Kernel Regression for Real-Time Building Energy Analysis

Matthew Brown[1], Chris Barrington-Leigh[2], and Zosia Brown[2]

[1]École Polytechnique Fédérale de Lausanne, Switzerland
[2]University of British Columbia, Vancouver, Canada

Abstract

This paper proposes a new technique for real-time building energy modelling and event detection using kernel regression. We show that this technique can exceed the performance of conventional neural network algorithms, and do so by a large margin when the available training dataset is small. Furthermore, unlike the synapse weights in a neural network, the parameters of our kernel regression models are amenable to human interpretation, and can give useful information about the building being studied. We extensively test our proposed algorithms using a new dataset consisting of 1.5 years of power and environmental measurements for 4 buildings, in addition to benchmarking against the ASHRAE Predictor Shootout dataset. On the new dataset our kernel regression algorithm gave the best prediction performance in 3 of 4 cases, and significantly outperformed neural networks (the nearest competitor) with training sets of 1/2 year or less.

1 Introduction

The goal of reducing energy consumption in buildings on a global scale creates a need for easily deployable, scalable tools that can help to understand the energy use characteristics of large numbers of buildings. To meet this need, recent years have seen rapid growth in the area of Building Dashboards (Berkeley Dashboard[1]; [AWG09]) and Energy Information Systems [New09], as both academic and commercial projects. See [GPGP09] for a survey of state-of-the-art tools, and case studies of their use. In the simplest case, an Energy Information System (EIS) provides feedback of energy usage to occupants and managers, and this feedback alone has been shown to yield benefits in terms of reduced energy consumption in commissioning [MM09] and in use [PSJ+07, Dar06]. However, the ability to also *predict* energy usage forward has potentially profound implications for intelligent building operation and control [GAB94, Mah01, KDM96, HDK98], for example, by allowing buildings to actively regulate their consumption in a smart grid scenario.

The building design community has already established advanced tools for building modelling and simulation, such as EnergyPlus [CLP+04, CLPW00]. Although such tools could in principle be adapted to prediction tasks, the complexity of these models and requirement of actual physical building data makes tuning their parameters to meet observed building performance difficult [HBS98]. Hence, it is attractive to look for simpler models that may not have such a strong physical basis, yet that perform well at prediction. This idea is not new, and was examined in detail for a batch prediction task in the 1993 ASHRAE "Great Energy Predictor Shootout" [KH94]. This competition posed the problem of predicting hourly building energy use from a series of environmental input variables and corresponding building power data over a 4-6 month period. Using this training data,

[1]http://demandless.org

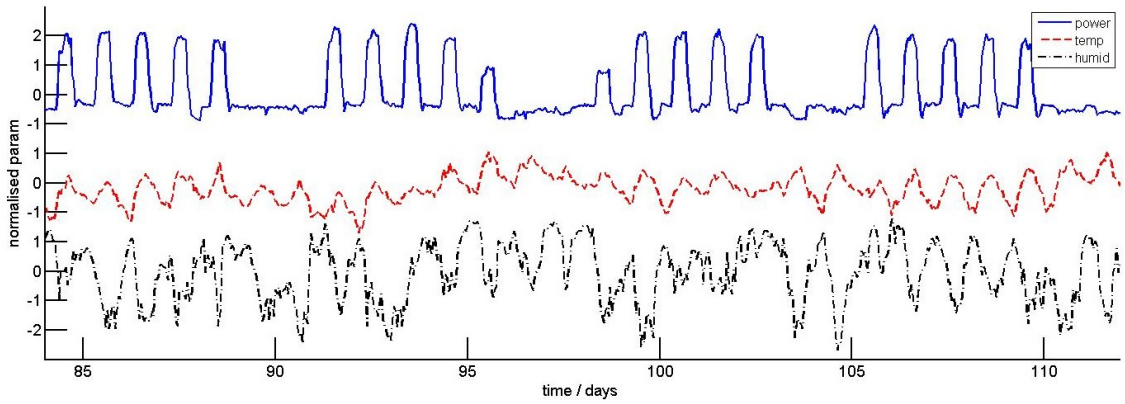

Figure 1: Energy modelling. We will attempt to predict unknown building power usage (top curve) from forecasts of weather parameters, such as humidity and temperature (lower two curves). We will also make use of the temporal structure and periodicities in the demand profile.

competitors were required to generate predictions over a 2 month test period, with no technical details about the buildings given. The competition attracted 150 entrants, who attempted to predict the unseen power loads from weather and solar radiation data using a variety of approaches.

The winner of the competition was an entry from David Mackay [Mac94]. Mackay's algorithm was based on Bayesian modelling using neural networks, with an "Automatic Relevance Determination" (ARD) prior to help select the relevant variables from the large number of possible inputs. Although this algorithm won the competition by some margin, a large fraction of the other highly ranked algorithms were also based on some form of neural network[2].

A second energy prediction competition was run the following year posing a similar mathematical problem, but applied to buildings that had undergone a retrofit to increase efficiency [KH96]. For this competition the entrants had to build models for the same buildings before and after the retrofits, with the intention of accurately estimating the energy savings that resulted. This led to a string of improved algorithms, again mostly neural network based. The winner used a variation of Wald's test to establish the relevance of the input variables [DH96].

The purpose of the Predictor Shootouts was to provide the building analysis community with a set of robust methods for predicting hourly energy use, which would have applications in diagnostics as well as building energy retrofit savings calculations. At the time, the Rio Earth Summit had just passed Agenda 21 – a blueprint for achieving global sustainability, the US Green Building Council was founded (1993), and across the building industry there was growing interest in tackling the energy efficiency problem.

Contemporary work has continued to improve upon the Energy Predictor Shootout notion of using inverse modelling to predict energy demand. Huang and Shih [HS03] extend the classic autoregressive model to deal with non-Gaussian error statistics for short term load forecasting. Dhar et al [DRC99] acknowledge the long term correlations in demand profiles, and explicitly formulate a frequency domain model to take advantage of periodicities in the data. Yang et al [YRZ05] consider the problem of on-line modelling and prediction, extending the popular neural network approach to deal with a continuously evolving training dataset.

An approach similar in spirit to our own is the work of Dong et al [DCL05]. Here the authors explore the use of Support Vector Machines (SVMs) for prediction of monthly energy usage based on temperature, humidity and solar radiation. In common with our kernel regression tech-

[2]Others entrants included local linear approximations and multi-layer perceptrons, in addition to several basic modelling approaches like linear and piecewise linear regression.

nique, SVM regression is based on kernel distances over all historical training data. Dong et al also optimise selected parameters using cross validation (as we will discuss in Section 4.3). However, their technique lacks a continuous optimisation framework for kernel bandwidths, and cannot generalise to large training sets, both of which we have found to be important when predicting hourly building energy usage. Neural networks remain very popular, although the range of reported results suggests that careful engineering and statistical checks may be required for them to function effectively [KSG06, Kal00].

