Predicting fuel consumption for commercial buildings with machine learning algorithms [*]

Aowabin Rahman, Amanda D. Smith [†]

August 7, 2017

ABSTRACT

This paper presents a modeling framework that uses machine learning algorithms to make long-term, i.e. one year-ahead predictions, of fuel consumption in multiple types of commercial prototype buildings at one-hour resolutions. Weather and schedule variables were used as model inputs, and the hourly fuel consumption simulated with EnergyPlus provided target values. The data was partitioned on a monthly basis, and a feature selection method was incorporated as part of the model to select the best subset of input variables for a given month. Neural networks (NN) and Gaussian process (GP) regression were shown to perform better than multivariate linear regression and ridge regression, and as such, were included as part of the model. The modeling framework was applied to make predictions about fuel consumption in a small office, supermarket, and restaurant in multiple climate zone. It was shown that for all climate zones for all months, the maximum errors pertaining to one year-ahead forecasts of fuel consumption made by the ML model are 15.7 MJ (14,880 Btu), 284.3 MJ (268,516 Btu) and 74.0 MJ (70,138 Btu) respectively. The methods and results from this study can be used to estimate on-site fuel consumption and emissions from buildings, thereby enabling improved decisions pertaining to building efficiency with respect to fuel use.

NOMENCLATURE

ARX	Auto-regressive eXogenous
CV	Cross-validation
CDD	Cooling degree-days
$e+$	EnergyPlus building simulation package
EPR	Evolutionary Polynomial Regression
GP	Gaussian Processes

[*] Author's version. Published version: https://doi.org/10.1016/j.enbuild.2017.07.017

[†] Site-Specific Energy Systems Laboratory, Department of Mechanical Engineering, University of Utah, Salt Lake City, Utah, USA, 84112. http://energysystems.mech.utah.edu amanda.d.smith@utah.edu

HDD	Heating degree-days
ML	Machine Learning
MLR	Multivariate linear regression
$NARX$	Neural auto-regressive with eXogenous input
NN	Neural network
σ^2_{gauss}	Predictive variance obtained from GP prediction
σ_t	Standard deviations of relevant weather variables in the training set
σ_e	Standard deviations of relevant weather variables in the test set
λ	Regularization parameter in ridge regression
ϕ	Expanded feature space in ridge regression/EPR
ω	Weights assigned to features/variables in a given ML algorithm
γ	Learning rate in NN weight-update step
μ_t	Mean values of relevant weather variables in the training set
μ_e	Mean values of relevant weather variables in the test set
$\rho_w(j)$	Normalized Pearson coefficient of weather variable j
b	Kernel width in Gaussian Processes
C	Cost Function
c	Constant associated with shifted/translated schedule
e_{CV}	Cross-validation error associated with a given ML algorithm.
e_t	Transient error in predicting fuel consumption at one-hour resolution
e_{rms}	Root mean squared error in predicting hourly fuel consumption for a given ML algorithm.
$e_{rms,NN}$	Root mean squared error in predicting hourly fuel consumption using static NN.
$e_{rms,GP}$	Root mean squared error in predicting hourly fuel consumption using GP regression.
$e_{cov,rms}$	RMS error due to target values lying outside GP-suggested covariance bounds
H	Number of nodes in the hidden layer
J	Total number of relevant variables considered for ML algorithm post feature-selection
j_1	Total number of relevant weather variables considered for ML algorithm post feature-selection
j_2	Total number of relevant schedule variables considered for ML algorithm post feature-selection
l	Number of layers in neural network
k	Number of subsets of training data used for cross validation
m	Number of nodes in a given neural network layer l
$K(\mathbf{X_t}, \mathbf{X_e})$	Covariance matrix obtained in GP with training features $\mathbf{X_t}$ and test features $\mathbf{X_e}$

N	Total number of observations/data points available for training
P	Total number of weather variables considered post data generation
Q	Total number of schedule variables considered post data generation
RH	Relative Humidity
R_w(p)	Pearson Coefficient of p^{th} weather variable.
R_v(q)	Pearson Coefficient of q^{th} schedule variable.
t	Time (in hours)
T_{db}	Dry-bulb temperature
$\mathbf{v_q}$	Generic q^{th} schedule variable
$\mathbf{v_t}$	Set of schedule variables in training phase prior to ordering by Pearson coefficient
$\mathbf{v_e}$	Set of schedule variables in test phase prior to ordering by Pearson coefficient
$\mathbf{v'_t}$	Set of schedule variables in training phase after ordering by Pearson coefficient
$\mathbf{v'_e}$	Set of schedule variables in test phase after ordering by Pearson coefficient
v_{inf}	Infiltration schedule
$\mathbf{w_p}$	Generic p^{th} weather variable
$\mathbf{w_t}$	Set of weather variables in training phase prior to ordering by Pearson coefficient
$\mathbf{w_e}$	Set of weather variables in test phase prior to ordering by Pearson coefficient
$\mathbf{w'_t}$	Set of weather variables in training phase after ordering by Pearson coefficient
$\mathbf{w'_e}$	Set of weather variables in test phase after ordering by Pearson coefficient
$\mathbf{X}$	Generic feature set
$\mathbf{X_t}$	Training feature set (also called S_1)
$\mathbf{X_e}$	Test feature set (also called S_2)
y_{pred}	Predicted fuel consumption using a given ML algorithm.
$y_{pred,GP}$	Predicted fuel consumption using Gaussian process regression
$y_{pred,NN}$	Predicted fuel consumption using static neural networks
$y_{pred,NARX}$	Predicted fuel consumption using NARX
$y_{pred,ridge}$	Predicted fuel consumption using ridge regression
$y_{sim,t}$	Simulated fuel consumption generated using EnergyPlus corresponding to training data.
$y_{sim,e}$	Simulated fuel consumption generated using EnergyPlus corresponding to test data.

1 Introduction

Building energy consumption contributes to as much as 39% of CO_2 emissions in the United States[1], a significant portion of which comes from space and water heating and gas equipment end uses [2]. As such, there is a need to develop emission models that can estimate emissions from on-site stationary combustion sources, particularly those contributed by boiler operation. Estimating transient building fuel consumption, therefore, is a key step in developing an integrated building-energy related emissions model, which can subsequently be coupled with regional climate models. Thus, knowledge of transient fuel consumption would lead to a better understanding of impact of building emissions on regional climate, and would allow for better assessment of potential benefits of improved building design and implementation of zero-energy building technologies [3].

The most popular approach to modeling building heating demand is using physics-based building energy simulation packages such as BLAST, DOE2.1, eQUEST and EnergyPlus [4]. Such physics-based or deterministic models usually contain a set of governing partial differential equations that are derived from energy or mass balance considerations [5]. For instance, EnergyPlus employs an integrated solution scheme that solves for transient, zero-dimensional heat, air and moisture transfer equations that interconnect the different zones, air handling system and central plant equipment system inside a building [4] [6]. While physics-based models enable the user to understand how different heat and mass transfer processes affect building loads, they are often constrained by the complexity of the building designs they can allow for [5]. Such models often fail to account for complex and/or stochastic interactions between the energy systems in a building, and often the resulting simplifications can result in a loss in accuracy. Thus, the accuracy of these models could be well in excess of 100% [7] [8], and as such, these models are often better used as comparative tools to analyze relative benefits due to a building modification, rather than an accurate predictor of building energy consumption.

The alternatives to building energy simulation are statistical models and machine learning algorithms [9]. Fumo and Biswas [10] extensively reviewed several statistical approaches (including univariate, multivariate, autoregressive, conditional demand analysis models and compared the relative performances of these models with respect to hourly electric loads for an unoccupied house at TxAIRE. The authors found that the root mean squared error (RMSE) corresponding to univariate linear, univariate quadratic and multivariate linear methods were all between 40.2% to 41.1% of average hourly load.

Machine Learning (ML) methods attempt to build a model as it “learns” the behavior of a system from measured or observed data; however while ML methods heavily borrow concepts from statistics, they have some subtle yet important differences with respect to conventional methods. Rather than determining model parameters with respect to a pre-set function (as is the case with standard statistical approaches), ML methods find an approximation to a function within some hypothesis space, given a set of observed data, or in the case of unsupervised learning, even without the presence of observed data [11]. While statistical methods employ goodness of fit (or similar criterion) to as accuracy, ML methods emphasize on the prediction accuracy over model accuracy. Relative advantages of ML methods are further discussed in [11].

Neural networks (NN) are a type of biologically-inspired machine learning algorithm, which can be often described as interconnected set of “parallel distributed processors”, where the interconnected “neurons” replicate the learning process with training data and subsequently use the knowledge to estimate actual solutions [12] [13], [14]. A multi-layer feed-forward algorithm architecture is a popular configuration that consists of neurons arranged in multiple layers: an input layer, one or more hidden layers and an output layer. The neurons in each layer is connected to the subsequent layer by an activation function, which is a function of weighted sum of the outputs in the previous layers. The weights are adjusted using a back-propagation technique where a gradient descent algorithm is used to minimize the error between the actual data and the predicted solution. Neural networks have the advantage of replicating non-linear transformations and as such they can approximate complex or unknown functions fairly accurately [13] [14]. This allows neural networks to be frequently used in estimation of time-dependent electric and thermal loads. Tso et al. [15] determined that for households in Hong Kong with a monthly energy consumption of 100 kWh or above, neural networks can perform equally well or better than conventional regression methods and decision trees in predicting electrical loads.

Previous studies have also shown that autoregressive models that consider prior outputs as model inputs can accurately predict the periodicity in building heating loads, compared to static networks [16]. Yun et al. [16] observed that for multiple building types including a small office, medium-sized office and a mid-rise apartment, an indexed ARX model that considers weather variables and heating loads at previous time steps as inputs performed comparatively better than static models, including multivariate linear regressive (MLR) models. The relative prediction errors for hourly heating loads, as indicated by the coefficient of variation are between 9.0% and 56.5%, depending on the building type [16].

As such, autoregressive neural networks are popular in short-term load prediction applications [17, 18, 19, 20, 21, 22]. Charytuneyik et al. [19] used a 21-day training window to make one week-ahead predictions in electric loads, with a prediction mean squared error of 3.57%. Similarly Park et al. [20] used a multi-layered perceptron (MLP) algorithm to make 24-hour forecasts of electric load forecasts with averaged errors of less than 5%. Gonzalez [17] applied autoregressive NN's to make one-step ahead predictions of ambient temperature and hourly electric consumption, achieving a coefficient of variance of less than 2%.

Aside from neural networks, this paper also applied Gaussian process (GP) regression to predict building fuel consumption. Gaussian processes are a generalization of the Gaussian distribution and can be defined by a mean (i.e.prediction) and a covariance function for a given input vector [23]. As such, one of the biggest advantages of using GP regression is its ability to provide a predictive variance along with a point estimate for each test input. Heo and Zavala [24] developed and applied a GP regression model to predict daily chilled water consumption in an once building in Chicago, and observed that GP regression can capture non-linear behavior in energy consumption more accurately than standard linear regression methods used by ASHRAE [25].

While autoregressive neural networks are quite successful in making short-term forecasts with high accuracy, this paper aims to make long-term predictions for fuel consumption, which would subsequently be used to estimate hourly emissions from buildings. The objective, therefore, is to make predictions of fuel consumption profile at one-hour resolution for a future year, using hourly data from a previous year. This is a more difficult objective than making forecasts over short-term, i.e. over the period of a few hours, as the ML algorithm does not have access to recent data and can not adjust itself as it is making predictions. Thus, for the month of January, the proposed model would make predictions over 744 hourly time-steps in the test phase, without knowledge of actual data during the test phase. The proposed method uses a feature selection method to find the optimal subset of input variables needed to make predictions, and uses a combination of neural network and Gaussian process regression, which can provide both point estimates and prediction variance, and is robust to variability in weather data from training to test set.

In order to address key gaps and limitations in prior research, the objectives of this analysis have been identified as:

- Compare different ML regression algorithms including neural networks, Gaussian Process regression, ridge regression, multivariate linear regression and autoregressive neural networks respect to accuracy.
- Develop a machine learning scheme that provides predictions, as well as confidence intervals for fuel consumption at one-hour resolution over the time period of one year using hourly fuel consumption data for a previous year.
- Determine methods to quantitatively analyze effects of variability of weather data on ML prediction accuracies, and investigate how the ML prediction accuracies vary across multiple climate zones and building types.

As mentioned previously, the overarching goal of this study is to make long-term predictions of fuel consumption (i.e. over a time horizon of one year) at one hour-resolutions. Previous studies have focused on making short-term predictions over the time horizon of a few hours to few days at one-hour resolution predictions with good accuracy. However, relatively less work has been done on long-term forecasts of energy consumption at hourly or sub-hourly levels, which is a much more difficult objective. Thus, results from this study can subsequently be used by building designers, owners and engineers to make improved decisions pertaining to building efficiency (e.g. better demand response management strategies), as well as be useful in smart grid applications.

This sections that follow detail the development and analysis of a modeling framework that attempts to meet the aforementioned objective of making long-term forecasts at one-hour resolution with a low prediction error. Section 2 details the existing machine learning methods used in this study, whereas section 3 presents a description of used data, as well as explains how the methods introduced in section 2 were used to develop and optimize the modeling framework. Section 4 compares the performance of different machine learning algorithms and discusses why static neural network and Gaussian processes were selected as part of the overall modeling framework, analyzes the performance of the overall model in predicting fuel consumption in different building types and multiple climate zones, and evaluates the robustness of the model under perturbation of key modeling parameters. Finally, the conclusions are presented in section 5.

