and weather data[1], the simulator outputs a quantity of interest (e.g., energy used for space heating/cooling). In this paper, we are interested in predicting annual energy-consumption which can be obtained by integrating the output of the simulator over time. We denote this quantity by a scalar y which takes non-negative values. The vector of inputs, denoted by $\boldsymbol{x}$, includes the building properties, environmental conditions, and human interactions, as well as (free) tuning parameters in the components of a simulator.

The simulator can be represented by a non-linear function f_s that outputs y given an input vector $\boldsymbol{x}$:

$$y = f_s(\boldsymbol{x}). \tag{1}$$

We wish to design an emulator that can predict the value of y as accurately as possible:

$$\hat{y} = f_e(\boldsymbol{x}), \tag{2}$$

where $f_e(\cdot)$ is the emulator. To achieve this, we can choose a function f_e in the set of functions $\mathcal{F}$ that minimises a cost function, e.g., Mean Square Error (MSE),

$$f_e^* = \arg \min_{f_e \in \mathcal{F}} \ \mathbb{E}[y - f_e(\boldsymbol{x})]^2, \tag{3}$$

where the expectation is taken with respect to the distribution $p(y, \boldsymbol{x})$ of the input-output pairs that occur in practice.

Unfortunately, p is unknown, but we can approximate the above expectation by collecting observations generated from p. For example, a building-design expert can collect N inputs $\boldsymbol{x}_n$ for $n = \{1, 2, \ldots, N\}$, where $\boldsymbol{x}_n$ denotes an instance of $\boldsymbol{x}$. She can do this in practice by collecting many basic building designs and weather data from a variety of locations. The outputs y_n can then be obtained by running a simulation on inputs $\boldsymbol{x}_n$. Thus, the cost in Equation (3) may be estimated by using a sample approximation over these observations (sample mean of squared errors).

We use the standard training-testing framework, which is a common practice in statistics and machine learning (Hastie, Tibshirani, and Friedman, 2009). Under this framework, we split the N observations into two mutually-exclusive sets. We use the first set of observations to estimate f_e^*:

$$\hat{f}_e^* = \arg \min_{f_e \in \mathcal{F}} \ \frac{1}{N_{train}} \sum_{n=1}^{N_{train}} [y_n - f_e(\boldsymbol{x}_n)]^2, \tag{4}$$

where N_{train} denotes the number of observations in the first set (the *training* set). The second set is used to assess the *goodness-of-fit* of the estimator, e.g., by computing the cost,

$$\mathcal{L}(\hat{f}_e^*) = \frac{1}{N_{test}} \sum_{n=1}^{N_{test}} [y_n - \hat{f}_e^*(\boldsymbol{x}_n)]^2, \tag{5}$$

where N_{test} denotes the number of observations in the second set (the *test* set). By construction, $N = N_{train} + N_{test}$.

This training-testing framework is the backbone of statistical machine-learning. The cost obtained on an *unseen* test observation-set provides a faithful measure of the emulator's performance in the real world. The degree of faithfulness depends on the size and quality of the training and testing data sets. A large amount of training data (N_{train}) would imply a better estimate of $\hat{f}_e^*$, i.e., close to the optimal f_e^*, while

[1]Human factors and simulator parameters were not varied in our study.

a large amount of testing data (N_{test}) would give us a good estimate of the emulator's performance in the real world. The quality of the data set is considered to be good when the data are independent and identically distributed, i.e., representative of $p(\boldsymbol{x}, y)$. Since the true distribution is unknown, we rely on experts to generate this data by running BPS on realistic building designs and weather conditions.

Model-Based Emulators

For a linear model, we have $f_e = \boldsymbol{\beta}^T \boldsymbol{x}$ where $\boldsymbol{\beta}$ is a real-valued vector of the same size as $\boldsymbol{x}$. The set of functions $\mathcal{F}$ is the set of all linear functions. The solution obtained by minimising Equation (4) is usually called the Ordinary Least-Squares (OLS) solution. In practice, $\boldsymbol{x}$ might be collinear, e.g., when two entries in $\boldsymbol{x}$ represent the same underlying variable. This gives rise to ill-conditioning and might make $\boldsymbol{\beta}$ explode to infinity. In such situations, it helps to impose a distribution over $\boldsymbol{\beta}$. In this work, we employ a Gaussian distribution: $\boldsymbol{\beta} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma}$ is a covariance matrix.