A neural network is clearly not a physically plausible model for a building, so ultimately one would expect that physically based models, with unknown parameters optimised from real observed data, would provide the most accurate predictions of building energy usage. Some examples of this idea in practice include the work of Lee and Braun [LB04] and Andersen et al [AMH00]. These models, sometimes called "grey-box" models, typically involve a simplified physical building model whose parameters are learnt from training data. This is in contrast to "black-box" techniques such as neural nets, where the parametric form may be unrelated to the physical system, and the parameters are often physically meaningless. Andersen et al [AMH00] formulate stochastic differential equations to model building heat dynamics, and learn the unknown parameters using a maximum likelihood approach. Lee and Braun [LB04] formulate an energy model using thermal resistance and capacitance parameters, whose values are also learnt by minimising prediction errors over a training set of real data. Ultimately, we envisage a continuum of possible energy modelling tools [Mah04], making use of a smaller or greater amounts of physical or operational building data as available, with any remaining unknown parameters learnt from real data.

1.1 Scalable Real-Time Energy Modelling

In the current context of global warming, carbon markets and pricing, and governments increasingly mandating building performance evaluation and display, the scenario of ubiquitous building energy monitoring is looking increasingly likely. Initiatives such as the European Energy Performance of Buildings Directive [WvD07], the State of California's Assembly Bill 1103 [Sec07], and the province of British Columbia's smart-grid system [BC 10] not only have profound implications for the metering industry, but will generate large databases of building energy information to be tapped into. Given the scale of such databases, it would be impractical to create detailed physical models of all of the buildings. Thus, there is an increased need for data-driven modelling tools that can predict and optimise energy usage using an minimal amount of information about each building.

In this work we revisit the idea of scalable building energy modelling in the context of a real-time application. This requires estimating the predicted energy usage of a building without a detailed physical specification of it. In such real-time applications, the prediction of building energy use is not always the final outcome, but rather an intermediate step needed for simulation based building control systems [GAB94, Mah01] or to alert a building operator of an unusual condition. For example, we may want to warn the facility manager if the energy usage of the building appears much greater than our model would predict. This requires a statistical decision to be made about the probability of the current building state. Alternatively, we might want to automatically adjust building loads in response to expected high temperatures in the future, for example, by using the electricity during periods of low demand to store ice in a thermal reservoir [KDM96, HDK98], or deferring large computing jobs for off-peak time. These types of scenarios require real-time decision making based on the prediction data.

In this work we propose the use of kernel regression [WJ95] for building energy modelling and event detection. Whilst the parameters of our model are not literal physical parameters as in [LB04], they are quite understandable to humans, and will give useful information about the buildings under study. In this sense our model has more in common with grey-box models such as [LB04] and [AMH00], than the black-box models used in [KH96]. Furthermore, we show that kernel re-

gression can provide more accurate predictions than neural networks, especially when the available training dataset is small.

1.2 Our Contribution

The main contributions of our work are:

- Efficient, data-driven building energy prediction using kernel regression. Our models use a small number of parameters, that can be learned from training data. We exceed the performance of an ensemble of 5 neural networks with identical inputs. We implement our smoother using an efficient k-nearest neighbour algorithm, which allows it to assimilate a very large training dataset.
- Statistical anomaly/event detection. By reasoning about the statistics of our modelling errors, we are able to make probabilistic decisions as to whether a building is experiencing an unusual state, or some event, such as exceeding a power threshold, is likely to occur. This could be used to alert a building operator or inform a control strategy.

1.3 Comparison to Neural Networks

Our method has several favourable properties in comparison to more commonly used neural network techniques:

- Kernel regression requires a smaller number of parameters than a neural network with equivalent inputs, and thus is less susceptible to overfitting. In addition, these parameters are easily interpreted by humans, unlike the synapse weights inside a neural network.
- Kernel regression behaves more smoothly than neural networks outside the range of the training data, as it works by smoothly combining existing observations, instead of explicitly fitting a function. The predictions of a kernel smoother lie in the convex hull of the training data, whereas a neural network can produce infeasible outputs (like negative power) for inputs that have not been seen before.

2 Problem Definition

In this section we introduce our ground truth dataset, and formalise the problems we wish to solve.

2.1 Ground Truth Dataset

Our dataset consists of hourly power data from four buildings, covering a period of 1.5 years starting from January 1st 2007. These buildings comprise:

Building A

A fifty year old, 12-storey, 10,500m^2 office tower. This building is primarily of concrete construction with a two-zone (perimeter/core) HVAC system providing heating, cooling and ventilation.

Building B

A modern, 16,700m^2 library building with mechanical ventilation and heating. This building has different envelope treatments based on orientation, with high glazing on the east facade and small punched windows on the west. There are no operable windows or other natural ventilation.

Building C

A 6-storey, 4920m^2 office and lab facility. The construction is brick on concrete, with operable windows, and mechanical heating and ventilation, but no cooling.

Building D

A 3-storey, 3160m^2 "green" building, with recycled timber frame and brick-cladding, fan assisted natural ventilation and cooling, and a daylight-responsive lighting control system.

Weather data from a national forecasting authority were used to generate an environmental variable dataset for the same time period, including temperature, humidity, wind speed and direction.

2.2 Energy Modelling

We use the same cost function for both training and evaluation of our energy modelling strategies. This is the root mean square (rms) power error between our predictions and the ground truth. For example, the rms test error is given by

$$e_{rms} = \left(\frac{1}{N_{\mathcal{S}}} \sum_{i \epsilon \mathcal{S}} (y_i - \hat{y}_i)^2 \right)^{\frac{1}{2}} \tag{1}$$

where $\mathcal{S}$ is the test dataset, y_i is the true and $\hat{y}_i$ the predicted power from the algorithm under study at hour i.

2.3 Event Detection

We focus on detecting peak power events. Peak power events can strain energy supply side and typically incur high demand charges from utility companies, so there is incentive for facility managers to minimize their occurrence. For the purposes of this work, we have defined a peak power threshold by hand for each building. We flag a likely peak power event if the probability that the future power exceeds the peak power value at hour i is greater than some threshold. Detecting anomalous usage events (i.e. higher or lower-than-average consumption due to equipment failure etc.) could be performed in a similar manner by computing the probability of the current power usage given the prediction.

2.4 Parameterisation

We use a common parametrisation for regression variables, and base our estimates of power on the following:

Time

We include 4 measurements of time, that are periodic on a daily, weekly, monthly and annual basis. Temporal measurements are mapped to the unit circle:

$$[\cos(2\pi t/T), \sin(2\pi t/T)].$$

Each timescale is represented by 2 dimensions (sine, cosine) giving a total of 8 time parameters. Distances in this space approximate differences in time for small distances, and the absence of a modulus operator allows us to use a general purpose nearest neighbour algorithm for the kernel smoother (described in Section 4).