2 Description of Machine Learning Methods Used

2.1 Neural Networks

Neural Networks are machine learning algorithms which can be described as a network of interconnected "neurons" that model non-linear relationships between the input vector and the predicted values. In a multi-layered feed-forward NN, the neurons are arranged in several layers, with each neuron consisting of a vector of inputs, weights associated with each input and an activation function. The outputs from the activation function become inputs for the neurons in the subsequent layer. The process of learning, thus, becomes the procedure of learning the weights to minimize the mean squared error cost function, which is done by back-propagation [12].

Figure 1 shows a schematic of a static multi-layered feed-forward NN, where each node represents an activation function. In a given node m located in layer l, the activation function can be expressed as follows:

Table 1. Configurations of neural networks used in this analysis

Number of epochs for CV	10
Number of epochs for prediction	100
Number of inputs, $\|\mathbf{X}\|$ for CV	2,3,....(P+Q)
Number of inputs, $\|\mathbf{X}\|$ for prediction	J
Number of hidden layers	1
Activation function in hidden layer	Sigmoid
Activation function in final layer	Linear
Number of nodes in the hidden layer, H	$[\frac{N}{\|\mathbf{X}\| \log(N)}]^{1/2}$

$$A(m)^l = \frac{1}{1 + \exp(S(m)^l)} \tag{1}$$

In this analysis, a sigmoid layer was used as an activation function in the hidden, so as to ensure that output from the given layer is within the interval [0,1]. Here, $(S(m)^l)$ is the linear combination of the inputs from the previous layer, expressed as follows:

$$S(m)^l = \sum_{i=0}^{I} x_i \omega_{mi} \tag{2}$$

The weights in each layer are updated using stochastic gradient descent, which is a simple optimization algorithm [26] that minimizes the mean squared error by applying gradient descent on a smaller subset of data in each iteration. The weight update for each example in the training set can be expressed as:

$$\omega_{mi}^l = \omega_{mi} - \gamma \frac{\Delta e}{\Delta \omega_{mi}^l} \tag{3}$$

Here, ω_{mi}^l is magnitude of the weight assigned to the input in node i in layer l corresponding to the output in node m in layer $(l+1)$. γ is the learning rate, which is a parameter that denotes the magnitude by which the weight is updated in each iteration. e is the root-mean squared error, which can be expressed as:

$$e^2 = (y_{pred,NN} - y_{sim,t})^2 \tag{4}$$

Here, $y_{pred,NN}$ is the predicted value i.e. output in the final NN layer L during training, and $y_{sim,t}$ is the target value in training phase. The partial derivative $\frac{\Delta e}{\Delta \omega_{mi}}$ is obtained by back-propagation [26], [12].

Barron [27] suggested that for a neural network with one hidden layer, the optimal number of hidden nodes is $[\frac{N}{|\mathbf{X}| log(N)}]^{1/2}$, where $\mathbf{X}$ is the feature set i.e. either $\mathbf{X_t}$ or $\mathbf{X_e}$. The neural network configurations, applicable to both static NN and NARX, are summarized in table 1. The neural network toolbox in MATLAB 2015b [28] was used to implement regression using static NN and NARX in this paper.

The NARX (nonlinear autoregressive with eXogenous input) model was compared with the aforementioned static network in this paper. The NARX model also uses the same multi-layered feed-forward algorithm as the static network; however, the NARX model uses lagged time-series data, i.e. lagged inputs as well as outputs, are used as inputs to predict the current output. For the NARX network in this model, the current output was assumed to be a function of current input values (i.e. current weather and schedules), as well as outputs in the previous time steps. This can be expressed as follows:

$$y_{pred,NARX}(t) = f(y_{pred,NARX}(t-1), y_{pred,NARX}(t-2), ..y_{pred,NARX}(t-\tau); \mathbf{X}(t)) \tag{5}$$

Here in this analysis, we implemented NARX using outputs from five previous time steps, i.e. $\tau = 5$. The NARX network is trained using target values as time-laggged outputs, i.e. $y_{sim,t}(t-1)...y_{sim,t}(t-\tau)$. However, as the objective is to make long-term forecasts over an entire year at one-hour resolution and we do not have access to the target values $y_{sim,e}$ during test/prediction phase, the NARX network is used in a closed-loop configuration during prediction phase [28]. This means that the predicted values using the NARX algorithm, i.e. $y_{pred,NARX}(t-1)....y_{pred,NARX}(t-\tau)$, are used as lagged outputs.

2.2 Gaussian Processes

A Gaussian process (GP) can be defined as a collection of random variables, where any subset of these variables have a joint Gaussian distribution [23]. GP regression method uses a slightly different approach to other regression methods such as linear regression or neural networks - it makes inferences in the function space rather than the weight space [23]. This means that instead of learning weights that can be assigned to features (or some functions of features), the GP regression method assumes that the training and test points come from a joint distribution and learns the covariance matrix that defines this distribution. One of the biggest strengths of GP regression is its ability to provide confidence specific to each test point. Thus, if there are a lot of training points in the proximity of a test point, the uncertainty in GP prediction will be low and the confidence bounds will be tighter. On the other hand, if there are not too many points near the test point, the GP will provide looser confidence intervals associated with its prediction.

The covariance is often modeled using a squared exponential kernel:

$$k(\mathbf{X_{n_1}}, \mathbf{X_{n_2}}) = \sigma_f^2 exp\big[-\frac{|\mathbf{X_{n_1}}^2 - \mathbf{X_{n_2}}^2|}{\sigma_m^2}\big] \tag{6}$$

Here $\mathbf{X_{n_1}}$ and $\mathbf{X_{n_2}}$ are both data points with J features, σ_f is a hyper-parameter indicating the signal variance of the kernel function, and σ_m is a cha. Hence if we are given training pairs $(\mathbf{X_t}, f_t)$, where $f_t = y_{sim,t}$, we can make predictions (f_e) on test set $\mathbf{X_e}$ by considering the following joint distribution:

$$\begin{bmatrix} f \\ f_e \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{X_t}, \mathbf{X_t}) & K(\mathbf{X_t}, \mathbf{X_e}) \\ K(\mathbf{X_e}, \mathbf{X_t}) & K(\mathbf{X_e}, \mathbf{X_e}) \end{bmatrix}\right) \tag{7}$$

Here, K represents the Kernel matrix. Thus, we can make predictions f_e as follows [23]:

$$\bar{f_e} = K(\mathbf{X_e}, \mathbf{X_t})[K(\mathbf{X_t}, \mathbf{X_t})]f_t \tag{8}$$

$$cov(f_e) = K(\mathbf{X_e}, \mathbf{X_e})K(\mathbf{X_e}, \mathbf{X_t})[K(\mathbf{X_t}, \mathbf{X_t})]^{-1}f_t \tag{9}$$

Hence $y_{pred,GP} = f_e$ denotes the predictions of GP regression at a given test point $(\mathbf{X_e})$, and $\sigma_{gauss} = cov(f_e)$ denotes the confidence interval at that point. The details on formulation of GP regression is available in other literature [23]. The GP regression tool in MATLAB [29] is used to implement GP regression in this analysis. The regression tool determines the hyper-parameters σ_f and σ_m by minimizing the negative log marginal likelihood, which can be expressed as [23]:

$$\log p(f_t|\mathbf{X_t}) = -\frac{1}{2}K(\mathbf{X_t}, \mathbf{X_t})^T f_t - \frac{1}{2}\log K(\mathbf{X_t}, \mathbf{X_t}) - \frac{N}{2}log 2\pi \tag{10}$$

2.3 Multivariate Linear Regression

Multivariate linear regression method is a simple regression method, and is currently employed by energy practitioners to model energy consumption/demand. The linear regression model is defined by the weight/coefficient vector $\omega_{\mathbf{MLR}} = [\omega_1,\ \omega_2 ... \omega_J]$, where J is the total number of features selected for the model. The model tries to minimize the sum of squared error, which can be expressed as following cost function:

$$C(\omega) = \frac{1}{2}\sum_i (y_{sim,t} - \omega_{MLR}^T(\mathbf{X_t})^2) \tag{11}$$

The weight vector is determined as follows:

$$\omega_{\mathbf{MLR}} = (\mathbf{X_t}^T\mathbf{X_t})^{-1}\mathbf{X_t}^T y_{sim,t} \tag{12}$$

The predictions from the linear regression model can be expressed as:

$$y_{MLR,ridge} = \omega_{MLR}^T \mathbf{X_e} \tag{13}$$

2.4 Ridge Regression

Ridge regression is one of the simplest algorithms that can be easily formulated and solved, and can account for non-linearlity by expanding the feature space of feature variables. When the feature space is expanded, the cost function that is minimized in ridge regression can be expressed as:

$$C(\omega) = \frac{1}{2}\sum_i (y_{sim,t} - \omega^T\phi(\mathbf{X_t})^2 + \frac{1}{2}\lambda||\omega||^2 \tag{14}$$

The weight vector can be found by taking derivative of C and setting it to zero. The final expression for the weight vector can be obtained as:

$$\omega = \phi(\phi^T\phi + \lambda\mathbf{I_n})^{-1} y_{sim,t} \tag{15}$$

The regression process in this particular study accounted for the non-linearities by employing a quadratic kernel. This means that the feature space is transformed into an expanded feature space $\phi(\mathbf{X_t}$ where $\phi(\mathbf{X_t})$ is the set of all monomials with a maximum exponent of 2 such that $\phi(\mathbf{X_t} = [1, X_{t,1}, X_{t,2}, ..., X_{t,J}, ...X_{t,1}X_{t,2}, ...X_{t,1}^2, X_{t,2}^2, ..., X_{t,J}^2]$,

Predictions for the test set can be made using the following expression:

$$y_{pred,ridge} = \omega^T\phi(\mathbf{X_e}) \tag{16}$$

Table 2. Details of building types used to simulate fuel consumption values in EnergyPlus

Small Office	**Supermarket**	**Restaurant**
Single-Story, five-zone building	Single-Story, five-zone building	Single-story, two-zone building
Area = 511 m^2	Area = 4,181 m^2	Area = 511 m^2
Mass walls, Attic roof, slab-in-grade floor	Mass walls, built-up-flat roof, slabe-on-grade floor	Steel-frame wall, attic roof, slab-on-grade floor
Window to wall ratio = 21.2%	Window to wall ratio = 10.9%	Window to wall ratio = 17.1%

2.5 Cross-validation

Cross-validation (CV) is a useful technique that allows us to estimate how a given ML algorithm will perform on a test set by only using training data. It also allows us to make modeling choices, e.g. choose optimal hyper-parameters, that minimizes the prediction error of a the ML algorithm. k-fold cross-validation is a type of CV, where the training data is divided into k subsets. The ML algorithm is applied over k epochs, and at each epoch, one of the k subsets is ‘held out’ as a test set. This means that at a single different subset is taken as a test set, and the remaining (k-1) subsets are used as training sets. Thus at each epoch, the ML algorithm is trained using (k-1) training pairs, and tested on the corresponding ’held out’ test set, and the corresponding prediction error is stored. Once all k epochs are exhausted, the cross-validation error (e_{CV}) is reported as the mean of the k prediction errors.

Cross-validation is a useful technique to optimize hyper-parameters in a given ML algorithm, as well as estimate how well the algorithm might do in a prediction set. The ML scheme described in section 4.2 uses a 10-fold cross-validation scheme. Besides giving an estimate of the prediction error on the actual test set, the cross-validation method allowed us to select the optimal subset of input features. Details of feature selection method are provided in section 2.3.

3 Model Description

3.1 Data Generation and Pre-processing

The machine learning scheme applies a combination of neural network and Gaussian process (GP) regression to make time-series predictions for each month, and uses weather and schedule variables as inputs or features. We analyzed the performance of the proposed scheme for hourly fuel consumption profiles in small office, supermarket and restaurant at four different climate zones. Table 2 presents the details of building types that were used to simulate fuel consumption values in EnergyPlus, on which the ML modeling framework was applied. Table 3 lists the locations for which the ML method was applied, whereas table 4 details all the weather and schedule variables considered for analysis. The climate zones were selected such that the ML scheme can be tested for a diverse range of weather patterns, as observed from the annual heating-degree days (HDD) and cooling-degree days (CDD) for each location in 2013.

Both the schedule and weather variables in the training and test sets are normalized such that they are scaled in the order of $\sim O(1)$. The weather and schedule variables are normalized using the maximum and minimum values of the corresponding variable in the training year, as expressed below:

Table 3. List of location and corresponding climate Zones for which ML regression method was applied. The annual heating degree days (HDD) and cooling degree days (CDD) were calculated for training year 2013 using 18.3 C (65 F) as reference temperature

Location	Climate Zone	Latitude	Longitude	Annual HDD	Annual CDD
Hill City, MN	6A	46° 59' N	93° 35' W	4281	424
Olympus, UT	5B	40° 39' N	111° 46' W	3520	424
Phoenix, AZ	2B	33° 27' N	112° 04' W	1084	1522
Baltimore, MD	4A	39° 17' N	70° 26' W	2775	544

Table 4. List of weather variables and schedule variables used for predicting fuel consumption in small office, supermarket and restaurant

Weather variables	Schedule variables		
	Small Office	Supermarket	Restaurant
Dry-bulb temperature Relative Humidity Wind-speed Precipitation Direct-normal Irradiation	Equipment Schedule Lighting Schedule Occupancy Schedule Water Heater Schedule Infilitration Schedule Multiple binary variables	Equipment Schedule Lighting Schedule Occupancy Schedule Water Heater Schedule Infiltration Schedule Freezer/Deli schedules (5) Binary variables (2)	Equipment Schedule Lighting Schedule Occupancy Schedule Water Heater Schedule Infiltration Schedule Gas Equipment Schedule Kitchen Schedules (4) Binary Schedule (2)

$$w_p \leftarrow \frac{w_p - min(w_p)}{max(w_{p,t}) - min(w_{p,t})} \tag{17}$$

$$v_q \leftarrow \frac{v_q - min(v_q)}{max(v_{q,t}) - min(v_{q,t})} \tag{18}$$

Here $max(w_{p,t})$ and $min(w_{p,t})$ are maximum and minimum values of a given weather variable w_p in the training year 2013, and likewise, $max(v_{q,t})$ and $min(v_{q,t})$ represent the maximum and minimum values of a given schedule variable $v_{q,t}$ in 2013. We used actual weather data for years 2013 and 2014 obtained from Mesowest web portal: mesowest.utah.edu [30], and used default values in EnergyPlus for schedule variables. The binary variables (in table 2) refer to schedule variables that can take either of two states ('on' or 'off'). Examples of schedules which are, or can be can be converted to binary schedules after normalization are: HVAC operation schedule, heating temperature schedule, cooling temperature schedule, etc. When two schedules are identical, they are merged into a single binary variable.