We propose to use a non-linear model obtained by non-linear transformation of the inputs:

$$f_e(\boldsymbol{x}) = \boldsymbol{\beta}^T \boldsymbol{\phi}(\boldsymbol{x}), \tag{6}$$

where $\boldsymbol{\phi}(\boldsymbol{x})$ is an M-length vector containing various non-linear transformation of the inputs. That is, $\boldsymbol{\phi}(\boldsymbol{x}) = [\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}), \ldots, \phi_M(\boldsymbol{x})]^T$ where each ϕ_i is a different non-linear function. This is referred to as the basis-function model in machine learning. As an example, the polynomial basis-function for a scalar x defines $\phi_i(x) = x^{i-1}$, therefore $\boldsymbol{\phi}(x) = [1, x, x^2, \ldots, x^{M-1}]^T$. A linear model can be obtained as a special case by setting $\boldsymbol{\phi}(\boldsymbol{x}) = \boldsymbol{x}$.

Gaussian Process and Kernels

We use the Gaussian process regression framework to estimate the function f_e that minimises Equation (4). Gaussian process regression uses Bayes' rule to compute the posterior distribution over f_e given outputs y_n (Rasmussen and Williams, 2006, ch. 2). This approach works directly in the space of f_e and avoids both a direct estimation of $\boldsymbol{\beta}$ and a direct specification of $\boldsymbol{\phi}(\boldsymbol{x})$. Instead, a 'kernel' function specifies the inner products of $\boldsymbol{\phi}$ as:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{\phi}(\boldsymbol{x}_i)^T \, \boldsymbol{\Sigma} \, \boldsymbol{\phi}(\boldsymbol{x}_j), \tag{7}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two inputs in our observation set. In practice, a kernel function is easier to specify than $\boldsymbol{\phi}$, even though it could be unintuitive.

A variety of models can be obtained this way, e.g., a linear model is obtained by using the linear kernel:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{\Sigma} \boldsymbol{x}_j. \tag{8}$$

In this paper, we will compare the linear model to the following nonlinear model which is obtained by using a Squared Exponential (SqE) kernel function:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \, exp\Big(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \, \boldsymbol{\Sigma} \, (\boldsymbol{x}_i - \boldsymbol{x}_j)\Big), \tag{9}$$

where $\sigma_f^2 > 0$ is the signal variance. This kernel is also referred to as the Radial Basis Function (RBF) kernel in the context of Artificial Neural Network (ANN).

Input-Variable Selection

Nonlinear models are powerful and flexible, which is good when we have a large data set of good quality, but otherwise nonlinear models might 'over-fit', i.e., they might fit to the noise instead of the signal. To avoid this, one solution is to reduce the complexity of the model (number of input or explanatory variables). In this paper, we use an automatic variable selection method called Automatic Relevance Determination (ARD) (Rasmussen and Williams, 2006, sec. 5.1). In this method, we set $\boldsymbol{\Sigma}$ to be a diagonal matrix with each diagonal entry $\Sigma_{ii} = 1/l_i$ where $l_i > 0$, and then estimate $\boldsymbol{l} = [l_1, l_2, \ldots, l_M]$. As $l_i \to \infty$ (or $1/l_i \to 0$), the importance of the corresponding input dimension x_i goes to 0. We use the log-marginal likelihood of the GP regression model to estimate $\boldsymbol{l}$ and other parameters of the kernel (Rasmussen and Williams, 2006, Eq. 2.30). We call this type of models the ARD models.

We compare these models to those in which there is no variable selection, i.e., we set all l_i to one value l. We call these models isotropic (ISO) models since this choice makes the Gaussian distribution on $\boldsymbol{\beta}$ an isotropic Gaussian distribution. We expect that ARD is useful for non-linear models while for linear models it may not matter much since the model is simple and in less danger of over-fitting. We expect all models to perform better than the 'Mean' model since this model is the simplest. We expect nonlinear models to perform better than linear models when a sufficient amount of data is available.

The Automatic Relevance Determination (ARD) procedure may or may not establish the practical relevance of the input variables, rather it determines their relevance in predicting the output variables. This procedure is also affected by the estimation procedure and the amount of data, e.g. the minimiser, might gets 'stuck' in a local minimum due to poor initialisation. This is also more likely when we do not have enough data to estimate the marginal likelihood.