Temperature

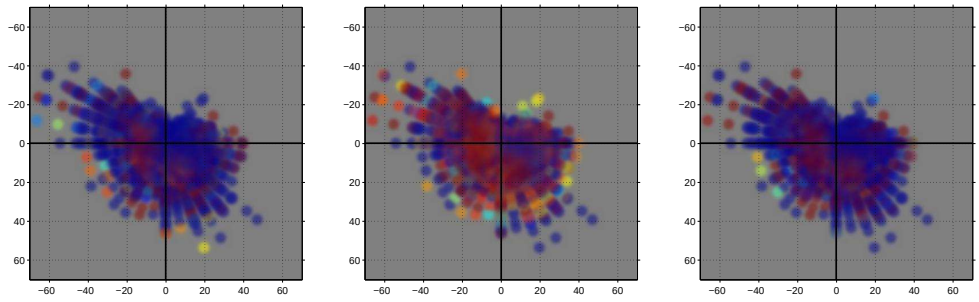

Figure 2: Wind and building power usage. Left to right: buildings A, B, C. These colour-coded plots show the effect of wind on building power usage. Red shades correspond to high (normalised) power, with blue corresponding to lower power. The higher power events (red shades) are often caused by warm south-westerly winds, which necessitate air conditioning. Note that wind and temperature will also often be correlated. The axes on each of these plots run from ±70 km/h in the North-South and East-West directions.

We sample temperature and other environmental measurements every hour. The temperature values are smoothed using exponential weighting functions with time constants of 1.5, 24 and 168 hours, enabling regression algorithms to pick up dependencies of power on temperature at various timescales, i.e., hourly, daily, weekly. These time constants were chosen by hand, though in principle they could also be learnt from training data (see Section 7.4).

Humidity

Humidity is smoothed using an exponential with a time constant of 24 hours.

Wind Velocity

We parametrise the wind speed and direction using the x, y components of the wind velocity (see Figure 2). These are also smoothed using an exponential with a time constant of 24 hours.

This gives a total of 14 parameters, and these identical inputs are fed to each of the learning algorithms that we will describe in the next section. As a preprocessing step, we normalise each input parameter by subtracting the mean and dividing by the standard deviation. We also train separate models for weekdays, and weekends/holidays, as these have quite different characteristics. Our parametrisation neglects many relevant building operational factors, such as event programming, occupancy, and human behaviour. However, note that operational factors may be correlated with weather factors in some cases.

3 Learning Algorithms

We compare the following six algorithms:

Temporal Average

This baseline algorithm is a simple average over each temporal instance, e.g. the power prediction at 3pm on Monday is the average over all Mondays at 3pm.

Temperature Neighbours

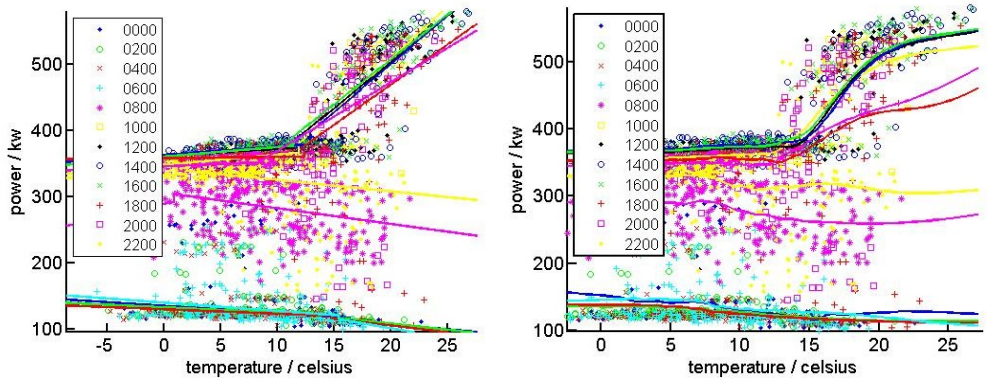

Figure 3: Kernel Regression for the B dataset. For this building, power is an increasing function of temperature, with a step change of 150kW at around 15°C when cooling units turn on. The piecewise linear model (left) picks up the discontinuity, but gives a much coarser model than the kernel smoother (right). Because kernel regression smoothly combines historical data measurements, it could adapt to even more complex changes such as multiple steps or non-linear behaviour.

We again average over each temporal instance, but only for nearest neighbours in terms of temperature that are within 5 degrees of the current temperature.

Multivariate Linear

We compute a least squares linear dependence of power on all of the input variables, except for the temporal variables. Separate multivariate linear models are learnt for each time of day, and separately for weekdays/holidays.

Piecewise Linear

This is a piecewise linear dependence of power with temperature. We learn the switching point, in addition to the line parameters, by minimising a robust cost function via a sampling approach. As above, these models are learnt separately for each hour of the week, for weekdays and weekends/holidays.

Neural Network

We train a committee of 5 feedforward neural networks, with 8 hidden units and sigmoidal transfer functions. The outputs of the 5 networks are averaged to generate the output power estimate. We trained the networks using the standard LM backpropogation algorithm in the MATLAB neural network toolbox.

Kernel Smoothing

We estimate the power based on a kernel density estimate over all the weather and time parameters. We learn the bandwidths of the kernel smoother by cross validation.

All the algorithms except the baseline "temporal average" are given a further input to specify public holidays, where the buildings are typically running at a fraction of their max capacity. In the next section we elaborate on our kernel smoothing approach.

4 Efficient Kernel Regression using K-Nearest Neighbours

4.1 Motivation

Kernel regression, also known as kernel smoothing, is a data driven approach to regression. It is closely related to a straightforward concept called a nearest neighbour smoother. In this algorithm,

one would look up in the historical dataset to find the day and time with the closest values of the regression parameters (e.g. time and temperature), and simply predict the power output of that nearest neighbour. In the context of a power prediction problem, this is equivalent to saying "what day/time in the past was most similar to today" and using that data to predict the current power output.

However, rather than taking a single nearest neighbour, a kernel smoother predicts a weighted average of nearby data items, with the weights being controlled by a kernel function. The kernel weights are important, because each parameter would normally have different units (e.g. temperature and humidity), and the relative importance of each parameter would also vary. Learning the weights in the kernel smoother tells the algorithm how to weight each dimension when computing nearest neighbours, for example, the algorithm might smooth over a 5 degree range of temperature and a 2 hour range in time of day.