We generated the target values using EnergyPlus, on which we applied the ML regression scheme [6]. EnergyPlus is an energy simulation package containing several physics-based modules that collectively calculate the heating and cooling loads of building based on heat and mass balances [6]. EnergyPlus is an open-source tool developed and supported by the U.S. Department of Energy, and it combines the capabilities of BLAST and DOE-2, along with new features [31].

EnergyPlus was used to generate the training and the training targets for the following reasons: (a) It allows us to generate fuel consumption data for multiple commercial building types under a diverse range of climate zones, thereby allowing us to investigate the robustness of the proposed model across multiple building types and climate zones and (b) it allows us to investigate the robustness of the model due to shift in schedule variables from the training to the test set. Thus, using EnergyPlus to generate fuel consumption data allows us to test the prediction accuracies of the ML framework for a given building type across multiple climate zones. This approach has been mentioned and used in previous literature pertaining to prediction of building energy consumption [16].

3.2 Modeling Framework

The modeling framework uses static neural network and Gaussian processes to make one-year ahead predictions at one-hour resolutions. To select suitable machine learning algorithms to be included as part of the overall model, the performances of several machine learning algorithms, including static neural networks, NARX, Gaussian process regression, ridge regression and multivariate linear regression were compared. It was discussed in the section 1 that previous studies showed static and autoregressive neural networks are quite accurate in making short-term predictions of energy consumption, and so they are considered as potential candidates as suitable machine learning algorithms for making long-term predictions as well. Gaussian process regression is selected as a potential candidate as it can produce a bounded interval that has a high probability of encapsulating the actual fuel consumption profile. The values of the covariance bounds suggested by GP regression is specific to each test point for the prediction year 2014. multivariate linear regression is currently used in practice to predict building consumption, and is included as part of ASHRAE protocol [25]. Ridge regression is a simple, quick algorithm that can account for non-linearity in fuel consumption profile using a higher-order kernel. As such, both multivariate linear regression and ridge regression are also considered as suitable ML algorithms.

Section 4.1 compares the performance of the aforementioned machine learning algorithms. The analysis shows that static NN and GP regression performs comparatively better than the other three ML algorithms. Thus, these two algorithms are included as part of the overall ML modeling framework

Both the training and test data are segregated by months, and as such, for a given month, the ML algorithm is trained using feature and target sets for year 2013, and tested using feature set for 2014. The scheme uses a combination of feature ranking and embedded forward selection methods [32]. The feature ranking method assigns a score to each weather and schedule variable by estimating the degree of its contribution to the fuel consumption profile, whereas the forward selection method finds the optimal subset of weather and schedule variables that is likely to yield a high prediction accuracy. The details of feature selection are provided in section 2.3.

The overall scheme can can be summarized as follows:

- **Step 1:** For a given month, initialize all weather variables $\mathbf{w} = [w_1, w_2, ...w_P]$ and schedule variables $\mathbf{v} = [v_1, v_2, v_3, ...v_Q]$ for all hours for both training and test data. Thus, for the month of January, for both training and test weather sets, the weather data would be a matrices of size [744 × P], corresponding to 744 hourly data points and P weather variables. Likewise, the schedule data would be matrices of size [744 × Q].

- **Step 2:** Compute the Pearson coefficients of all weather variables ($\mathbf{R_w}$) and schedule variables ($\mathbf{R_v}$) for the training data. The Pearson coefficient is a measure of the linear correlation between a given variable and the output, and is used as a metric for feature ranking in this analysis.

- **Step 3:** Sort the weather and schedule variable sets separately in descending order with respect to their Pearson coefficients obtained in the previous step, and order the weather variables $\mathbf{w}$ and schedule variables $\mathbf{v}$ in the same order - both for training and test set. Thus, the new matrices are $\mathbf{w_t'}$ and $\mathbf{v_t'}$ for the training set and $\mathbf{w_e'}$ and $\mathbf{v_e'}$ for the test set, such that $w_{t,1}$ and $v_{t,2}$ have the highest Pearson coefficients among the weather and schedule variables respectively in the training set.

- **Step 4:** Apply forward selection (as detailed in section 2.3) using neural network and Gaussian process regression model to select the subset of weather and schedule variables that minimize the cross-validation (CV) error. For feature selection, only relative errors are of interest, and so for neural networks, only 10 epochs are used to improve computational efficiency. Here, one epoch means one pass of the machine learning algorithm over the entire training data. Thus the forward selection would provide the training and test feature sets $\mathbf{X_t}$ and $\mathbf{X_e}$. Both the training and the test feature sets are sized $[N \times J]$, where N is total number of observations/data points in the feature set, J is the total number of 'relevant' variables, i.e. optimal subset of input variables corresponding to minimum CV error in the forward selection method.

- **Step 5:** Apply static neural network (NN) for 100 epochs, using the subset of weather and schedule variables found in the previous step. The neural network is trained using the input-output pair $(\mathbf{X_t}, y_{sim.t})$ and tested using the feature set $\mathbf{X_e}$. Record the predictions on the test set $y_{pred,NN}$ the cross-validation error, $e_{CV,NN}$. The value of $e_{CV,NN}$ reported is the minimum of cross-validation errors obtained over 100 epochs.

- **Step 6:** Apply Gaussian process (GP) regression using the same feature sets $\mathbf{X_t}$ and $\mathbf{X_e}$ and training target values $y_{sim,t}$ to compute the GP prediction $y_{pred,GP}$, corresponding cross-validation error ($e_{CV,GP}$).

- **Step 7:** Compare the relative CV errors of NN and GP. If $e_{CV,NN} < e_{CV,GP}$, assign the final prediction to be $y_{pred,NN}$, else assign the final prediction to be $y_{pred,GP}$.

- **Step 8:** Determine the RMS error e_{rms} to represent point accuracy and the covariance error $e_{cov,rms}$ to represent range accuracy - the latter applicable to Gaussian process regression only.

  For cases where NARX, ridge regression and linear regression were reapplied, the final prediction in step 7 was assigned to be the prediction of the ML algorithm with the minimum cross-validation accuracy.

  A schematic diagram of the static neural network is presented in figure 1.

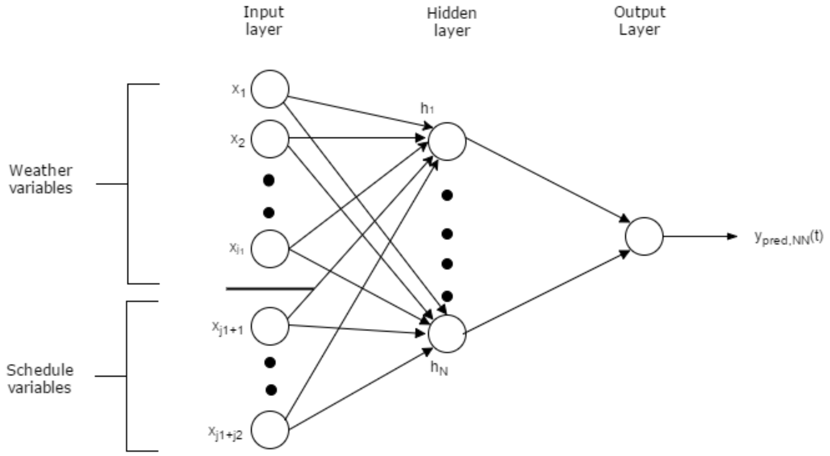

Figure 1. Static Network to compute fuel consumption at hourly intervals

Algorithm 1 Description of ML scheme

```
Set $\mathbf{w} = [w_1, w_2, ....w_P]$ and $\mathbf{v} = [v_1, v_2, ....v_Q]$ for both training $(\mathbf{w_t}, \mathbf{v_t})$and test feature sets $(\mathbf{w_e}, \mathbf{v_e})$.
Compute $\mathbf{R_w} = [R_{w.1}, R_{w.2}, ....R_{w,P}]$ and $\mathbf{R_v} = [R_{v,1}, R_{v,2}, ....R_{v,Q}]$ for training feature set.
$[\mathbf{R_w}, idx_w] \leftarrow sort(\mathbf{R_w})$ , $[\mathbf{R_v}, idx_v] \leftarrow sort(R_w)$ for training feature set.
$\mathbf{w}' \leftarrow Order(\mathbf{w}, idx_w)$ and $\mathbf{v}' \leftarrow Order(\mathbf{v}, idx_v)$ for both training $(\mathbf{w'_t}, \mathbf{v'_t})$and test feature sets $(\mathbf{w'_e}, \mathbf{v'_e})$.
Apply forward selection with chosen ML algorithm(s) to find the optimal feature sets $\mathbf{X_t}$ and $\mathbf{X_e}$.
for epoch = 1,2....100 do
    Apply static NN: train using $[\mathbf{X_t}, y_{sim,t}]$ and test on $\mathbf{X_e}$ to make predictions $y_{pred,NN}$
    Find cross-validation error in each epoch. $e(epoch)$
end for
Determine $e_{CV,NN} \leftarrow min(e)$.
Apply GP regression to determine $y_{pred,GP}$, predictive variance $\sigma^2_{gauss}$ and cross-validation error $e_{CV,GP}$
Compare ($e_{CV,NN}$, $e_{CV.GP}$). If $e_{CV,NN} < e_{CV.GP}$, $y_{pred} \leftarrow y_{pred,NN}$, else $y_{pred} \leftarrow y_{pred,GP}$
Report $y_{pred}$ and $\sigma_{gauss}$.
```

3.3 Feature Ranking and Selection

As mentioned in section 2.2, the first step is rank the elements within the sets of weather variables ($\mathbf{w_t}$ and $\mathbf{w_e}$) and schedule variables ($\mathbf{v_t}$ and $\mathbf{v_e}$) . This is done using the Pearson coefficient, which can be expressed as:

$$R_w(p) = \frac{cov(\mathbf{w_p}, y_{sim,t})}{\sqrt{cov(\mathbf{w_p}) var(y_{sim,t})}} \tag{19}$$

Here, $R_w(p)$ denotes the Pearson coefficient of a given weather variable p, and cov and var represent the covariance and variance operators respectively. The Pearson coefficients of a generic schedule variable Q can be similarly computed as follows:

$$R_v(q) = \frac{cov(\mathbf{v_q}, y_{sim,t})}{\sqrt{cov(\mathbf{v_p}) var(y_{sim,t})}} \tag{20}$$

The feature ranking using Pearson coefficients are only able to detect linear dependencies, and so it is used in conjunction with a more rigorous embedded method. The embedded method employed is a varaint of the forward selection approach [32], as stated in section 2.2), which uses the ML algorithm itself to determine a nested subset of variables which are likely to be most relevant [32]. Assuming that the ordered sets from the Pearson coefficient criteria are $\mathbf{w'_t} = [w_{t,1}, w_{t,2}...w_{t,P}]$ and $\mathbf{v'_t} = [v_{t,1}, v_{t,2}...v_{t,Q}]$, with P and Q being the maximum dimensions of these sets respectively, the forward selection starts with $w_{t,1}$ and $v_{t,1}$, and uses them as inputs for the static network 1, GP regression or other ML algorithm, before recording the corresponding CV error. Progressively, one entry from each set $\mathbf{w'}$ and $\mathbf{v'}$ are added before all entries in each set are exhausted. The combination of variables yielding the lowest CV error are considered as most relevant and are kept for subsequent steps. Thus, the forward selection algorithm returns the training feature set $\mathbf{X_t}$ and test feature set $\mathbf{X_e}$, each sized $N \times J$. The number of weather and schedule variables in either set are j_1 and j_2 respectively, where $j_1 \leq P$, $j_2 \leq Q$, and $j_1 + j_2 = J$. Therefore, the feature sets $\mathbf{X_t}$ and $\mathbf{X_e}$ are subsets of the original set that minimises the CV error, and hypothetically, is the optimal subset that could potentially minimize the prediction error as well.