Data Description (Case Studies)

The case studies are based on abstract representations of typical buildings, taken from the United States Department of Energy (USDOE) Commercial Buildings Reference Database (Deru et al., 2011) (referred to as the 'USDOE' series after this). Details about how the data were produced can be found

Table 1: *List of models compared in this study.*

Model	Description
Mean	Mean of the outputs y_n
Lin-ISO	Linear model with ISO
Lin-ARD	Linear model with ARD
NonLin-ISO	SE Kernel with ISO
NonLin-ARD	SE Kernel with ARD

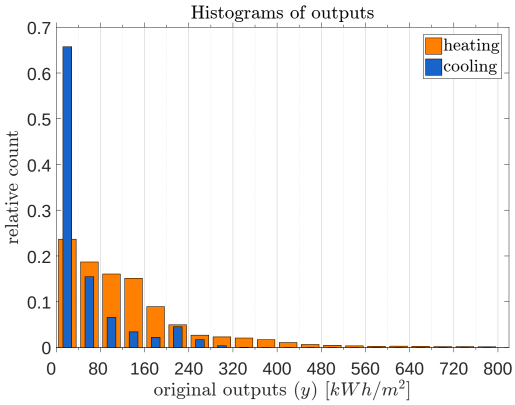

Figure 1: Distribution of the outputs $y_n^{heating}$ for heating loads and $y_n^{cooling}$ for cooling loads. The weight of the distributions is skewed towards lower values because we picked, unintentionally, more moderate climates than extreme ones.

in Rastogi (2016, sec. 4.2 and B.4). The buildings were modelled without Heating, Ventilation, and Air Conditioning (HVAC) systems, in EnergyPlus v8.5 (NREL and USDOE, 2015). The weather data used are described in Rastogi (2016, sec. A.5 and B.4). Seventeen different building types, distinguished by usage, were simulated (e.g., hospitals, apartments). In addition to a different usage profile or programme, each type had a different layout and arrangement of rooms. Further variety was introduced by using three wall construction profiles for each building type.

Using the USDOE database and weather data from several climates (regions) worldwide, we obtain a total of $N = 88242$ input-output pairs. Each input $\boldsymbol{x}_n$ contains 28 dimensions, i.e., 28 input variables, which are described in Table 2. We used two outputs, $y_n^{heating}$ and $y_n^{cooling}$, the ideal heating and cooling energy consumption respectively. Both these outputs were obtained by summing the respective hourly loads over one year to give the ideal heating or cooling energy consumption. The distribution of these outputs is shown in Figure 1. We fit two separate regression models for $y_n^{heating}$ and $y_n^{cooling}$, although it is also possible to use one model to predict both the outputs.

We standardise all inputs and outputs, i.e., we compute their means and standard deviations, and first subtract the mean and then divide by the standard deviation. This way all the inputs and outputs lie within the same range, which improves the robustness of the numerical procedures used during estimation.

Results

We now compare the models shown in Table 1 using the training-testing framework discussed earlier. In our first experiment, we investigate the effect of the amount of training data on prediction quality. We expect all models to perform badly when the amount of training data is limited. As the amount of data is increased, we expect nonlinear models to perform better than linear models.

To estimate the real-world performance of a model, we set aside about 45% of the 88,242 observations (simulations) for testing ($N_{test} = 39,776$) and use parts of the remaining data set for training the model. Note that the test set is always 'left out', i.e., a model never uses the data in the test set for training and is completely unaware of it. The test data is fairly large and therefore we expect it to produce a faithful estimate of the real-world prediction error of the model. We present the Root Mean Square Errors (RMSEs) on the test set, which are obtained by taking the square root of Equation (4).

The training data set size N_{train} varies from 50 to 4000. For a given training size N_{train}, we draw that many training observations from the large overall training set and use it to train a model. We then predict the left-out test data using the trained model and compute RMSE. We repeat this process 100 times to obtain an empirical distribution of the RMSE estimate. This gives us a confidence estimate which can be used in a significance test.

Figure 2 shows the results for heating loads in the left column and cooling loads in the right column. Figures 2a and 2b show the evolution of RMSE as a function of the size of training data for heating and cooling loads respectively. Each curve in the plots shows the performance of a model from Table 1. The thick line shows the median of the RMSE distribution while the shaded area around the thick line shows the 25th and 75th percentiles. As N_{train} increases, all models perform much better than the 'Mean' model, which only uses the mean of the output from the training data. The linear models, for example, are about 25% better, while the non-linear models are 50-75% better (depending on the type of nonlinear model). This comparison is similar to how the R^2 is calculated for a fit to data. If a model performs worse than a mean of the training data, i.e., a horizontal line fit, then it should be discarded.

We now present a detailed analysis of the prediction errors obtained in the subsequent plots. Figure 2e shows the empirical distributions of (the absolute value of) prediction errors for two different data size, $N_{train} = 100$ and $4,000$ respectively. The errors obtained by using the larger data size are much smaller than those obtained by using a smaller data size, as expected. With more training data, we are consistently able to reduce the larger values of error. Figure 2c show a similar comparison between linear and nonlinear models. The nonlinear model too decreases the number of larger mistakes. Similar comparisons for the cooling loads are shown in Figures 2d and 2f.