4.2 Mathematical Description

In contrast to a neural network, which makes assumptions about the shape of the function that can be modelled, a kernel smoother begins with an assumption about the probability density of the data. Specifically, the assumption made is that the data has a kernel density, i.e.,

$$p(y, \mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} k(\mathbf{x}, \mathbf{x}_i)\, k'(y, y_i) \tag{2}$$

where y_i are the function values and $\mathbf{x}_i$ the associated regression parameters. $k(.,.)$ and $k'(.,.)$ are kernel functions, which are typically designed to give a high probability for nearby data. The expected value of y given $\mathbf{x}$ is given by

$$E(y|\mathbf{x}) = \int y\, \frac{p(y, \mathbf{x})}{p(\mathbf{x})} dy \tag{3}$$

$$= \frac{\sum_i \left[k(\mathbf{x}, \mathbf{x}_i) \int y\, k'(y, y_i)\, dy\right]}{\sum_i k(\mathbf{x}, \mathbf{x}_i)} \,. \tag{4}$$

For a zero mean kernel this gives rise to the well known Nadaraya-Watson Estimator [Nad64]

$$\hat{y} = \frac{\sum_i k(\mathbf{x}, \mathbf{x}_i)\, y_i}{\sum_i k(\mathbf{x}, \mathbf{x}_i)} \,, \tag{5}$$

where $\hat{y} = E(y|\mathbf{x})$. The expected value of power $\hat{y}$ is a weighted sum of power values for historical data with nearby time/weather parameters $\mathbf{x}$. In this work we will assume a Gaussian kernel with diagonal covariance, so

$$k(\mathbf{x}, \mathbf{x}_i) = N(\mathbf{x} - \mathbf{x}_i; \mathbf{0}, diag(\boldsymbol{\sigma}^2)) \,, \tag{6}$$

where $\boldsymbol{\sigma}$ is a vector of unknown kernel bandwidths corresponding to each dimension of the measurement space $\mathbf{x}$. See Figure 3 for an example of Kernel Smoothing applied to building energy data.

4.3 Learning

We learn the parameters of our kernel smoother by cross-validation, using the same objective function as in Equation 1, but evaluated over a validation set $\mathcal{V}$. $\hat{y}$ is the predicted power from the kernel smoother:

$$\hat{y}_i = \frac{\sum_{j \epsilon \mathcal{T}} k(\mathbf{x}_i, \mathbf{x}_j)\, y_j}{\sum_j k(\mathbf{x}_i, \mathbf{x}_j)} \ , \tag{7}$$

where $\mathbf{x}_i$ is the current measurement vector including time and environmental parameters such as the temperature, and $\{\mathbf{x}_j, y_j\}$ are pairs of environmental and power measurements from the set of training data $\mathcal{T}$. We can learn the optimal settings for $\boldsymbol{\sigma}$ by minimising Equation 1 with respect to $\boldsymbol{\sigma}$ over a cross validation set $\mathcal{V}$:

$$\boldsymbol{\sigma}^* = \arg\min_{\boldsymbol{\sigma}} \sum_{i \epsilon \mathcal{V}} (y_i - \hat{y}_i)^2 \ . \tag{8}$$

This is a non-linear least squares problem which can be solved using the Levenberg-Marquardt algorithm [NW99].

4.4 Fast Kernel Regression using a k-NN Approximation

Computing the kernel regression estimate for a single time instance requires a summation over every historical power/weather measurement in the training dataset (equation 7). In practice, this is too slow to compute, and would render the learning strategy of Section 4.3 infeasible using current hardware. To solve this problem, we instead use an approximation to equation 7, that can be computed very efficiently:

$$\hat{y}_i = \frac{\sum_{j \epsilon NN(i)} k(\mathbf{x}_i, \mathbf{x}_j)\, y_j}{\sum_{j \epsilon NN(i)} k(\mathbf{x}_i, \mathbf{x}_j)} \ . \tag{9}$$

The set $NN(i)$ consists of the k measurements with the largest values of $k(\mathbf{x}_i, \mathbf{x}_j)$ in the training set. Since we are using a Gaussian kernel, these can be computed very efficiently using an approximate euclidean k-nearest neighbour algorithm in a scaled space $\mathbf{x}' = [x'_1, x'_2 ...]$ where $x'_i = x_i / \sigma_i$ and $\mathbf{x} = [x_1, x_2 ...]$. To compute approximate nearest-neighbours efficiently, we use a best-bin first k-d tree as in [BL97].

Since the training dataset is organised as a tree, new data can be quickly assimilated into the model with a cost of $O(\log N)$. This is ideal for a real-time application, where retraining of $\boldsymbol{\sigma}$ values is performed on a fixed cycle (e.g. weekly), but it is desirable to incorporate the effect of new data as soon as it is available.

5 Probabilistic Peak Power Event Detection

One advantage of a predictive model of building energy usage is the ability to detect and flag anomalous events. For example, we might alert the facility manager if the actual power usage is outside of 3 standard deviations of the normal energy usage given the current weather conditions. This could also be used in a control setting. For example, if we knew that a high power usage was expected tomorrow, we might balance our future loads accordingly, for instance by pre-heating or pre-cooling the building. To demonstrate probabilistic event detection, we attempt to detect and flag all peak power events in the dataset, when the actual power usage is above a given threshold. To do this we assume a Gaussian distribution for the residuals

$$r_i = y_i - \hat{y}_i \sim N(0, \sigma_n^2) \ , \tag{10}$$

where σ_n is computed for each building model from the standard deviation of the residuals over the entire training set. The probability that the actual power exceeds some value y_{max} is given by

$$p(y_i > y_{max}) = \int_{y_{max}-\hat{y}_i}^{\infty} N(r; 0, \sigma_n^2)\, dr \tag{11}$$

$$= 0.5 \times (1 - \text{erf}((y_{max} - \hat{y}_i)/\sigma_n)) \,. \tag{12}$$

We declare a peak power event if this probability exceeds a certain threshold p_{thresh}, i.e. if

$$0.5 \times (1 - \text{erf}((y_{max} - \hat{y}_i)/\sigma_n)) > p_{thresh} \,. \tag{13}$$

In reality any such scheme will be subject to false positives and false negatives, and an appropriate operating point should be chosen. We do this by plotting receiver operating characteristic (ROC) curves obtained by varying p_{thresh} over the range $[0, 1]$. These curves show the tradeoff between true positive (correct detection rate) and false positive (false alarm rate) for all possible thresholds, evaluated over the test set, e.g., a correct detection would occur if the actual power were above y_{max} and the estimated probability of exceeding it $p(y_i > y_{max})$ was greater than the threshold.

6 Experiments

We perform two main experiments using our new dataset, as well as benchmarking using the ASHRAE Predictor Shootout dataset [KH94]. Firstly, we evaluate the relative performance of all of the algorithms with a large training dataset (1 year). A test set of 26 weeks is selected as every 3rd week in the 1.5 year dataset, and true power values in the test set are unknown to the algorithms. Each algorithm generates predictions for each of the 26 test weeks by attempting to generalise from the training dataset, and we evaluate their performance by computing the rms errors against the held-out ground truth. In addition, we use the methodology described in Section 5 to detect peak power events in the held-out test set, and compare this prediction performance against the ground truth.

Secondly, we perform experiments to test the performance of each algorithm as the size of the training dataset is decreased from 1 year down to 2 weeks. The small training dataset cases are intended to simulate the situation that occurs when a new building is added to a portfolio of buildings to be modelled, or an existing building undergoes a significant retrofit that alters it's energy performance. It is important for the models to quickly adapt and generate accurate predictions as soon as possible in this case.