The algorithm can be summarized as follows:

Algorithm 2 Description of Feature Selection Algorithm

```
for p = 1.....P do
    for q = 1.....Q do
        Set feature set to $\mathbf{X_t} = [w'_1...w'_p, v'_1, ....v'_Q]$.
        Apply ML regression algorithm (NN or GP) and find the corresponding cross-validation error, $e_{CV}(p, q)$.
    end for
end for
Find $(j_1, j_2) = \arg\min_{(p,q)} e_{CV}(p, q)$
Return $X_t$ and $X_e$ where the number of weather and schedule variables are $j_1$ and $j_2$.
```

We will now demonstrate this process with an example: table 3.3 shows the matrix of neural network cross-validation errors ($e_{NN,CV}$) obtained for different combinations of $\mathbf{w'_t}$ and $\mathbf{v'_t}$. The ordered weather and schedule variable sets post feature-ranking are $\mathbf{w'_t} = [w'_{t,1}, w'_{t,2}, ..., w'_{t,5}]$ and $\mathbf{v'_t} = [v'_{t,1}, v'_{t,2}, ..., v'_{t,7}]$; where $[w'_{t,1}, w'_{t,2}, ..., w'_{t,5}]$ denotes [dry-bulb temperature, relative humidity, direct-normal radiation, wind-speed and precipitation] and $[v'_{t,1}, v'_{t,2}, ..., v'_{t,7}]$ correspond to [binary schedule 1, infiltration schedule, binary schedule 2, lighting schedule, water heater schedule, occupancy schedule and equipment schedule]. Once the entries in weather variable and schedule variable sets are sorted, we notice that the minimum cross-validation error is obtained when $\mathbf{w'_t}$

Table 5. Matrix of $e_{CV,NN}$ obtained for January fuel consumption profile in Baltimore. Each row represents an additional weather variable added to $\mathbf{w'_t}$ and each column represents an additional schedule variable to $\mathbf{v'_t}$. Thus, the minimum CV error for this example is obtained when $\mathbf{w'_t}$ = $[w_{t,1}]$ and $\mathbf{v'_t}$ = $[v'_{t,1}, v'_{t,2}, ..., v'_{t,7}]$, where $e_{CV,NN}$ = 0.2497.

	$v'_{t,1}$	$v'_{t,2}$	$v'_{t,3}$	$v'_{t,4}$	$v'_{t,5}$	$v'_{t,6}$	$v'_{t,7}$
$w'_{t,1}$	0.422	0.4135	0.3603	0.3146	0.3133	0.2635	**0.2497**
$w'_{t,2}$	0.3837	0.3811	0.3405	0.3120	0.3127	0.2635	0.2497
$w'_{t,3}$	0.3944	0.3808	0.3525	0.3118	0.3049	0.2728	0.2709
$w'_{t,4}$	0.3815	0.3843	0.3529	0.3096	0.3251	0.2854	0.2708
$w'_{t,5}$	0.3685	0.3604	0.3222	0.2859	0.2981	0.2543	0.2604

= $[w_{t,1}]$ and $\mathbf{v'_t}$ = $[v'_{t,1}, v'_{t,2}, ..., v'_{t,7}]$. So the training and test feature sets are assigned to be: $\mathbf{X_t} = [w'_{t,1}, v'_{t,1}, v'_{t,2}, ..., v'_{t,7}]$ and $\mathbf{X_e} = [w'_{t,1}, v'_{t,1}, v'_{t,2}, ..., v'_{t,7}]$, such that j_1 = 1, j_2 = 7 and J = 8.

3.3.1 Evolutionary Polynomial Regression as a benchmark model

The feature selection method proposed in Algorithm 2 is compared with a benchmark Evolutionary Polynomial Regression (EPR) model [33]. Evolutionary polynomial algorithm is an integrated regression framework that combines numerical regression with a genetic algorithm-based feature selection method.

In EPR [33], the target vector y can be expressed as follows:

$$\mathbf{y_{N\times 1}} = [\mathbf{I_{N\times 1}}\ \phi_{\mathbf{N\times M}}] \times [\beta_0\ \beta_1\ ...\ \beta_m]^T \tag{21}$$

Here $\mathbf{y}$ is the target vector containing N data points, $\mathbf{I}$ is a vector of unit values, and $\beta = [\beta_0\ \beta_1\ ...\ \beta_m]$ are regression coefficients obtained using least squares. ϕ is the expanded feature space from original features $\mathbf{X_t}$, and can be expressed as:

$$\phi = \phi_{im} = x_{t,i1}^{ES(m,1)} x_{t,i2}^{ES(m,2)} ....x_{t,iJ}^{ES(m,J)} = \prod_{j=1}^{J} x_{t,ij}^{ES(m,J)} \tag{22}$$

Here m is the index of the expanded feature in feature space ϕ and $\mathbf{ES}$ is a matrix of exponents where the entry $ES(m,j)$ corresponds to the exponent of original feature $\mathbf{X_{t,j}}$ inside the m^{th} term in ϕ. Thus, EPR works on the basis of optimizing the matrix of exponents $\mathbf{ES}$ using a genetic algorithm and determining the regression coefficients β using least squares. In this paper, the genetic algorithm was implemented using MATLAB. Details of the EPR algorithm can be obtained in [33]. In this study, the entries $ES(m,j)$ are limited to integer values $[0,1,2]$. Thus, the EPR algorithm serves as a feature selection method besides a regression method - as features with zero exponent in all terms in an expression can be considered redundant.

3.4 Model Optimization

The overall modeling framework presented in section 3.2 requires two key modeling choices: (i) selection of machine learning algorithms and (ii) selection of the subset of input variables/features. The process of selection of ML algorithms is detailed in sections 3.2 and 4.1. The subset of relevant input variables is selected using the feature selection method, as demonstrated in section 3.3.

Table 6. List of model parameters and hyper-parameters for each ML algorithm in this analysis.

ML Algorithm	Model Parameters	Hyper-parameters
Static NN	$\omega_{\mathbf{NN}}$	H
NARX	$\omega_{\mathbf{NARX}}$	H, τ
GP regression	$\bar{f}_e$, σ_{gauss}	b
Ridge Regression	ω_{ridge}	λ
MLR	ω_{MLR}	—

The parameters pertaining to each individual machine learning algorithm also need to be optimized. These parameters can be grouped into two categories: standard model parameters and hyper-parameters. The model parameters define the state of a given ML model and can be learned from the data. Hyper-parameters, on the other hand, are relatively higher-level modeling choices, that are not learned directly from the data, are either learned through cross-validation or empirical relations. Table 3.4 refers to the model parameters and hyper-parameters for each ML model.

The neural network weight vectors $\omega_{\mathbf{NN}}$ and $\omega_{\mathbf{NARX}}$ define the static NN and NARX models respectively, and are learned through the back-propagation algorithm, as described in section 2.1. The number of nodes in the hidden layer, H is a hyper-parameter for both static NN and NARX networks. As mentioned in section 2.1, the optimal number of hidden nodes for a neural network with one hidden layer is determined as $H = [\frac{N}{|\mathbf{X}|\log(N)}]^{1/2}$ [27], where $|\mathbf{X}|$ is the total number of input features. For NARX models, the hyper-parameter τ signifies the number of prior time-steps for which lagged outputs are considered as features, and is determined using cross-validation. The weights vectors/coefficients $\omega_{\mathbf{MLR}}$ and $\omega_{\mathbf{ridge}}$ are model parameters that define the state of multivariate linear regression and ridge regression respectively. In case of ridge regression, the hyper-parameter λ is obtained using 10-fold cross-validation, by performing a grid search within the interval $[10^{-4}, 10^{4}]$.

As mentioned in section 2.2, Gaussian Process regression is slightly different from the other ML algorithms mentioned in this study. Rather than being defined by a set of weights or other model parameters, GP make inferences directly in the function-space [23]. The covariance matrix that determines the mean and covariance function is modeled using a squared exponential kernel, which contains the hyper-parameters σ_f and σ_m. The Gaussian process regression tool in MATLAB internally these hyper-parameters by minimizing the negative log marginal likelihood function [29], which is presented in equation 10.

It should be noted that for a particular building type in a climate zone, the learned weights and hyper-parameters for a ML algorithm are specific to a given month only. Considering fuel consumption predictions of the aforementioned ML algorithms for a small office in Hill City, MN as example, the hyper-parameters are listed in table 3.4.

4 Results and Discussion

Figures 2-23 show how the machine learning algorithms perform across different building types and climate locations. As mentioned previously, the ML models are trained separately for each month using hourly data sets for year 2013, as long as the mean squared value of the hourly heating load was at least 5% of the peak hourly load in that year. The predictions were subsequently done for hourly fuel consumption in the corresponding months for the year 2014. The following accuracy

Table 7. Values of hyper-parameters for each ML algorithm corresponding to fuel consumption predictions in a small office in Hill City, January.

ML Algorithm	Hyper-parameters
Static NN	$H = 4$
NARX	$H = 3$, $\tau = 5$
GP regression	$\sigma_m = 0.216$, $\sigma_f = 0.269$
Ridge Regression	$\lambda = 12.38$
MLR	—

metrics were used to quantitatively evaluate the performance of ML models.

The transient hourly error (e_t) can be expressed as:

$$e_t = y_{sim,e} - y_{pred} \tag{23}$$

The relative mean squared error (e_{rms}) can be expressed as:

$$e_{rms} = \frac{\sqrt{\sum_{i=1}^{T} e_t^2}}{\sqrt{\sum_{i=1}^{T} y_{sim,e}^2}} \tag{24}$$

For Gaussian process (GP) regression, an error associated with its predictive covariane can also be defined, such that the predictions are only penalized if they are outside the GP-suggested covariance bounds.

$$e_{cov,rms} = \begin{cases} \frac{\sqrt{\sum_{i=1}^{T}(y_{sim,e}-(y_{pred,GP}+\sigma_{gauss}))^2}}{\sqrt{\sum_{i=1}^{T} y_{sim,e}^2}} & \text{if } y_{sim,e} > y_{pred,GP} + \sigma_{gauss} \\ \frac{\sqrt{\sum_{i=1}^{T}(y_{sim,e}-(y_{pred,GP}-\sigma_{gauss}))^2}}{\sqrt{\sum_{i=1}^{T} y_{sim,e}^2}} & \text{if } y_{sim,e} < y_{pred,GP} + \sigma_{gauss} \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

Finally, the relative monthly error can be computed as follows:

$$e_{month} = \frac{\sum_{i=1}^{T} e_t}{\sum_{i=1}^{T} y_{sim,e}} \tag{26}$$

We will compare the performance of different ML algorithms using the error metrics presented above, and discuss and quantitatively evaluate the role of co-variate shift, i.e. variability in our weather data, in affecting our prediction error. Subsequently we will use the error metrics to quantify the accuracy of the presented ML regression scheme, and analyze how the accuracy varies across different climate zones and building types. Finally, we will discuss the effect of key parameters on the prediction error.

4.1 Sample results and comparison between ML algorithms

Figures 2 - 4 show how the different machine learning algorithms perform on an EnergyPlus-simulated data set for a small office located in Hill City, Minnesota. Figure 2 illustrate how the ML predictions of the hourly fuel consumption for January compare with those obtained using

EnergyPlus. The plots show that in general, both the static neural networks and GP regression follow the EnergyPlus-simulated fuel consumption comparatively better than the NARX network, ridge regression with a second order polynomial kernel, and multivariate linear regression. The fuel consumption is normalized in these plots with respect to its maximum value in a given year. Figure 3 shows how the simulated fuel consumption profile compares with predictions from ML algorithms compare over a 24-hour time period in a typical January day. The figure shows that, for the given day, the static neural network and the GP regression prediction are within 19.61% and 27.44% of the simulated fuel consumption prediction. The GP covariance bounds also encapsulate the simulated load fairly well, with a covariance error of 15.18%. The plot also shows that for the duration of said 24 hours, 41.67% of the data points lie within the Gaussian covariance bounds.

Figure 4 compares the hourly transient error, e_t for static NN and GP regression. The two plots appear to correspond to each other closely, and for both ML algorithms, e_t appears to oscillate about a close-to-zero mean. The latter claim is supported by a small value (<2%) of the relative monthly error (e_{month}), which aggregates e_t over an entire month.

Figure 5 shows how predictions of different ML algorithms compare with each other for a typical day in April for a small office at the same location. The weather and gas consumption profile for a small office in Hill City differs from January to April: the mean of the hourly ambient temperature and the RMS of hourly gas consumption for January are -11.4 C and 83.6 MJ respectively, whereas the corresponding values for April are 7.54 C and 20.7 MJ. The figure re-affirms the claim that static NN and GP regression perform better than NARX, multivariate linear regression and ridge regression - in fact, for the given consumption pattern, the improvement in performance of static NN and GP regression compared to the other algorithms appears to be even greater. The prediction errors (e_{rms}) for NN and GP in April are 37.9% and 36.9% respectively, while the corresponding e_{rms} for NARX, ridge and multivariate regression all exceed 59%. Thus, employing NN or GP is likely to produce more accurate results compared to multivariate regression methods employed by ASHRAE [25], which agrees with suggestions by Heo and Zavala [24]. Figure 6 compares the relative errors of different ML algorithms for each month during the heating period, i.e. months where the RMS of fuel consumption was 5% or more than the yearly peak value. For all months within this heating period,the prediction errors for static NN and GP regression are lower than NARX, ridge regression and multivariate linear regression. The figure also presents the cross-validation error for static NN and GP regression, which is an indicator for in-model performance of the respective ML algorithms.

To verify whether this observation holds for a different fuel consumption pattern, the aforementioned ML algorithms were tested on a supermarket in Phoenix, AZ. Figure 7 shows that for a different fuel consumption profile, static NN and GP regression, in general, perform comparatively better than the other algorithms. Both NARX network and ridge regression require a greater number of input features, and as such, are more prone to over-fitting. On the other hand, multivariate linear regression is likely to be prone to under-fitting, as it does not account for non-linearities in the target function, i.e. simulated fuel consumption from EnergyPlus). As static NN and GP regression provide a compromise between generalization and function expressiveness and consistently perform better than NARX model, ridge regression and multi-regression, static NN and GP regression will be considered for further analysis.