Overall, the performance of the linear models is uniformly worse than that of the non-linear models. As expected, when the amount of data is small, the performance of the linear and non-linear models is comparable. However, the linear model errors plateau very quickly and the performance does not improve as the amount of data is increased. This clearly shows that the linear models are not adequate for modelling the complex nonlinear data, and are unable to use the information present in the data. On the other hand, the non-linear models continue to improve as the amount of data is increased. Between the two nonlinear models, the model with ARD performs significantly better. This shows that reducing the number of features improves the performance. This is also expected since a nonlinear model with too many features might overfit, and reducing the feature dimensionality reduces this problem.

We now discuss the relevance of features found by using the ARD method on the GP regression model with a nonlinear kernel (SqE). Recall that as $1/l_i \to 0$, the relevance of the feature x_i reduces (compared to other features and assuming that the range of features are roughly in the same order). As discussed earlier, the selection of features is affected by the estimation procedure and the amount of data. We present results for $N_{train} = 4,000$ since the model performance seems to plateau around this training data size (Figure 2a). To clearly see the artifact introduced by the estimation procedure, we plot the empirical distribution of $1/l_i$ obtained by using the 100 runs for $N_{train} = 4,000$. For irrelevant variables, we expect $1/l_i$ to be nearly zero most of the time.

Figure 3 containes 28 plots, each of which show the distribution of $1/l_i$ for a feature. The name of the feature is shown in the title and its description is available Table 2. The distribution is plotted over 100 values and is normalised (scaled to 0-1). Both x and y axis are limited to 0-1 for easy visualisation. Inputs which take non-zero values most of the time are marked with thicker boxes around the plot. We now discuss the consequences of these results.

Only two inputs, Window-to-Wall Ratio and Window-to-Floor Ratio, are consistently significant, for both models. The Internal Heat Gain variable (*sumihg*) is significant in the heating model but missing in the cooling model. Also in the heating model, the influence of mean sunlit percentage ($MsunP$) is small, though consistent, while that of the volume-to-wall-area ratio (form-factor or *ff*) is occasionally moderate. The median and Inter-Quartile Range (IQR) of Dry Bulb Temperature (TDB) (*medtdb* and *iqrtdb*) are weakly significant for cooling models, though not for heating. While the significance of Window-to-Wall Ratio (WWR) and Window-to-Floor Ratio (WFR) is not surprising, the presence of *both* in a single fit is unexpected, given their high correlation (Rastogi, 2016, fig. 4.3). The dominance of these variables could be because the buildings we chose are more driven by envelope (fenestration) loads. The IQR of TDB in a year tends to be moderately correlated with its median (Rastogi, 2016, fig. 4.3). We guess that the solar parameters are missing because their effect is well accounted for by the envelope ratios: WWR and WFR. Overall, the dominance of building parameters (WWR, WFR, and Internal Heat Gain) is not surprising in what is largely a sample from commercial buildings with medium to high fenestration ratios. The almost complete rejection of all climate parameters *is* surprising, however, and we expect may be different in a data set dominated by residential buildings.

Discussion & Conclusion

In this paper, we presented non-linear and linear Gaussian process regression models for emulating Building Performance Simulation (BPS). We showed that nonlinear models achieve RMSE values that are 3-4 times lower than those of linear models. Finally, we discussed the results from using fits in which the input variables are selected using the Automatic Relevance Determination (ARD) method.

Gaussian process regression offers a promising and flexible approach to creating emulators for building performance simulation. We have built upon previous work (both ours and from the literature), to present alternatives to classical linear regression in the form of Gaussian process regression. The proposed regression models, though more expensive to train, give more accurate predictions. We expect to further refine these and other regression methods in future studies for use in emulating BPS. The development of easy-to-use regression modules for BPS programs, which help a user to probe and customise a regression model for their design problem, is ongoing. We also plan to test Gaussian process regression and other machine learning approaches, like ANN and random forests, on larger data sets with many more features.

Building simulation is essentially a solution of several interconnected partial and ordinary differential equations, feeding human preference models and being fed by empirical models for individual components, so it cannot reasonably be expected to conform to a linear regime. We do not support the idea of large databases of pre-simulated cases being emulated with linear regressors, where a new design problem may be 'located' and its energy use estimated. We argue, instead, that the benefits and performance of a non-