Finally, we compare the performance of our algorithm with those entered in the Energy Predictor Shootout competition, training the algorithm using the supplied training data, and testing the performance in prediction of the 3 targets of the "A" dataset: namely whole-building electricity, cooling and heating water loads for the EC building at Texas A&M University.

7 Results

7.1 Energy Prediction Performance (New Dataset)

Prediction results in terms of the overall rms errors for the 26 test weeks are shown in Table 1. These rms errors have been normalised by the mean power usage for each building, which is equivalent to the coefficient of variation CV(RMSE) (standard deviation of prediction errors divided by the mean target value). For every building except D, the kernel smoother gives the best results. These results exceed the performance of neural networks by 2% (A), 7% (C) and 26% (B). The largest range of performance results was obtained for building B, which also has the most varied demand profile, with significant variations with respect to temperature, weekly cycles etc. Kernel smoothers are ideal at adapting to complex non-linear behaviour such as this.

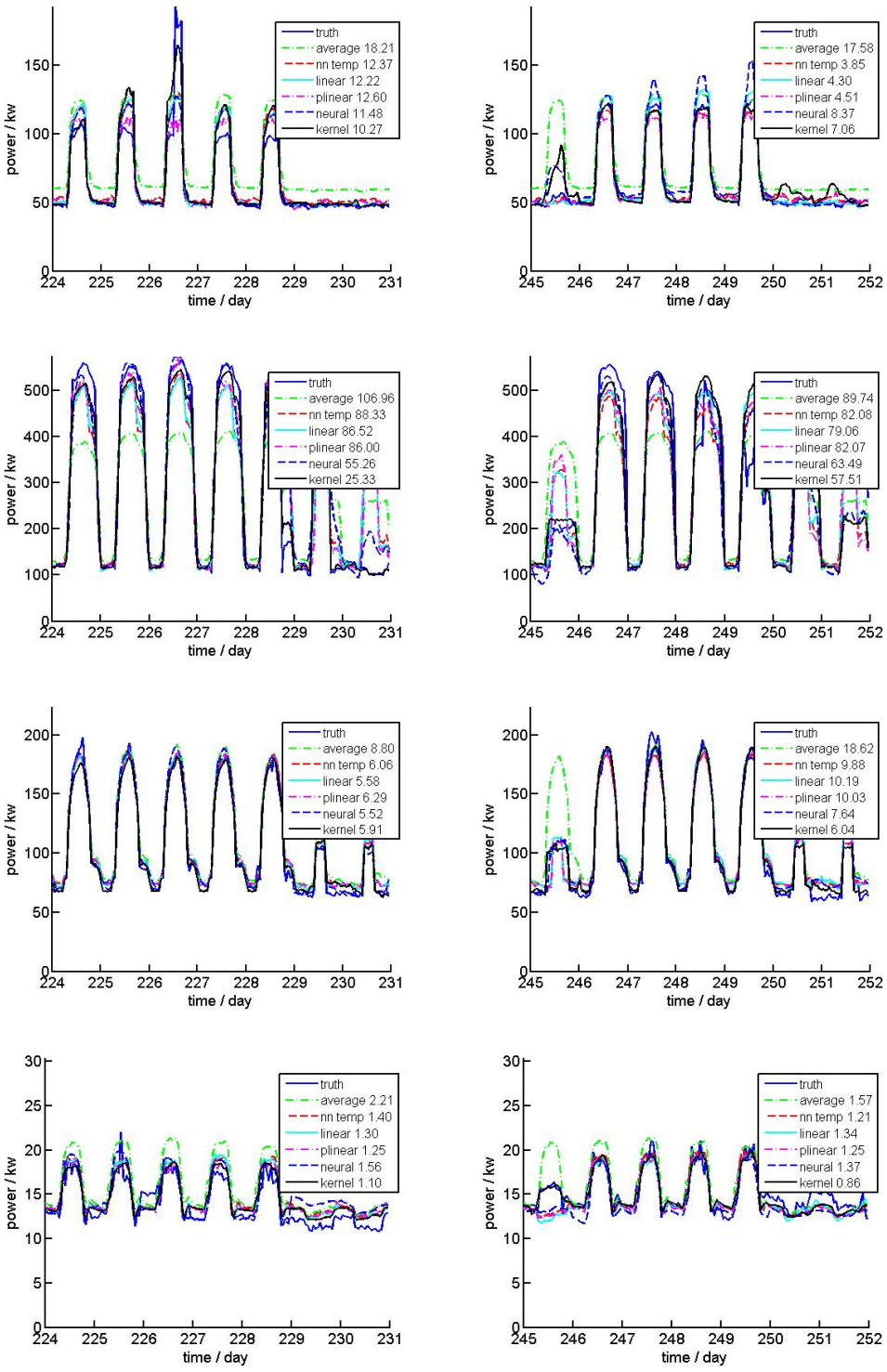

Figure 4: Predicted curves for all 4 buildings, weeks 11-12 (this is a 2 week subset of the 26 test weeks). Top to bottom: A, B, C, D. Note that day 245 is a holiday, leading to widely divergent power predictions for this unusual event. Also, building A experiences a demand spike on day 226, which is correctly predicted by the kernel smoother but missed by the other models. The rms errors for each algorithm for each week are shown in the top right of each plot.

Building	Algorithm					
	average	temp nn	linear	p. linear	neural net	kernel
A	16.99	11.86	12.20	12.41	10.68	**10.49**
B	22.46	19.47	20.05	19.87	14.76	**10.86**
C	9.98	8.49	8.54	8.61	7.75	**7.14**
D	12.00	**10.33**	10.34	10.49	10.75	10.67

Table 1: Rms errors for each of the 4 buildings for each algorithm. These results are quoted as a percentage of the buildings mean power output. This is equivalent to the coefficient of variation CV(RMSE). The best results for each building are shown in boldface.

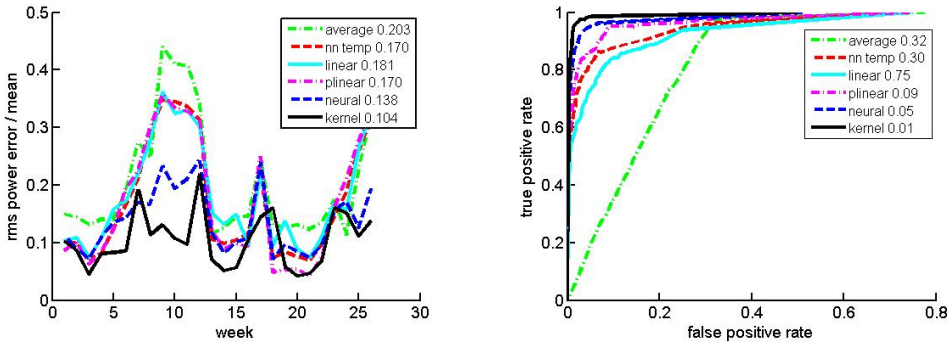

Figure 5: Modelling and event detection performance for the B building. The left figure shows the rms errors (normalised by the mean building power usage) for each of the 26 test weeks. Note that the kernel smoother gives the lowest fitting errors on almost every test week, followed by the neural network, and the conventional modelling techniques. The right figure shows the ROC performance for detection of peak power events. The 95% error rates for each algorithm are quoted in the legend. As expected from the more accurate fitting results, the kernel smoother is also more reliable at predicting peak power events.