Figures 6 and 7 also show that the feature selection method presented in section 3.3, in general, performs better than the EPR algorithm. This could be because of one or both of the following reasons: (i) The performance of the EPR algorithm might have dropped with increasing number of features and (ii) The EPR algorithm does not treat the weather and the schedule variables in a

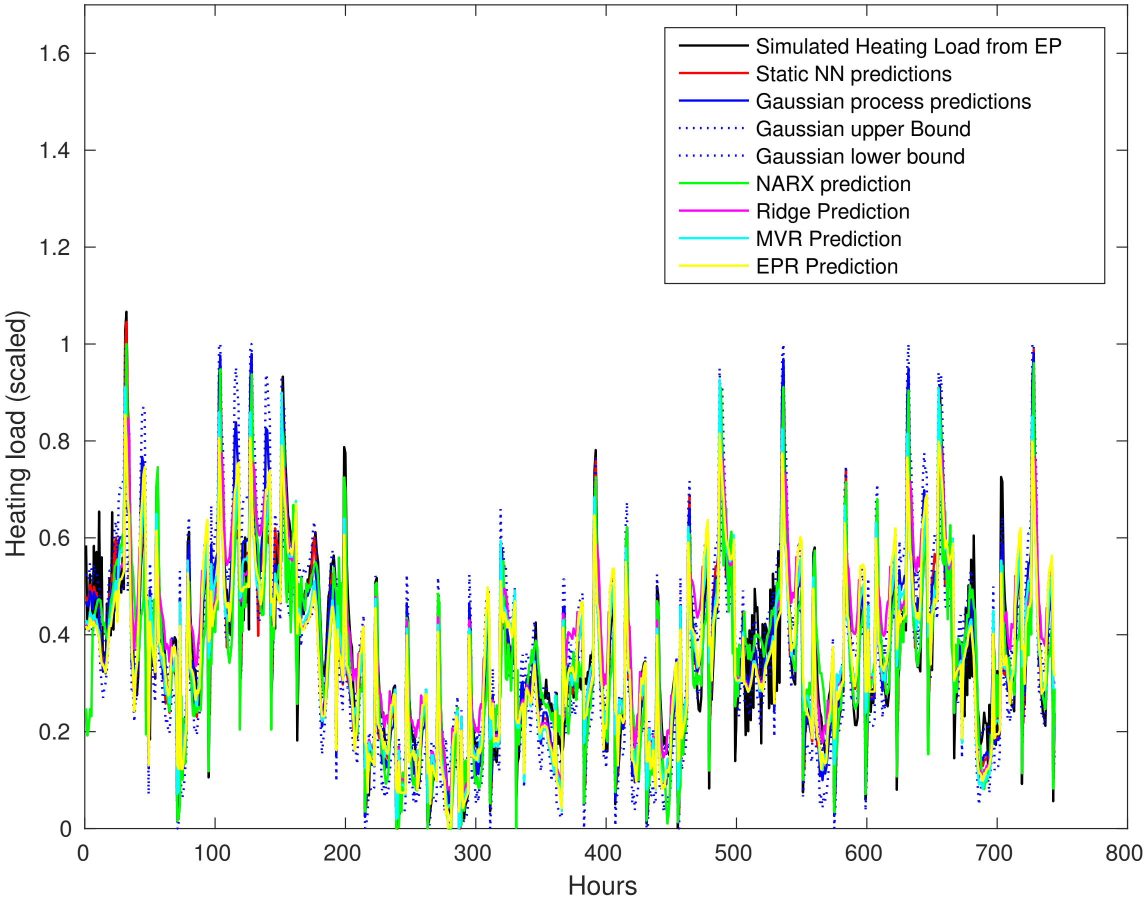

Figure 2. Hourly fuel consumption predictions of ML algorithms including: NN, NARX, GP regression, ridge regression and multivariate linear regression for a small office in Hill City, MN during January 2014. The interval between $t = [97, 120]$ hours is presented in detail in figure 3.

segregated manner.

4.2 Effect of weather variability between training and test sets

So far, it has been assumed that the schedule variables do not change from training feature set to test feature set, and so the discrepancies in distributions of inputs would occur only for weather data between the two feature sets. However, each weather variable does not contribute equally to the target function, and an approximation of which weather variables are important and to what extent, can be determined using the Pearson coefficient criterion detailed in section 3.3. Variability in weather from training to test set causes co-variate shift, which is simply a discrepancy in distribution between the training weather data $\mathbf{w_t}$ and test weather data $\mathbf{w_e}$ [34].

The January weather data and corresponding fuel consumption data for a small office in Hill City can be used as an example to illustrate the effect of covariate shift on the prediction accuracies. From

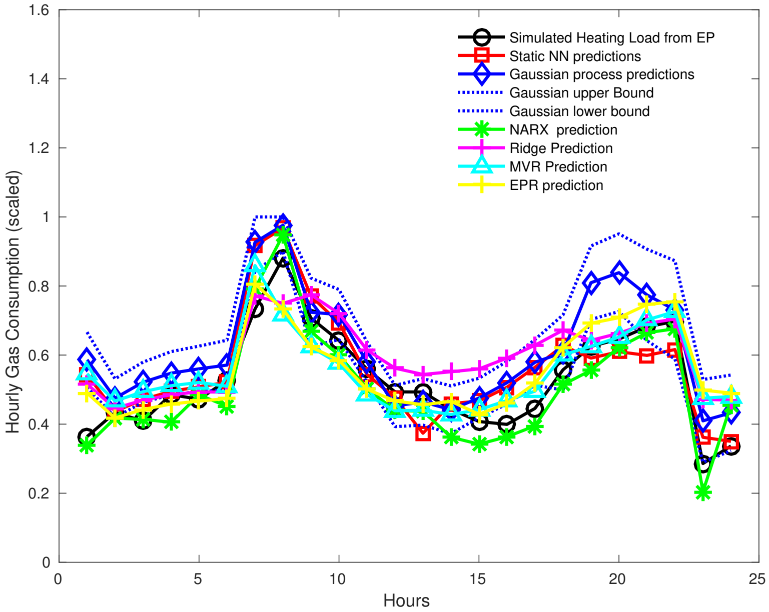

Figure 3. Hourly fuel consumption predictions of different machine learning algorithms over a time period of 24 hours for a small office in Hill City, MN during January 5, 2014. The 24-hour profile in this figure corresponds to the profile within the segmented interval in figure 2.

the forward selection method in the feature selection procedure using static NN, it was determined that the lowest cross-validation error was obtained when the subset of input variables were: dry-bulb temperature (weather variable); equipment schedule, lighting schedule, occupancy schedule, water heater schedule, and infiltration schedule (schedule variable). Since dry-bulb temperature is by far the most significant weather variable, the target and the predicted values of fuel consumption are plotted as a function of dry-bulb temperature (figure 8).

Figure 8 illustrates an example of simple covariate shift, where the relationship between the fuel consumption and the relevant subset of inputs do not change, but the prior distribution of the inputs change from training to test set. In figure 8, the red domain represents the test set and the black domain the training set. The non-dimensional mean values of the training and the test try-bulb temperatures are: $\mu_t = 0.289$ (corresponding to -11.4 C), $\mu_e = 0.2058$ (-17.0 C), whereas the corresponding variances are $\sigma_t = 0.0113$ (7.27 C) and $\sigma_e = 0.0177$ (9.09 C). Figure 9 provides an illustration of a case where the combination of Pearson coefficient feature ranking and forward feature selection picks two weather variables. For predicting fuel consumption in December at a small office in Phoenix AZ, the relevant subset of weather variables were found to be dry-bulb temperature and relative humidity. The mean values corresponding to the training and test set were found to be: $\mu_t = (x, y) = (T_{db}, RH) = (0.479, 0.284)$ (corresponding to $T_{db} = 9.97$ C and $RH = 35.4\%$) and $\mu_e = (0.495, 0.474)$ (corresponding to 10.4 C and 52.2%), whereas the corresponding standard deviations are: $\sigma_t = (0.1746, 0.221)$ (corresponding to 5.33 C and 19.6%) and $\sigma_e = (0.193, 0.260)$ (corresponding to 4.83 C and 23.2%). The positive shift in mean relative humidity is the predominating change from the training to the test.

It is hypothesized that the discrepancies in prediction errors can be explained using similarity

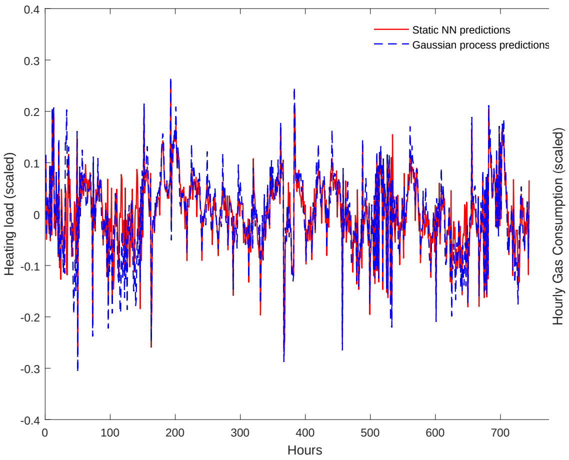

Figure 4. Transient hourly error for static neural network and Gaussian process regression in predicting hourly fuel consumption in January

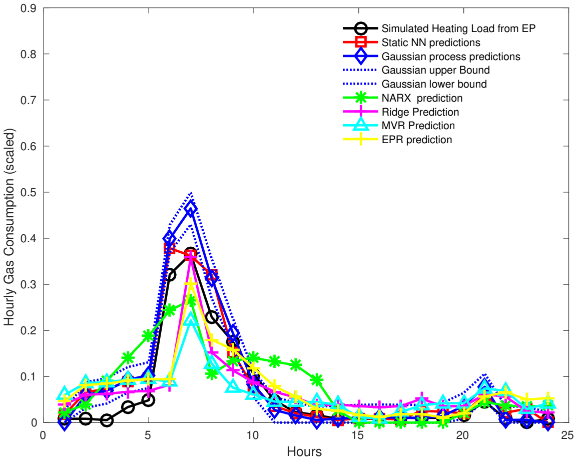

Figure 5. Fuel consumption predictions of different machine learning algorithms over a time period of 24 hours for April 5, 2014

scores that describe that variability of weather data between the training set and the test set and the variance of weather data within the training set itself. As such, a variant of the squared exponential function can be used to find the similarity matrix $(K(n_1, n_2))$.

$$K(n_1, n_2) = \exp\left(- \sum_{j=1}^{j_1} \frac{\rho_w(j)\ |w_{t:n_1,j} - w_{e:n_2,j}|^2}{\sqrt{\sum_{i=1}^{N} y_{sim,t}^2}/N} \right) \tag{27}$$

$\rho_w(j)$ is the normalized Pearson coefficient of relevant weather variable j, and can be expressed as:

$$\rho_w(j) = \frac{R_w(j)}{\sum_{j=1}^{j_1} R_w(j)} \tag{28}$$

Here n_1 and n_2 are generic data points within the training and test feature sets $\mathbf{w_t}$ and $\mathbf{w_e}$ respectively, and j_1 is the total number of relevant weather variables. The term in the denominator of equation 27 is the root mean square of the target values in the training set. This term ensures that when the average fuel consumption is relatively lower in a given month, the entries are penalized comparatively more for being dissimilar. The normalized Pearson coefficients $\rho_w(j)$ indicate the extent to which a given weather variable j could potentially affect the hourly gas consumption profile.

The similarity between the training and the test weather data$(\mathbf{w_t}, \mathbf{w_e})$ can be expressed as follows:

$$s(\mathbf{w_t}, \mathbf{w_e}) = \sum_{n_1=1}^{N_1} \sum_{n_2=1}^{N_2} \frac{K(n_1, n_2)}{N_1 N_2} \tag{29}$$

Here N_1 and N_2 are the number of observations in sets w_t and w_e respectively, and the term

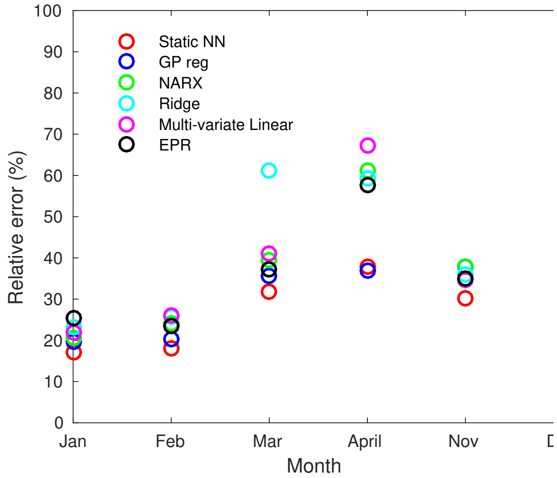

Figure 6. Relative errors of Static neural network, Gaussian Process Regression, NARX network, ridge regression and multivariate linear regression in predicting hourly fuel consumption for a small office in Hill City, MN.

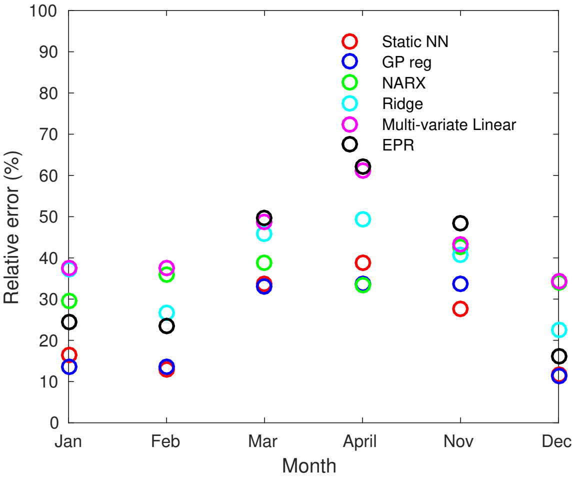

Figure 7. Relative errors of Static neural network, Gaussian Process Regression, NARX network, ridge regression and multivariate linear regression in predicting hourly fuel consumption for a supermarket in Phoenix, AZ.

$N_1 N_2$ is introduced to bound the similarity values between [0, 1]. Similarly a term $s(\mathbf{w_t}, \mathbf{w_t})$ could be similarly computed to quantify the variance within the training weather data itself.