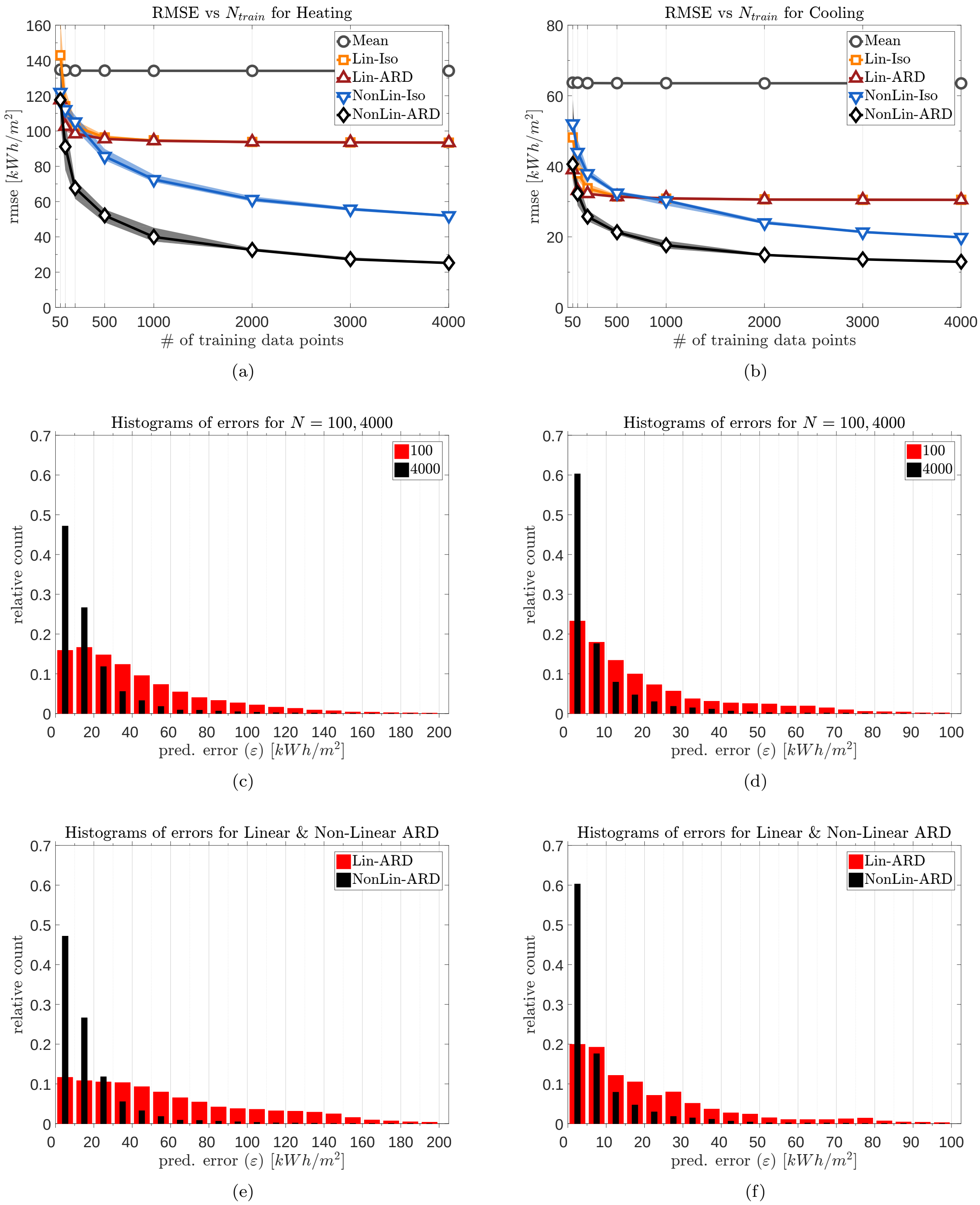

Figure 2: Plots on the left show results for the heating loads and on the right show results for the cooling loads. Figures a and b show the Root Mean Square Error (RMSE), plotted against the size of the training data set. The lines indicate median errors, and the filled areas are bounded by the 25th and 75th percentiles. The non-linear models outperform linear models, and the predictions improve with size of training data set. Figures c and d compare the histograms of absolute errors, $|\varepsilon| = |y_j - f_e(\boldsymbol{x}_j)|$, for $N_{train} = 100$ and $4,000$. We see that a large training data size reduces larger values of error. Figures e and f compare the same for the linear and non-linear models, where we see that the nonlinear model makes fewer errors.

linear emulator outweigh the costs of training it on anything from a few dozen to a few hundred simulations (depending on the complexity of the simulation model and design exercise, usually indicated by the number of building and environmental parameters used as input variables). The regression model data and parameters do need to be stored (perhaps on a server), but the response from a query would be instantaneous during a design exploration or MC exercise.

The use of emulators greatly eases the process of quantifying uncertainty and sensitivity in building simulation, an argument we have previously summarised in Rastogi (2016). The effort needed to simulate enough data for training may be lessened by using a combination of a large variety of automatically-sampled random inputs, e.g., weather, and a manageable number of manually-created 'design variants', e.g., 5-6 combinations of building properties. This is how the data used in this study was generated. In future work, we expect to increase the features used in the model to account for more aspect of the building. The emulators may also be extended to deliver time series of temperatures, energy, etc., which may be more informative in making decisions. These emulators are envisioned as part of a larger project to use emulators for machine-assisted design exploration.