These results were obtained by training each algorithm over 52 weeks of training data, and testing on 26 weeks. For examples of the actual predictions produced by each algorithm, please see Figure 4. Detailed rms errors per week and peak power event detection results for the building B are shown in Figure 5. As expected, the performance in terms of event detection increases with the accuracy of prediction, with kernel smoothers giving the most accurate predictions, followed by neural networks and linear models.

7.2 Small Training Sets

A key advantage of kernel regression in comparison to neural networks in the application presented here is that it is much less susceptible to overfitting. See Figure 6 for the results of an experiment to test the performance with a continuous range of training dataset size. Note that for small numbers of training weeks (less than a couple of months), the neural networks exhibit a poor average performance, with large variance. For small numbers of training weeks, the kernel smoother performs almost equivalently to the baseline "average" model, showing that it is not overfitting using small amounts of data. The main reason for this is that neural networks require many more parameters (120 in the examples shown here) than does kernel regression (14 in our examples). This is made possible because the 14 parameters used by the kernel smoother are each highly relevant to the problem. Note that the asymptotic performance of the kernel smoother is about 25% better

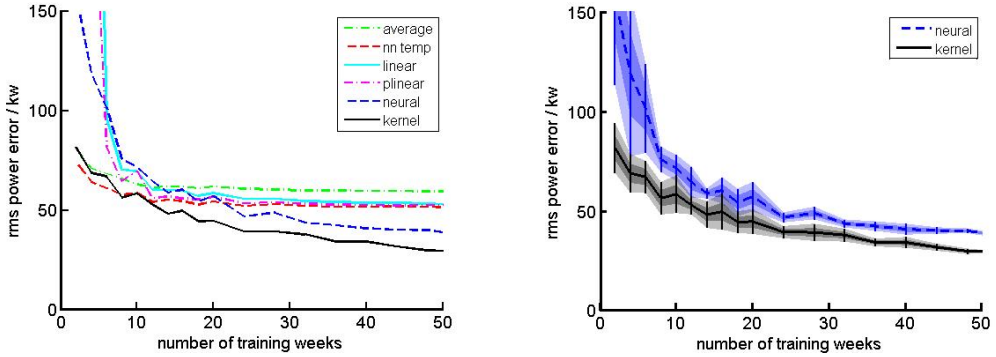

Figure 6: Algorithm performance versus size of the training dataset. The left figure shows the mean rms error of each algorithm for the B dataset, as the number of training weeks is varied from 2 to 26. Each data point is computed as the mean of 10 separate training runs, with a randomised selection of training weeks. The right figure shows the mean and standard deviation for just the kernel smoother and the neural network models. Note that the neural network has high error and high variance for small numbers of training weeks, showing that it is overfitting the data. Kernel smoothers, on the other hand, have almost identical error to the basic average models for small numbers of training weeks, showing their robustness to overfitting. The kernel smoother asymptotes to an error around 25% less than for the neural network as the training set becomes larger, with a low standard deviation.

than the neural networks as well. We again averaged results from a committee of 5 neural networks to generate these results. The results for neural networks would be much worse if this was not done.

7.3 Interpretation of Kernel Weights

Another positive benefit of our kernel smoothing approach over neural networks is the ability to interpret the model parameters that are learnt. Examples of optimal parameter settings are shown in Figure 7. Here we have plotted the reciprocal of the σ parameters, as this quantity gives some idea of the importance attached to that parameter. The highest weighted parameters are usually time of day, followed by time of year for most buildings (week and month are less important). This suggests that annual variations, such as holidays and exam periods are more important than weekly or monthly cycles when predicting power.

The importance of the weather parameters varies quite a bit between buildings. Temperature was always the highest weighted weather parameter, but the most important time scale for temperature varies (e.g., 1.5 hours for building A and 24 hours for building B). This is consistent with the physics of buildings: A is an older building with a poorly sealed envelope, whereas B is better insulated and thermally massive, so we expect it to have a larger time constant in its response to temperature. The A model also has by far the largest weights for the wind parameters, which is consistent with the fact that it is the tallest building in the surrounding area and highly exposed to the wind.

The buildings also have significantly different weights for the different time periodicities. For example, B is the only building with a significant weight for the time of week parameters. The reason for this can be seen in Figure 4 – for all of the buildings except B, each day of the week is almost identical, whereas for B the power trace on Friday looks quite different, dropping to a lower power in the early evening. Hence, day of the week is an important input parameter when predicting power usage for building B.

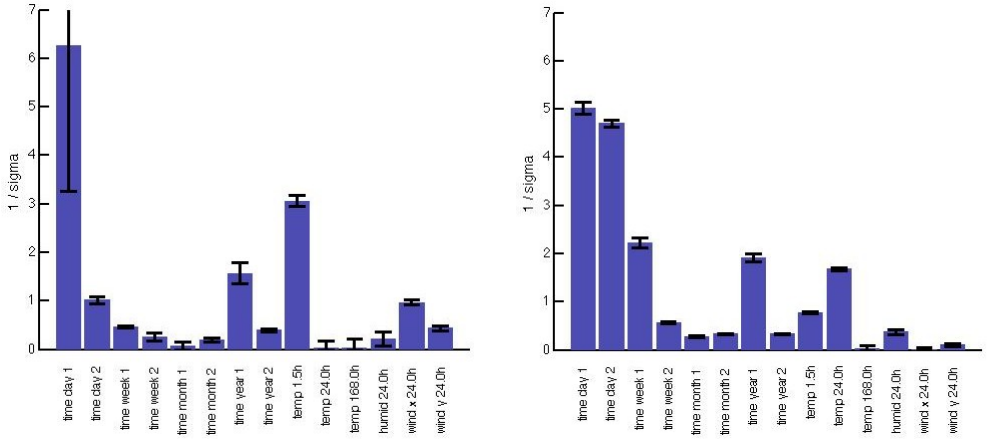

Figure 7: Optimal kernel weights learned from training A and B weekly kernel smoother models over a 1 year training period. The error bars show standard deviations over 10 randomised initialisations.