Figures 10 and 11 show that the NN prediction error shows a negative correlation with respect to both $s(\mathbf{w_t}, \mathbf{w_e})$ and $s(\mathbf{w_t}, \mathbf{w_t})$. The plots show that 90% of all data points with values of both $s(\mathbf{w_t}, \mathbf{w_e})$ and $s(\mathbf{w_t}, \mathbf{w_t})$ greater 0.6 have a relative error of 10% or lower. Thus, the similarity scores provide a way to generalize discrepancies in prediction errors across multiple climate zones and building types.

4.3 Performance of ML algorithms over different climate zones and building types

Figures 12 - 14 show how the root-mean squared (RMS) average of hourly fuel consumption and RMS of associated absolute errors vary over different months at Hill City, MN for a small office, supermarket and restaurant respectively. As mentioned previously, only months where the RMS of hourly fuel consumption is $\geq$ 5% of the peak hourly fuel consumption was considered for this analysis. The figures 12 and 13 show that, in general for small office and supermarket, the relative error in NN prediction is comparatively higher in 'warmer' months (i.e. months between March to October), but the corresponding absolute errors are still relatively low in these months due to its low fuel consumption.

Figures 15 - 17 show how the absolute values of root-mean squared hourly NN prediction errors vary for different climate zones. The plots reiterate the observations that the colder locations, i.e. Hill City and Baltimore, have higher values of absolute errors. For all climate zones presented in this paper, the maximum errors in fuel consumption for a small office, supermarket and restaurant are 15.7 MJ, 284.3 and 74.0 MJ respectively.

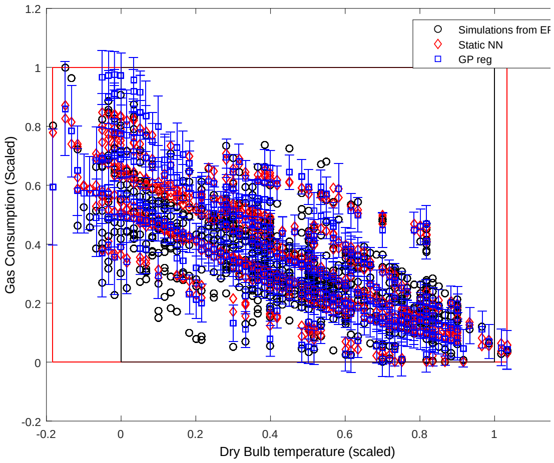

Figure 8. Simulated fuel consumption from EP, static NN predictions and GP regression predictions as a function of dry-bulb temperature for a small office at Hill City, MN during January. Red box indicates training domain, black box indicates test domain

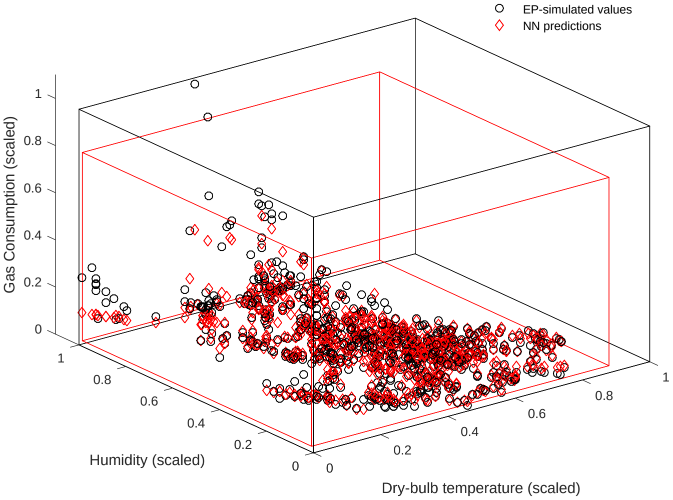

Figure 9. Simulated fuel consumption from EP and static NN predictions as a function of dry-bulb temperature and relative humidity for a small office at Pheonix, AZ during December. Red box indicates training domain, black box indicates test domain

The figures also illustrate that for supermarket and for restaurant, the absolute errors in NN prediction are highest in March for all locations except Phoenix, which is a comparatively much warmer climate zone. The absolute errors corresponding to the small office case in March are also high - for instance, the absolute error for Hill City in March is as high as 14.83 MJ. This is mainly due to the relatively high values of relative error ($e_{rms,NN}$) in predicting transient fuel consumption in March compared to December to February, as presented in figures 18 - 20.

Figures 18-19 also show that the relative errors for all locations follow a similar trend for small office and a supermarket. The maximum value of $e_{rms,NN}$ for the small office case for all locations is 49.3% at Olympus, UT at March and for the supermarket case, is 38.7% at Phoenix, AZ in April. Figures 18-19 also illustrate that the values of $e_{rms,NN}$ during the months of March to November for these two building types are, in general, higher for the comparatively warmer climate zones, i.e. Olympus UT and Phoenix AZ. This could be because the mean fuel consumption at these locations during the months March to November are relatively lower at these two locations compared to the mean fuel consumption at Hill City, MN and Baltimore, MD.

Figure 20 presents the relative errors for restaurant for all climate zones. We can observe that for all locations, the relative errors are lower in magnitude for a restaurant than for small office or supermarket - for instance, the maximum relative error ($e_{rms,NN}$) for a restaurant for all locations is 16.3%, which is lower than the maximum values of $e_{rms,NN}$ obtained for small office and supermarket. The RMS of hourly fuel consumption in a restaurant is greater than 5% of the peak value over the entire year, even during the summer months. This could be because the fuel consumption profile for a restaurant is likely to be more dependent on schedules than in the case of small office or a supermarket. The low prediction errors for the restaurant cases are indicated by the high similarity scores, as observed in figures 10 and 11.

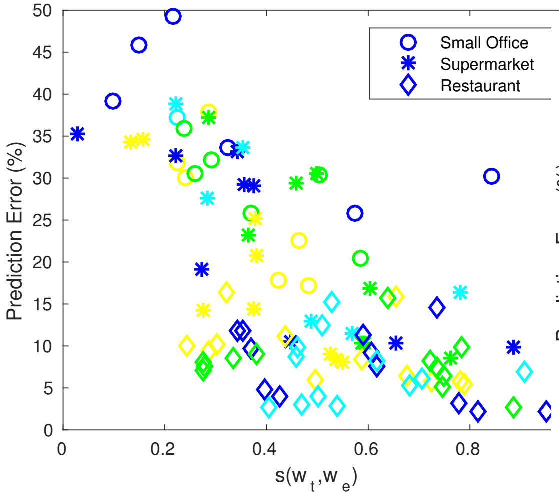

Figure 10. Prediction Errors vs $s(\mathbf{w_t}, \mathbf{w_e})$ - indicates discrepancy between the training weather data and the test weather data

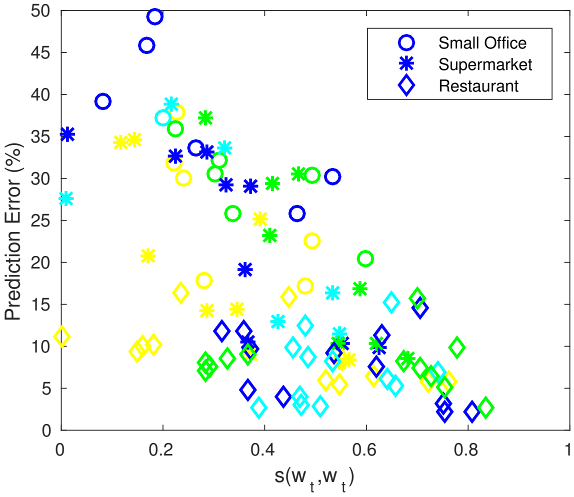

Figure 11. Prediction Errors vs $s(\mathbf{w_t}, \mathbf{w_t})$ - indicates variance within the training weather data

Figure 21 - 23 show how the error in NN predictions of monthly fuel consumption ($e_{month,NN}$) varies over different climate zones. The maximum values of $e_{month,NN}$ corresponding to small office, supermarket and restaurant are 23.5%, 6.78% and 5.50% respectively. As mentioned, e_{month} is an aggregation of the transient hourly error e_t. Since e_t can take both positive an negative values, the monthly error in NN precition is always lower than the RMS error of transient hourly error.

4.4 Effects of Model Parameters

4.4.1 Effect of scaling covariance bounds in Gaussian process regression

Section 2.2 details the theoretical background of Gaussian process regression and suggests that one of its strengths is its ability to make predictions on the test set, as well as provide a covariance function, σ_{gauss} that indicates the uncertainty in prediction at a given test point. The covariance function has a value specific to each test point, and depends on the number of training points available in the proximity of the test point to support the GP prediction. This is in contrast to linear regression, which has a constant standard deviation to indicate the uncertainty in prediction for all test points.

Thus, the GP regression hypothesizes that a high fraction of the target values, f, lies within the bounds $[y_{pred,GP}(\mathbf{X_e}) - \sigma_{gauss}(\mathbf{X_e}), y_{pred,GP}(\mathbf{X_e}) + \sigma_{gauss}(\mathbf{X_e})]$. While the covariance function σ_{gauss} depends on the test points $(\mathbf{X_e})$, σ_{gauss} can be scaled by a constant factor to regulate the confidence levels of the predictions, i.e. regulate the probability f of a target value lying within the GP bounds, $f = p[y_{pred,GP}(\mathbf{X_e}) - \sigma_{gauss}(\mathbf{X_e}) < y_{sim,e} < y_{pred,GP}(\mathbf{X_e}) + \sigma_{gauss}(\mathbf{X_e})]$. Thus, σ_{gauss} can be modified as follows:

$$\sigma_{gauss} \leftarrow (1 + \kappa)(\sigma_{gauss}) \tag{30}$$

Here κ is a constant parameter within the interval [0, 1]. Figure 25 presents how the performance

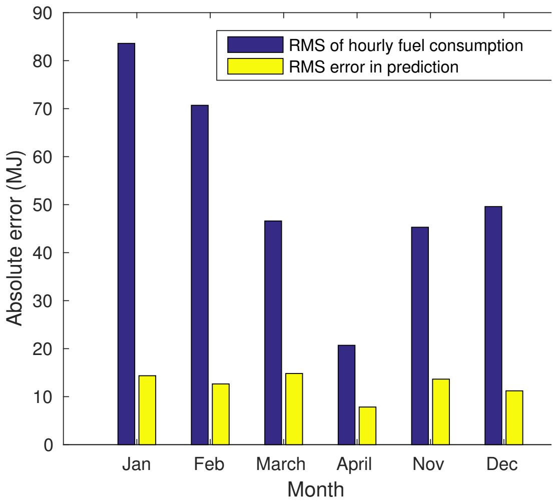

Figure 12. Bar chart showing RMS of simulated fuel consumption in a small office (in MJ) at Hill City, MN, with associated errors in NN predictions

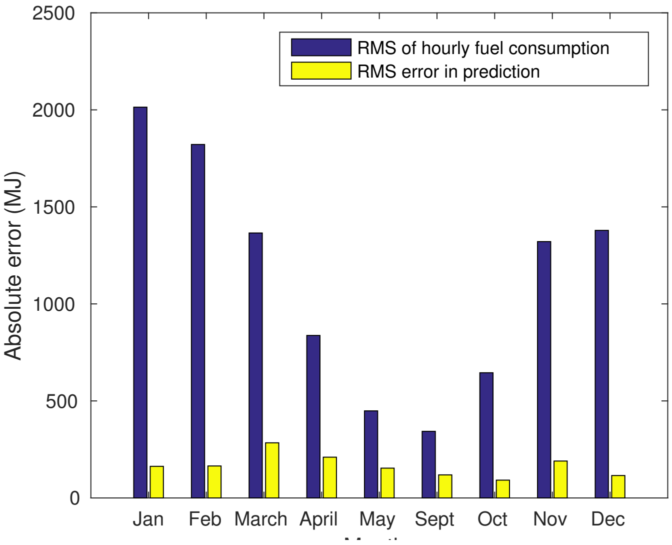

Figure 13. Bar chart showing RMS of simulated fuel consumption in a supermarket (in MJ) at Hill City, MN, with associated errors in NN predictions

metrics f, the ratio $\sigma_{gauss,rms}/RMS(y_{sim,t})$ and $e_{cov,rms}$ vary with scale factor κ for a January fuel consumption profile at Hill City, MN. The figure shows that as the root-mean-squared value of the covariance interval/uncertainty (σ_{gauss}) increases linearly with scale factor κ, the probability f increases with diminishing returns. The plot also shows that fixing one of the performance parameters f, $\sigma_{gauss,rms}/RMS(y_{sim,t})$ and $e_{cov,rms}$ automatically fixes the other two. Thus, for the aforementioned fuel consumption profile, if we choose to have a confidence level of greater than 90%, we need to fix $\kappa > 0.2$, and have to concede $\sigma_{gauss,rms}/RMS(y_{sim,t}) > 11.8\%$. Physically, this means that at $\kappa = 0.2$, the EnergyPlus-simulated fuel consumption for small office will be within ± 11.8% of the GP predictions with a probability of 90%. Hence, there is a trade-off between the confidence level and the associated uncertainty in prediction.

Figure 26 depicts the corresponding plot for GP predictions for a January fuel consumption profile at a supermarket in Phoenix, AZ. While the trends are similar to those observed in figure 25, the user-defined choices of κ for the two target functions are unlikely to be identical. For the case of a small office in Hill City, MN, the value of f and $\sigma_{gauss,rms}/RMS(y_{sim,t})$ at $\kappa = 0$ are 0.59 and 20% respectively; whereas for the case of a supermarket in Phoenix, AZ, the corresponding values are 0.86 and 9.87% respectively. Thus, for the reference case of $\kappa = 0$, the GP covariance functions can encapsulate the January hourly gas consumption profile for a supermarket in Phoenix, AZ with a higher probability and a lower uncertainty, compared to the gas consumption profile for a small office in Hill City, MN. As a result, GP predictions of the January gas consumption profile in supermarket in Phoenix, AZ is likely to require a comparatively lower scale factor.