Acknowledgement

The authors would like to thank CB for his patient and valuable support, and AD for his incisive and constructive criticism. PR was funded by ABCD and EFGH during his work at Some University. This work was partly carried out at the MS group in XYZ Institute.

Glossary

$\boldsymbol{\Psi}$ super set of all inputs into the building simulator, consisting of building design options ($\boldsymbol{B}$), weather/climate conditions ($\boldsymbol{W}$), and human factors ($\boldsymbol{H}$). Eventually, the set should also include simulator conditions or tuning parameters ($\boldsymbol{S}$).

ANN Artificial Neural Network

ARD Automatic Relevance Determination

BPS Building Performance Simulation

building envelope Used interchangeably with 'facade', it refers to those elements of a building that form an interface between a building and its environment. This usually is taken to include the walls, roof, and floor. In our analyses, we often ignore the floor, not including it, for example, in the factorial experiment of U-values.

facade Spelt as both facade and façade, this term is used interchangeably with the word building envelope in this work.

Gaussian process regression Gaussian process regression is a supervised kernel-based machine-learning method. In this technique, the "data has to do the talking" (Ebden, 2008), though it is not completely free-form. "Formally, a Gaussian process generates data located throughout some domain such that any finite subset of the range follows a multivariate Gaussian distribution" (Ebden, 2008). Rasmussen and Williams (2006) define a Gaussian process as "a collection of random variables, any finite number of which have a joint Gaussian distribution".

GP Gaussian process

GUI Graphical User Interface

HVAC Heating, Ventilation, and Air Conditioning

Internal Heat Gain Defined in this work as the sum of (usually sensible) heat gains from people, equipment, and lights. Technically, internal heat gains should include latent gains, but we do not consider them in the context of this work.

IQR Inter-Quartile Range

MC Monte Carlo

MSE Mean Square Error

OLS Ordinary Least-Squares

PPD Percentage People Dissatisfied

RBF Radial Basis Function

RMSE Root Mean Square Error

SA Sensitivity Analysis

SQ Sensitivity Quantification

SqE Squared Exponential

SVM Support Vector Machine

TDB Dry Bulb Temperature

UA Uncertainty Analysis

UQ Uncertainty Quantification

USDOE United States Department of Energy

WFR Window-to-Floor Ratio

Window-to-Floor Ratio The ratio between the area of windows (transparent elements) and the area of the floor or footprint of the building.

Window-to-Wall Ratio The ratio between the area of windows (transparent elements) and the area of the wall.