7.4 Comparison to Predictor Shootout Winners

Finally, we have compared our algorithm against the best performing competitors in the original ASHRAE Energy Predictor Shootout. After training using the supplied 17 week dataset, we evaluate Coefficient of Variation (CV) and Mean Bias Error (MBE) statistics over the 8 test weeks. The results are shown in Table 2. In addition to training a single kernel smoother as described previously, we also experimented with using a committee of 5 kernel smoothers, averaging the results of each separately trained predictor to get the final output. The kernel committee results are denoted "kernelc" and the single kernel smoother results denoted "kernel" in the table, "neural" again shows results from the MATLAB neural network implementation. Each of these algorithms is given identical inputs of: cos/sin of the day, half-day, month and year angles, temperature and solar radiation smoothed on timescales of 1.5, 24 and 72 hours, and humidity and windspeed smoothed over 24 hours. The other algorithms in the table are (9) Mackay's Bayesian Non-Linear Modelling, (6) Ohlsson's Feedforward Multi-layer Perceptron, (2) Feuston's Neural Network with Pre and Post Processing.

As can be seen from the table, our algorithms generate results that are comparable with the ASHRAE Shootout winners. For the second target (WBCW) the kernelc algorithm gave the best overall result in terms of Coefficient of Variation. For targets 1 and 3, the other algorithms performed better. In terms of average CV our algorithm places 6th overall in the predictor shootout results reported in [KH94]. However, in terms of MBE our algorithm performs worse. This could be due to our choice of objective function: we effectively optimise for CV by minimising rms over the training set. We could explore alternative loss functions, such as Huber [Hub81], to tradeoff between CV and MBE. The results we quote are averages for 5 trials (with randomised subsets of training data).

Algorithm (9) (Mackay) still generates the best results overall, beating many other neural network implementations in the competition, which suggests that it is some particular feature of this implementation, e.g., input preprocessing, network architecture, or training approach, that is important. One feature of (9) is the use of priors on the relevance of input parameters (Automatic Relevance Determination). Although to some extent our kernel bandwidths should allow for variable relevance weighting in this way, we have found that providing a smaller set of manually selected relevant input parameters gave better results in some cases. For example, using only time parameters (excluding weather parameters) improved the performance of the kernel committee from a CV of 13.61 to 11.92 for WBE, and 33.01 to 24.26 for WBHW (although the result was worse for WBCW). This observation could in principle be utilised to automatically generate parsimonious models. For example, one could use a sparsity (L1) prior on the kernel weights ($1/\sigma_i$), which would favour models with

Algorithm	WBE CV	WBE MBE	CHW CV	CHW MBE	HW CV	HW MBE	AVG CV	AVG MBE
9	**10.36**	8.06	13.02	-6.37	**15.24**	-5.84	**12.87**	6.75
6	11.78	10.50	12.97	-5.95	30.63	-27.33	18.46	14.59
2	11.89	8.01	13.69	-6.67	31.65	-27.55	19.08	14.08
kernelc	13.61	11.00	**12.40**	-9.05	33.01	-30.17	19.67	16.74
kernel	14.55	11.59	13.15	-9.41	34.75	-31.53	20.82	17.51
neural	13.03	8.20	15.75	-6.51	72.18	-69.41	33.65	28.61

Table 2: Results on the Predictor Shootout Dataset A. CV = Coefficient of Variation, MBE = Mean Bias Error, WBE = Whole Building Electricity, CHW = Chilled Water, HW = Hot Water. The best performing CV results for each test case are shown in boldface.

fewer parameters having significant weights [HTF09]. As well as automatically selecting relevant environmental variables, one could in principle include optimisation over the smoothing timescales of these variables in our learning approach. This would incur the additional computational expense of re-smoothing the training set each iteration to compute derivatives, but should be computationally feasible.

In future work we would like to perform a more thorough study with reimplementations of these algorithms over a larger training set. Based on our results (e.g., Figure 6), 17 weeks is rather a short time for training, so we posit that there could be significant changes in the results quoted in Table 2 with slightly different training data. In other words, it would be instructive to know the uncertainty of prediction for the top performing algorithms in the shootout competition.

As previously, we can interpret our models by visualising the kernel bandwidths, see Figure 8. We found that the chilled and hot water loads (WBCW, WBHW) have similar models, with little dependence on time of day or month, but strong dependence on the current and daily average temperature. In contrast the Whole Building Electricity (WBE) target depends strongly on the time of day, and annual time, with a much smaller influence of the weather parameters.

8 Conclusion

We have proposed a new technique for building energy modelling using kernel regression. Using a new dataset of power and weather measurements for 4 buildings over 1.5 years, we have tested our technique and compared it to a standard neural network algorithm. We have also compared performance against the top performing algorithms in the ASHRAE Shootout dataset. We find that our kernel smoothers give results that are comparable to neural networks in prediction tasks, and can outperform them significantly when the training set is small. Furthermore, our models have been found to provide an additional level of information about the building under study, which other models lack, and are capable of showing the sensitivity of its power response to time and weather parameters. We have also described an efficient implementation suitable for use in scalable, real-time energy modelling systems.

References

[AMH00] K. Andersen, H. Madsen, and L. Hansen. Modelling the heat dynamics of a building using stochastic differential equations. *Energy and Buildings*, 31:13–24, 2000.

[AWG09] Y. Agarwal, T. Weng, and R. Gupta. The energy dashboard: Improving the visibility of energy consumption at a campus-wide scale. In *BuildSys'09. Proceedings of the*