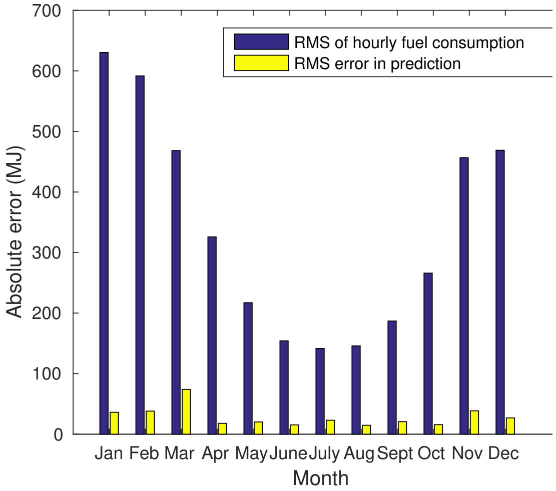

Figure 14. Bar chart showing RMS of simulated fuel consumption in a restaurant (in MJ) at Hill City, MN with associated errors in NN predictions

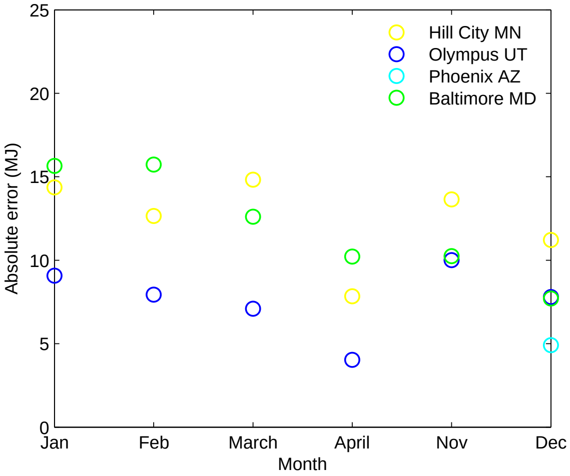

Figure 15. Absolute Errors in fuel consumption (MJ) for a small office at different locations. The maximum absolute error for all locations is 15.7 MJ.

4.4.2 Effect of shifting schedules on machine learning model accuracy

So far, we have assumed that the schedule variables $\mathbf{v}$ are invariant from the training set to the test set. We will relax this assumption in this section by perturbing the infiltration schedule in a restaurant at small office from the training feature to test feature set, and observing the increment in $e_{rms,NN}$, $e_{rms,GP}$ and $e_{cov,rms}$. The transformed schedule in the test set $\mathbf{v'_{inf}}$ can be expressed as follows:

$$v_{inf,e}(t) = v_{inf,t}(t-c) \tag{31}$$

Here t is time in hours, where t = 2,3....24, and c is a constant (in hours) associated with the shifting function. Figure 24 displays the translation expressed in equation 31, which is essentially a co-variate shift with respect to schedule variables $\mathbf{v}$. We can reason that the infiltration schedule contributes significantly to building fuel consumption, as the corresponding Pearson coefficient R_{inf} is between [0.6, 0.8] for all months. Figure 27 shows the relative errors $e_{rms,NN}$, $e_{rms,GP}$ and $e_{cov,rms}$ for c = 0,1 and 2. The figure reiterates the claim that neural networks are more robust to overfitting compared to Gaussian processes when predicting point estimates. The increment in $e_{rms,NN}$ < 2% for c = 1 and < 4% for c = 2, for all months; while the maximum increment in $e_{rms,GP}$ is 5% and 9% for c = 1 and c = 2 respectively. The covariance error $e_{cov,rms}$, however, does not increase as much: the marginal increments in $e_{cov,rms}$ for c = 1 and c = 2 are 1.6% and 3.6% respectively. This could be because with increasing dissimilarity between the training and test sets due to covariance shift, the uncertainty in prediction, as indicated by the predictive variance σ^2_{gauss} increase. This increase in predictive variance, to an extent, ensures that the covariance bounds determined by GP either encapsulate or are close to the actual targets - thus preventing a drastic increase in $e_{cov,rms}$ with increasing value of c. We can use the GP predictions for the month of May as an example: the relative error $e_{rms,GP}$ increase by a margin of 9% from $c = 0$ (baseline case) to $c = 2$: when the corresponding $e_{cov,rms}$ only increases by a margin of 3.6%. This occurs as the

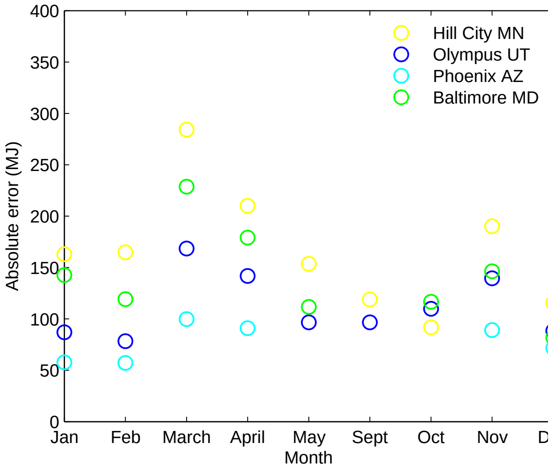

Figure 16. Absolute Errors in fuel consumption (MJ) for a supermarket at different locations. The maximum absolute error for all locations is 284.3 MJ.

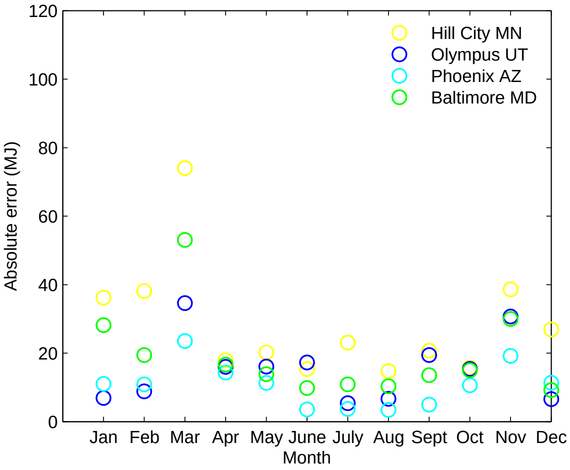

Figure 17. Absolute Errors in fuel consumption (MJ) for a restaurant at different locations. The maximum absolute error for all locations is 74.0 MJ.

ratio $\frac{\sigma_{gauss}}{RMS(y_{sim,t})}$ increases from 0.0272 to 0.0820 as c increases from 0 to 2, thus compensating for the co-variate shift in schedule variables.

However, in practice, it is more likely that occupancy schedule will vary significantly from one year to the next, as opposed to variables associated with building operation. Figure 27 shows how the errors $e_{rms,NN}$, $e_{rms,GP}$ and $e_{cov,rms}$ vary for different c values corresponding to a similar shift in occupancy schedule, i.e. the occupancy schedule undergoes a similar transformation from training set to test set as one described by equation 31. In physical terms, this transformation represents a case where the mean and the peak occupancy values remain the same, but the time slot at which a given occupancy value occurs is translated by c hours from training to test set. We have discussed previously that the relative errors associated with small office are higher compared to those for other two building types. Figure 28 shows that for a small office, the relative errors are more sensitive to a shift in schedule variables. We notice again that for the given configurations of NN and GP, neural networks are more robust to a shift in occupancy schedule. The maximum increase in errors $e_{rms,NN}$ are 4.3% and 4.5% for c = 1 and c = 2 respectively, whereas the corresponding maximum increase in $e_{rms,GP}$ are 18% and 35%. The covariance error $e_{cov,rms}$, again, is not as prone to co-variate shift, with maximum increments in $e_{cov,rms}$ being 4.5% and 5.6% for c = 1 and c = 2 respectively. Thus, static NN is preferred to GP regression for computing point estimates of fuel consumption at one-hour resolution when schedules are likely to be perturbed from one year to the next, whereas GP regression can still accurately predict range estimates under such schedule shifts.

5 Conclusions

This analysis develops a robust machine learning framework to make one year ahead forecasts of fuel consumption at one-hour resolution for multiple building types and climate zones. The scheme uses a feature selection procedure to find an optimal set of input variables that a given ML algorithm

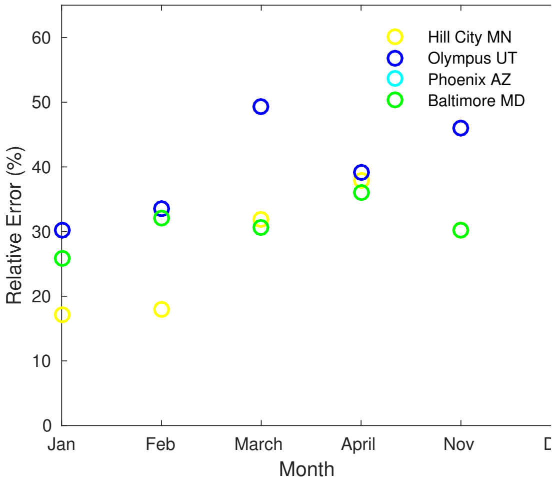

Figure 18. Scatter Plot showing error values in predicting fuel consumption in small office at different climate zones

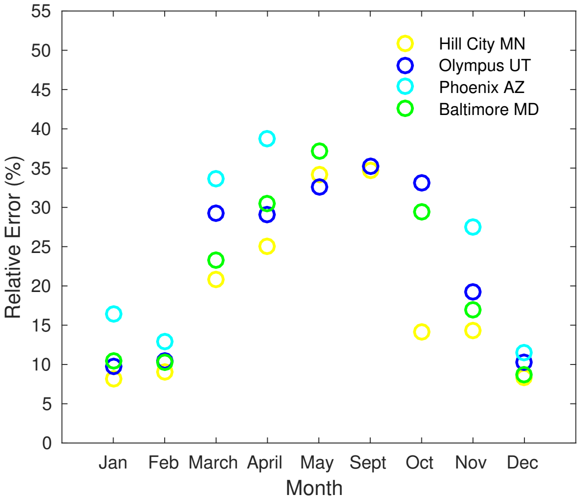

Figure 19. Scatter Plot showing error values in predicting fuel consumption in supermarket at different climate zones

can use to make hourly fuel consumption predictions. The analysis also recommends static neural network for predicting point estimates of hourly fuel consumption and Gaussian process regression for predicting an interval that has a high probability of bounding the target values of hourly fuel consumption.

The key observations from this analysis can be summarized as follows:

- Static neural network and Gaussian Process regression have lower prediction errors compared to NARX, multivariate linear regression and ridge regression. The NN and GP prediction errors are, in general, usually within a margin of 5% of each other.

- The effects of variability of weather variables: both between the training and the test feature set, and within the test feature set can quantitatively evaluated using similarity metrics. These metrics can be used to generalize trends in ML performance across multiple building types and climate zones.

- The maximum absolute error in model prediction for all climate zones were 15.7 MJ (14,880 Btu), 284.3 MJ (268,516 Btu) and 74.0 MJ (70,138 Btu) for small office, supermarket and restaurant respectively. The maximum relative errors in predicting monthly fuel consumption are 23.5%, 6.78% and 5.50% respectively for the aforementioned building types.

  Future work comparing ML algorithms with deterministic energy balance methods will investigate techniques for their integration to improve on prediction accuracies obtained in this analysis. Clustering algorithms also show promise for cases where a high number of training observations are available (>3000 data points).

6 Acknowledgments

The authors would like to thank Vivek Srikumar (School of Computing, University of Utah) for his helpful suggestions related to the development of this research.

This material is based upon work supported by the National Science Foundation under the following Grant: CBET 1512740.

References

[1] Building and climate change.
URL `http://www.eesi.org/files/climate.pdf`

[2] N. Tanaka, others, Technology roadmap: Electric and plug-in hybrid electric vehicles, International Energy Agency, Tech. Rep.

[3] P. Torcellini, S. Pless, M. Deru, D. Crawley, Zero energy buildings: a critical look at the definition, National Renewable Energy Laboratory and Department of Energy, US.

[4] D. B. Crawley, J. W. Hand, M. Kummert, B. T. Griffith, Contrasting the capabilities of building energy performance simulation programs, Building and Environment 43 (4) (2008) 661–673. doi:10.1016/j.buildenv.2006.10.027.
URL `http://linkinghub.elsevier.com/retrieve/pii/S0360132306003234`

[5] X. L, T. Lu, C. J. Kibert, M. Viljanen, Modeling and forecasting energy consumption for heterogeneous buildings using a physicalstatistical approach, Applied Energy 144 (2015) 261–275. doi:10.1016/j.apenergy.2014.12.019.
URL `http://linkinghub.elsevier.com/retrieve/pii/S0306261914012689`

[6] US Department of Energy, EnergyPlus Documentation (2013).

[7] H. S. Rallapalli, A comparison of energyplus and equest whole building energy simulation results for a medium sized office building, Ph.D. thesis, Arizona State University (2010).
URL `http://repository.asu.edu/attachments/56303/content/rallapalli`$_{a}su_{0}010n_{1}0220.pdf$

[8] Building Energy Simulation Accuracy.