WWR Window-to-Wall Ratio

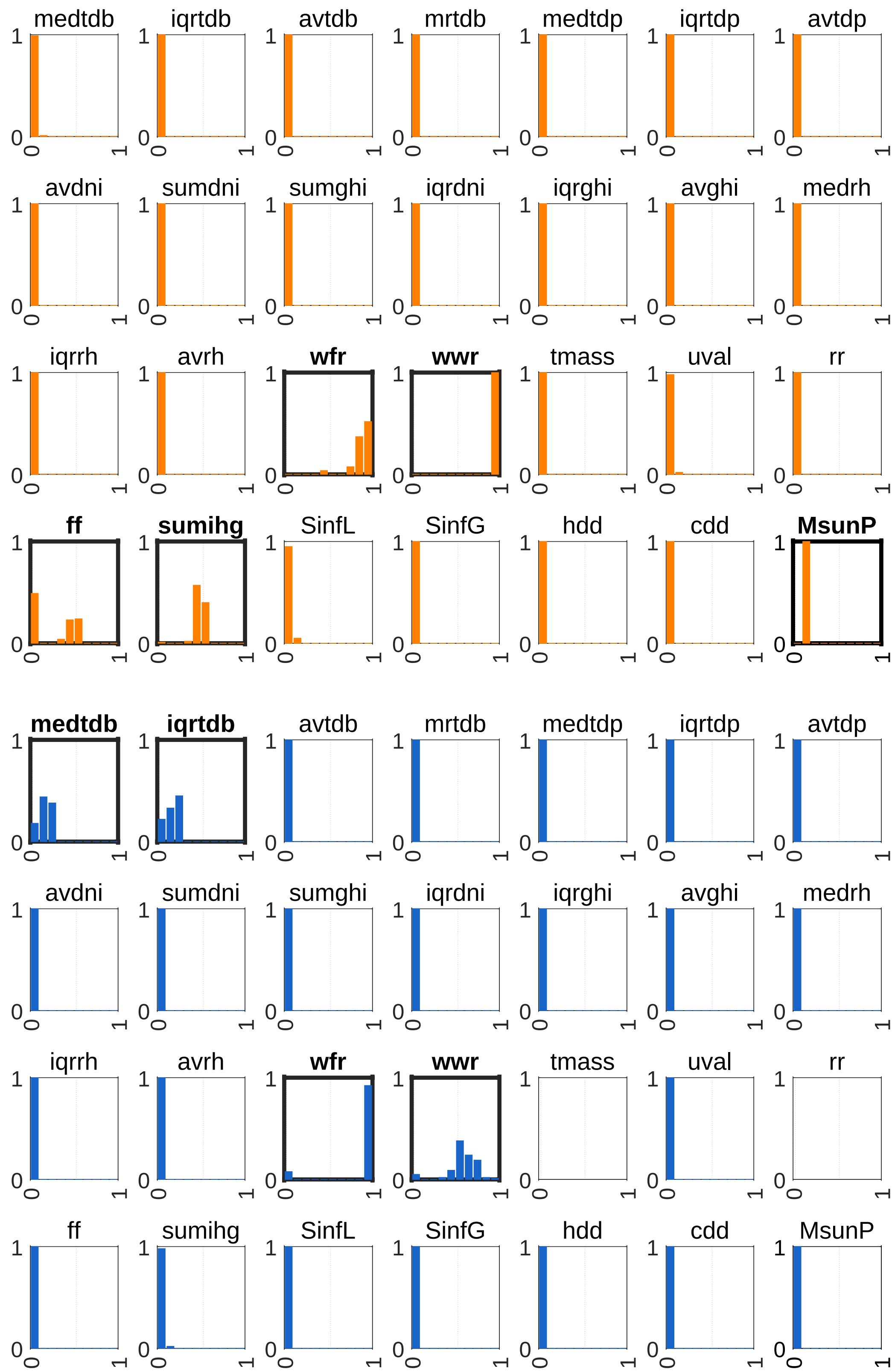

Figure 3: Histograms of the relevance parameters $1/l_i$ for all 28 input variables. The plots at the top (in red) show results for the heating load, while at the bottom (in blue) show results for the cooling loads. Each plot contain the histogram of $1/l_i$ for an input whose name is shown in the title of that plot (see Table 2 for an explanation of the variable codes). The histogram is obtained using the 100 runs over training-data subsets of size $N_{train} = 4,000$. A value of $1/l_i$ close to zero indicates that the input was not involved in determining the prediction accuracy of the model. The thick boxes indicate the input for which $1/l_i$ was non-zero 20% of the time.

Table 2: Initial inputs, codes, and units.

Group	Quantity	Statistic	Name	Code	Units
Building	U-value	Average	Average U-value of envelope	*uval*	W/m^2K
	Thermal Mass	Sum	Sum of thermal storage capacity	*tmass*	MWh/K
	Envelope Ratios	Ratio	Ratio of window area to wall area	*WWR*	—
			Ratio of window area to floor area	*WFR*	
	Massing	Ratio	Form Factor (Volume / Wall Area)	*ff*	—
			Roof Ratio (Roof / Wall Area)	*rr*	
Mixed	Shading	Average	Average sunlit percentage of envelope	*MsunP*	%
	Infiltration	Sum	Annual sum of energy gained from infiltration	*SinfG*	GWh
			Annual sum of energy lost to infiltration	*SinfL*	
	Internal Heat Gain		Annual sum of Internal Heat Gain	*sumIHG*	GWh
Climate	Degree Days	Sum	Annual sum of cooling degree days	*cdd*	°C-day
			Annual sum of heating degree days	*hdd*	
	Dry Bulb Temperature (Hourly)	Average	Annual average of dry bulb temperature	*avgtdb*	°C
		Median	Median dry bulb temperature	*medtdb*	
		IQR	Inter-quartile range of dry bulb temperature	*iqrtdb*	
	Dew Point Temperature (Hourly)	Average	Annual average of dew point temperature	*avgtdb*	°C
		Median	Median dew point temperature	*medtdb*	
		IQR	Inter-quartile range of dew point temperature	*iqrtdb*	
	Global Horizontal Irradiation (Hourly)	Average	Annual average of global horizontal irradiation	*avgghi*	MWh/m^2
		Sum	Annual sum of global horizontal irradiation	*sumghi*	
		IQR	Inter-quartile range of global horizontal irradiation	*iqrghi*	
	Direct Normal Irradiation (Hourly)	Average	Annual average of direct normal irradiation	*avgdni*	MWh/m^2
		Sum	Annual sum of direct normal irradiation	*sumdni*	
		IQR	Inter-quartile range of direct normal irradiation	*iqrdni*	
	Humidity (Hourly)	Average	Annual average of relative humidity	*avgrh*	%
		Median	Median relative humidity	*medrh*	