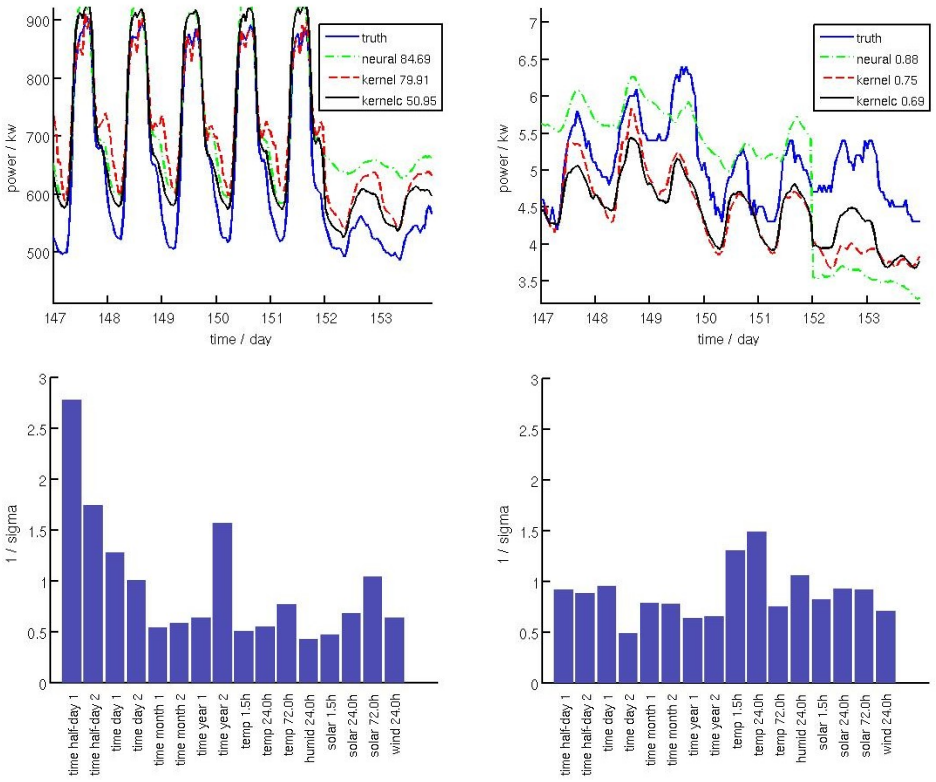

Figure 8: Predictions and models for the ASHRAE Predictor Shootout Dataset. Top row: prediction results for the kernel smoother and neural network models (legend shows rms errors). Bottom row: kernel smoother model parameters used to generate the predictions. Prediction targets are (left to right): Whole Building Electricity (WBE), Whole Building Chilled Water (WBCW) for the "A" dataset.

First ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Buildings., Berkeley, CA, November 2009.

[BC 10] BC Hydro. BC Hydro Smart Metering and Infrastructure Program, 2010.

[BL97] J. Beis and D. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *Computer Vision and Pattern Recognition*, pages 1000–1006, 1997.

[CLP+04] D. Crawley, L. Lawrie, C. Pedersen, F. Winkelmann, M. Witte, R. Strand, R. Liesen, W. Buhl, Y. Huang, R. Henninger, J. Glazer, D. Fisher, D. Shirey, B. Griffith, P. Ellis, and L. Gu. EnergyPlus: An update. In *Proceedings of SimBuild, Building Sustainability and Performance Through Simulation*, Boulder, August 2004.

[CLPW00] D. Crawley, L. Lawrie, C. Pedersen, and F. Winkelmann. EnergyPlus: Energy simulation program. *ASHRAE Journal*, 42(4):49–56, 2000.

[Dar06] S. Darby. The effectiveness of feedback on energy consumption. a review for defra of the literature on metering, billing and direct displays, 2006. Environmental Change Institute, University of Oxford.

[DCL05] B. Dong, C. Cao, and S. Lee. Applying support vector machines to predict building energy consumption in tropical region. *Energy and Buildings*, 37:545–553, 2005.

[DH96] R. Dodier and G. Henze. A statistical analysis of neural networks as applied to building energy prediction. In *ASME/JSME International Solar Energy Conference*, March 1996.

[DRC99] A. Dhar, T. Reddy, and D. Claridge. A Fourier series model to predict hourly heating and cooling energy use in commercial buildings with outdoor temperature as the only weather variable. *Journal of Solar Energy Engineering*, 121(1):47–53, 1999.

[GAB94] C. Georgescu, A. Afshari, and G. Bornard. Optimal adpative predictive control and fault detection of residential building heating systems. In *Proceedings of the Third IEEE Conference on Control Applications*, volume 3, pages 1601–1606, Glasgow, August 1994.

[GPGP09] J. Granderson, M. Piette, G. Ghatikar, and P. Price. Building energy information systems: State of technology and user case studies. Technical Report LBNL-2899E, Lawrence Berkeley National Laboratory, 2009.

[HBS98] J. S. Haberl and T. E. Bou-Saada. Procedures for calibrating hourly simulation models to measured building energy and environmental data. *Journal of Solar Energy Engineering*, 120:193–204, 1998.

[HDK98] G. Henze, R. Dodier, and M. Krarti. Development of a predictive optimal controller for thermal energy storage systems. *ASHRAE Transactions*, 104:54, 1998.

[HS03] S. Huang and K. Shih. Short-term load forecasting via ARMA model identification including non-Gaussian process considerations. *IEEE Transactions on Power Systems*, 18(2):673–679, 2003.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, second edition edition, February 2009.

[Hub81] P. J. Huber. *Robust Statistics*. Wiley, 1981.

[Kal00] S.A. Kalogirou. Applications of artificial neural-networks for energy systems. *Applied Energy*, 67:17–35, 2000.

[KDM96] M. Kawashima, C. Dorgan, and J. Mitchell. Optimizing system control with load prediction by neural networks for an ice-storage system. *ASHRAE Transactions*, 102(1):1169–1178, 1996.

[KH94] J. Kreider and J. Haberl. Predicting hourly building energy use: the great energy predictor shoot-out. Overview and discussion of results. *ASHRAE Transactions*, 94(17/7):1104–1118, 1994.

[KH96] J. Kreider and J. Haberl. The great energy predictor shoot-out II: measuring retrofit savings. Overview and discussion of results. *ASHRAE Transactions*, 96(3/4):419–435, 1996.

[KSG06] S. Karatasou, M. Santamouris, and V. Geros. Modeling and predicting building's energy use with artificial neural networks: Methods and results. *Energy and Buildings*, 38(8):949–958, August 2006.

[LB04] K. Lee and J. Braun. Development and application of an inverse building model for demand response in small commercial buildings. In *SimBuild 2004, IBPSA-USA National Conference*, Boulder, CO, August 2004.

[Mac94] David Mackay. Bayesian non-linear modelling for the prediction competition. In *ASHRAE Transactions*, volume 100, pages 1053–1062, 1994.

[Mah01] Ardeshir Mahdavi. Simulation-based control of building systems operation. *Building and Environment*, 36:789–796, 2001.

[Mah04] A. Mahdavi. Reflections on computational building models. *Building and Environment*, 39:913–925, 2004.

[MM09] E. Mills and P. Mathew. Monitoring-based commissioning: Benchmarking analysis of 24 uc/csu/iou projects. Technical Report LBNL-1972E, Lawrence Berkeley National Laboratory, 2009.

[Nad64] E. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9(1):141–142, 1964.

[New09] New Buildings Institute. Advanced metering and energy information systems, 2009. Grant 83378201.

[NW99] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, Berlin, Heidelberg, 1999.

[PSJ+07] J. Petersen, V. Shunturov, K. Janda, G. Platt, and K. Weinberger. Dormitory residents reduce electricity consumption when exposed to real-time visual feedback and incentives. *International Journal of Sustainability in Higher Education*, 8(1):16–33, 2007.

[Sec07] Secretary of State of California. Assembly Bill 1103: An act to add Section 25402.10 to the Public Resources Code, relating to energy, 2007.

[WJ95] M. Wand and M. Jones. *Kernel Smoothing*. Chapman and Hall, London, 1995.

[WvD07] P. Wouters and D. van Dijk. Energy performance building directive platform: Overall context and activities. *EPBD Buildings Platform*, P039, 2007.

[YRZ05] J. Yang, H. Rivard, and R. Zmeureanu. On-line building energy prediction using adaptive artificial neural networks. *Energy and Buildings*, 37:1250–1259, 2005.