[9] L. Pedersen, Use of different methodologies for thermal load and energy estimations in buildings including meteorological and sociological input parameters, Renewable and Sustainable Energy Reviews 11 (5) (2007) 998–1007. doi:10.1016/j.rser.2005.08.005.
URL `http://linkinghub.elsevier.com/retrieve/pii/S1364032105000924`

[10] N. Fumo, M. Rafe Biswas, Regression analysis for prediction of residential energy consumption, Renewable and Sustainable Energy Reviews 47 (2015) 332–343. doi:10.1016/j.rser.2015.03.035.
URL `http://linkinghub.elsevier.com/retrieve/pii/S1364032115001884`

[11] L. Breiman, others, Statistical modeling: The two cultures (with comments and a rejoinder by the author), Statistical Science 16 (3) (2001) 199–231.
URL `http://projecteuclid.org/euclid.ss/1009213726`

[12] D. Kriesel, A Brief Introduction to Neural networks.

[13] S. A. Kalogirou, Artificial neural networks in renewable energy systems applications: a review, Renewable and sustainable energy reviews 5 (4) (2001) 373–401.
URL http://www.sciencedirect.com/science/article/pii/S1364032101000065

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

[15] G. K. Tso, K. K. Yau, Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks, Energy 32 (9) (2007) 1761–1768. doi:10.1016/j.energy.2006.11.010.
URL http://linkinghub.elsevier.com/retrieve/pii/S0360544206003288

[16] K. Yun, R. Luck, P. J. Mago, H. Cho, Building hourly thermal load prediction using an indexed ARX model, Energy and Buildings 54 (2012) 225–233. doi:10.1016/j.enbuild.2012.08.007.
URL http://linkinghub.elsevier.com/retrieve/pii/S0378778812003933

[17] P. A. Gonzlez, J. M. Zamarreo, Prediction of hourly energy consumption in buildings based on a feedback artificial neural network, Energy and Buildings 37 (6) (2005) 595–601. doi:10.1016/j.enbuild.2004.09.006.
URL http://linkinghub.elsevier.com/retrieve/pii/S0378778804003032

[18] E. Busseti, I. Osband, S. Wong, Deep learning for time series modeling, Tech. rep., Technical report, Stanford University (2012).
URL http://www.stanford.edu/ iosband/docs/CS229.pdf

[19] W. Charytoniuk, M.-S. Chen, P. Van Olinda, Nonparametric regression based short-term load forecasting, Power Systems, IEEE Transactions on 13 (3) (1998) 725–730.
URL http://ieeexplore.ieee.org/xpls/abs$_{a}ll.jsp?arnumber = 708572$

[20] D. C. Park, M. A. El-Sharkawi, R. J. Marks, L. E. Atlas, M. J. Damborg, others, Electric load forecasting using an artificial neural network, Power Systems, IEEE Transactions on 6 (2) (1991) 442–449.
URL http://ieeexplore.ieee.org/xpls/abs$_{a}ll.jsp?arnumber = 76685$

[21] B. B. Ekici, U. T. Aksoy, Prediction of building energy consumption by using artificial neural networks, Advances in Engineering Software 40 (5) (2009) 356–362. doi:10.1016/j.advengsoft.2008.05.003.
URL http://linkinghub.elsevier.com/retrieve/pii/S0965997808001105

[22] J. Yang, H. Rivard, R. Zmeureanu, On-line building energy prediction using adaptive artificial neural networks, Energy and Buildings 37 (12) (2005) 1250–1259. doi:10.1016/j.enbuild.2005.02.005.
URL http://linkinghub.elsevier.com/retrieve/pii/S0378778805000502

[23] C. E. Rasmussen, r. o. K. I. Williams, Gaussian processes for machine learning, MIT Press, Cambridge, Mass., 2006.
URL http://www.books24x7.com/marc.asp?bookid=12939

[24] V. M. Z. Yeonsook Heo, Gaussian process modeling for measurement and verification of building energy savings, Energy and Buidings (53) (2012) 7–18.

[25] L. L. C. Stetz Consulting, Regression for M&V: Reference Guide.

[26] I. G. Y. Bengio, A. Courville, Deep learning, book in preparation for MIT Press (2016).
URL http://www.deeplearningbook.org

[27] A. Barron, Approximation and estimation bounds for artificial neural networks, in: Computational Learning Theory, Proceedings of the Fourth Annual Workshop, 1991.

[28] MATLAB, MATLAB Neural network Toolbox 6.

[29] MATLAB, The GPML Toolbox version 3.5.

[30] L. D. J. P. B. W. C. C. S. L. J. S. D. Z. J. Horel, M. Splitt, J. Burls, Mesowest: Cooperative mesonets in the western united states, Energy and Buildings 42 (2002) 211–225.

[31] N. Fumo, P. Mago, L. Rogelio, Methodology to estimate building energy consumption using energyplus benchmark models, Energy and Buildings 42 (2010) 2331–2337.

[32] I. Guyon, A. Elisseeff, An introduction to variable and feature selection, The Journal of Machine Learning Research 3 (2003) 1157–1182.
URL http://dl.acm.org/citation.cfm?id=944968

[33] O. Giustolisi, D. Savic, Advances in data-driven analyses and modelling using epr-moga, Journal of Hydroinformatics 11 (3-4) (2009) 225–236.

[34] A. Storkey, When training and test sets are different: characterizing learning transfer, Dataset shift in machine learning (2009) 3–28.
URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.298.1373rep=rep1type=pdf

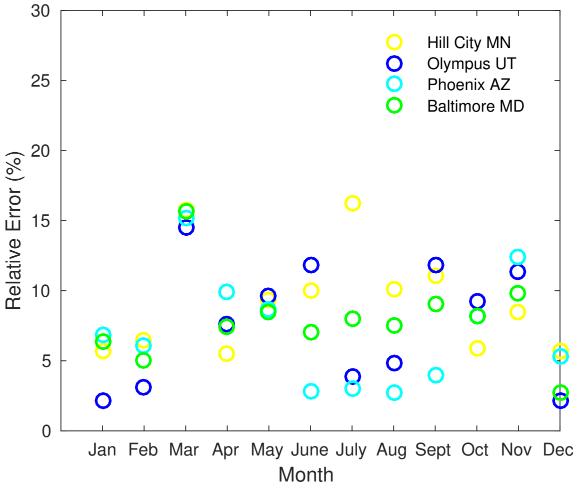

Figure 20. Scatter Plot showing error values in predicting fuel consumption in small office at different climate zones

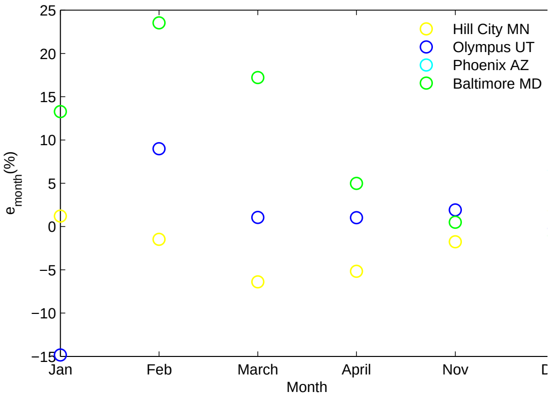

Figure 21. Scatter Plot showing error values in predicting monthly fuel consumption for a small at different climate zones

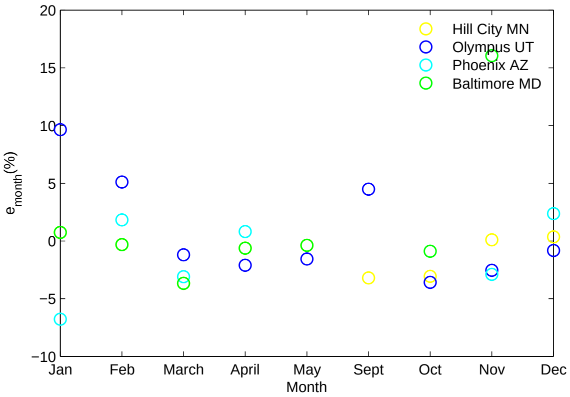

Figure 22. Scatter Plot showing error values in predicting monthly fuel consumption in a supermarket at different climate zones

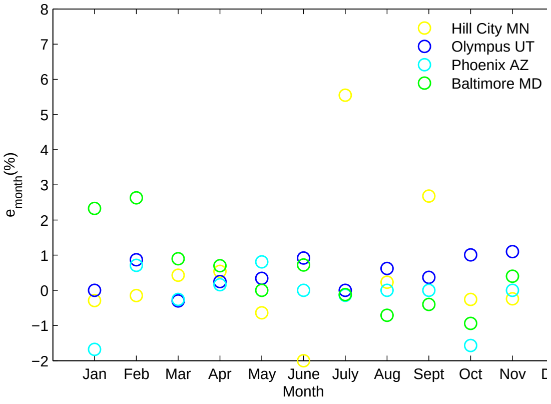

Figure 23. Scatter Plot showing error values in predicting monthly fuel consumption in restaurant at different climate zones

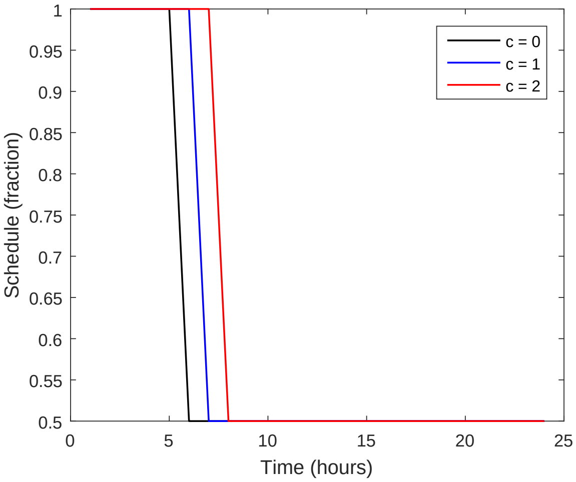

Figure 24. Plot showing the transformed schedules for different ’c’ values

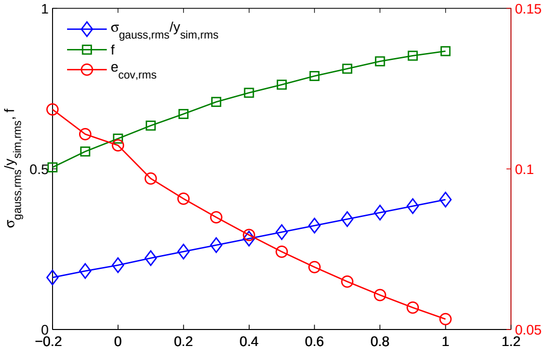

Figure 25. f, $\sigma_{gauss,rms}/RMS(y_{sim,T})$ and $e_{cov,rms}$ vs. κ corresponding to GP predictions for fuel consumption profile in January in a small office at Hill City, MN

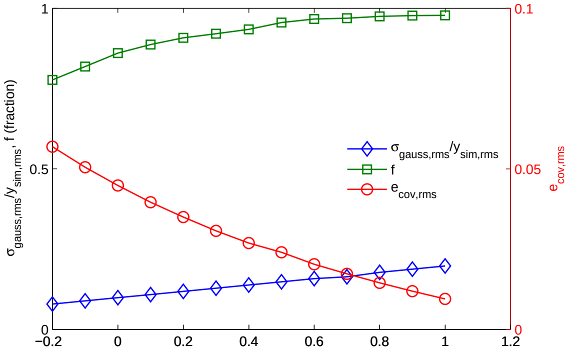

Figure 26. f, $\sigma_{gauss,rms}/RMS(y_{sim,t})$ and $e_{cov,rms}$ vs. κ corresponding to GP predictions for fuel consumption profile in January in a supermarket at Phoenix, AZ

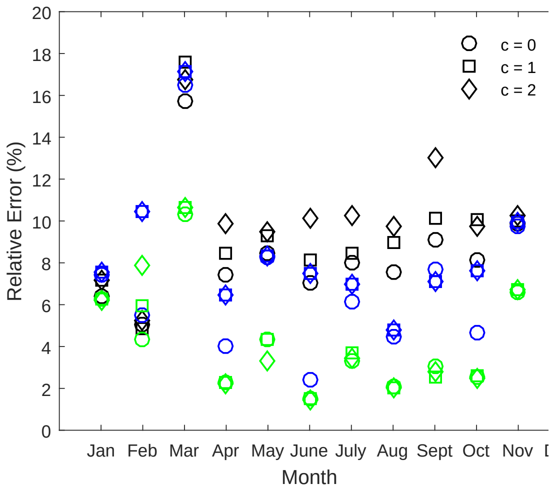

Figure 27. Scatter plot showing relative errors $e_{rms,NN}$, $e_{rms,GP}$ and $e_{cov,rms}$ corresponding to different c values for a restaurant in Baltimore, MD. The black, blue and green markers correspond to $e_{rms,NN}$, $e_{rms,GP}$ and $e_{cov,rms}$ respectively

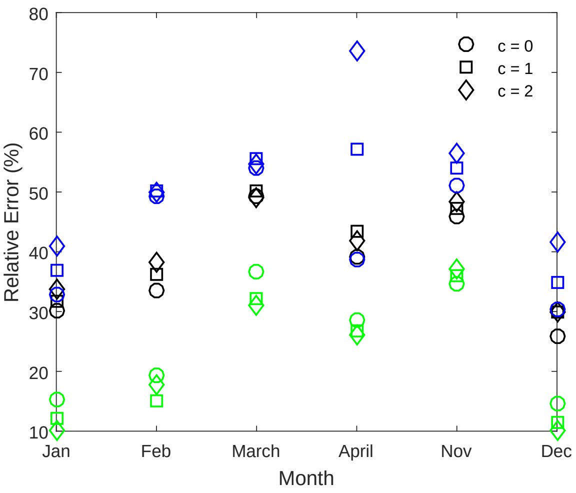

Figure 28. Relative errors $e_{rms,NN}$, $e_{rms,GP}$ $e_{cov,rms}$ corresponding to different c values for a small office in Olympus, UT. The black, blue and green markers correspond to $e_{rms,NN}$, $e_{rms,GP}$ and $e_{cov,rms}$ respectively.