References

Amiri, Shideh Shams, Mohammad Mottahedi, and Somayeh Asadi. 2015. "Using multiple regression analysis to develop energy consumption indicators for commercial buildings in the U.S.." *Energy and Buildings* 109: 209–216. http://www.sciencedirect.com/science/article/pii/S0378778815303133.

de Wit, Sten. 2001. "Uncertainty in predictions of thermal comfort in buildings." PhD. Delft University of Technology. Delft, The Netherlands. http://resolver.tudelft.nl/uuid:a231bca8-ec81-4e22-8b34-4bafc062950e.

Deru, Michael, Kristin Field, Daniel Studer, Kyle Benne, Brent Griffith, Paul Torcellini, Bing Liu, et al. 2011. *U.S. Department of Energy commercial reference building models of the national building stock.* Technical report. National Renewable Energy Laboratory (NREL). http://digitalscholarship.unlv.edu/renew_pubs/44.

Ebden, Mark. 2008. *Gaussian Processes for Regression: A Quick Introduction.* Technical report. University of Oxford. http://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf.

Eisenhower, Bryan, Zheng O'Neill, Satish Narayanan, Vladimir A. Fonoberov, and Igor Mezić. 2012. "A methodology for meta-model based optimization in building energy models." *Energy and Buildings* 47: 292–301. http://www.sciencedirect.com/science/article/pii/S0378778811005962.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition.* Springer.

Hopfe, Christina Johanna. 2009. "Uncertainty and sensitivity analysis in building performance simulation for decision support and design optimization." PhD. Technische Universiteit Eindhoven. Eindhoven, The Netherlands. http://www.bwk.tue.nl/bps/hensen/team/past/Hopfe.pdf.

Hygh, Janelle S, Joseph F. DeCarolis, David B Hill, and S Ranji Ranjithan. 2012. "Multivariate regression as an energy assessment tool in early building design." *Building and Environment* 57: 165–175. http://dx.doi.org/10.1016/j.buildenv.2012.04.021.

Kalogirou, Soteris A. 2006. "Artificial neural networks in energy applications in buildings." *International Journal of Low-Carbon Technologies* 1 (3): 201–216. http://ijlct.oxfordjournals.org/content/1/3/201.short.

Kim, Young-Jin, Ki-Uhn Ahn, CS Park, and In-Han Kim. 2013. "Gaussian emulator for stochastic optimal design of a double glazing system." In *Proceedings of the 13th IBPSA Conference, August,* 25–28.

Nault, Emilie. 2016. "Solar Potential in Early Neighborhood Design. A Decision-Support Workflow Based on Predictive Models." PhD Thesis. Ecole polytechnique fédérale de Lausanne. Lausanne, Switzerland.

Nault, Emilie, Parag Rastogi, Emmanuel Rey, and Marilyne Andersen. 2015. "The sensitivity of predicted energy use to urban geometrical factors in various climates." In *Proceedings of PLEA 2015,* Bologna, Italy. Sep.. http://infoscience.epfl.ch/record/211101?ln=en.

NREL, and USDOE. 2015. "EnergyPlus." https://energyplus.net/weather.

Rasmussen, Carl Edward, and Christopher K. I. Williams. 2006. *Gaussian processes for machine learning.* Adaptive computation and machine learning. Cambridge, Mass: MIT Press.

Rastogi, Parag. 2016. "On the sensitivity of buildings to climate: the interaction of weather and building envelopes in determining future building energy consumption." PhD. Ecole polytechnique fédérale de Lausanne. Lausanne, Switzerland. Doi:10.5075/epfl-thesis-6881. https://infoscience.epfl.ch/record/220971?ln=en.

Wood, M, ME Eames, and Peter Challenor. 2015. "A comparison between Gaussian Process emulation and Genetic Algorithms for optimising energy use of buildings." In *Proceedings of BS 2015,* Hyderabad, India: IBPSA. http://www.ibpsa.org/proceedings/BS2015/p2420.pdf.

Yan, JE, Young-Jin Kim, Ki-Uhn Ahn, and Cheol-Soo Park. 2013. "Gaussian process emulator for optimal operation of a high rise office building." In *Proceedings of BS 2013,* Chambéry, France.

Zhao, Hai-xiang, and Frédéric Magoulès. 2012. "A review on the prediction of building energy consumption." *Renewable and Sustainable Energy Reviews* 16 (6): 3586–3592. http://www.sciencedirect.com/science/article/pii/S1364032112001438.