Int. J. of Thermal & Environmental Engineering
Volume 13, No. 2 (2016) 81-89

Short Term Forecasts of Internal Temperature with Stable Accuracy in Smart Homes

Bruce Spencer[a] *, Feras Al-Obeidat[b], Omar Alfandi[b,c]

[a] *University of New Brunswick, Fredericton, Canada*
[b] *Zayed Universty, Abu Dhabi, UAE*
[c] *University of Göttingen, Göttingen, Germany*

Abstract

We forecast internal temperature in two homes, using variants of regression with data from the readings of multiple sensors. We use 48 separate models, where each forecasts mean temperatures that will occur in one future 15-minute interval, to compose a forecast for the next 12 hours. The sensors report internal and external atmospheric and environmental conditions such as temperature, pressure, sunlight, rain and wind, as well as evidence of human activity, including CO_2 saturation, motion sensors and electrical load from areas within the house and large appliances. The models use both current and historical sensor values, each of which increases the number of predictors in the linear regression model. We use model simplification techniques including forward stepwise regression, principal component regression, and partial least squares regression. In both houses the forecast accuracy is stable; the mean absolute error over 12 hours is less than 1, while the root mean squared error is less than 1.3. Our accuracy compares favorably to previous work. Our work indicates long sensor histories for forecasts in the next 12 hours do not significantly improve accuracy.

Keywords: *Smart Home, Linear Regression, Temperature Forecasting*

1. Introduction

According to recent studies, about 40% of energy produced worldwide is consumed by buildings, and more than half of this is used by Heating, Ventilation and Air Conditioning (HVAC) systems [1,2]. Accurate temperature forecasts can reduce energy usage in buildings in two ways. First, as Pan et al. [3] state, due to thermal inertia, it is more efficient to maintain temperature in a room or building than to heat or cool it, so an HVAC controller with foresight can do a better job of maintaining temperatures and avoiding contributing to temperature swings. Second, Model Predictive Controllers, which produce a control signal for HVAC systems, minimize a cost function based on energy consumption. The cost function takes into account a prediction horizon and a control horizon[4]. Considerable savings can result: Moreno et al. [5] achieve estimated energy savings of 20%. Thus, a rough estimate of potential savings arising from forecasting temperature in buildings is 4% of all energy produced.

We analyze publicly available data from two different houses to forecast temperature. Data provided by Zamora-Martinez et al. [6] reports 18 sensors from the SML house over one period of about four weeks and another period of about three weeks. The Smart* Project [7] provides publicly available data including environmental readings, circuit loads, motion detectors, and switches controlling lights and fans, over a three-month period. We predict temperature in these homes for every 15-minute interval i.e. a temperature estimate 15 minutes in the future, 30 minutes, and so on, up to twelve hours. We measure the accuracy of each forecast individually and the accuracy over the entire twelve-hour horizon, i.e. over all 192 intervals, and up to sub-horizons that end sooner. We consider only the first of the two periods for the SML house, and only *Home A* from the Smart* house.

In the remainder of this paper, we review the data sets, statistical methods applied to these data sources based on linear regression. Because we access many sensors and for each sensor, we include some number of historical readings, our models can contain many predictors. For n sensors using a current value and b historical values for each sensor, our models include n (b + 1) predictors. Since the model sizes can be large, we employ a variety of model reduction techniques, including forward stepwise regression, principal components, partial least squares regression. We compare to previous work and conclude

* Corresponding author.
E-mail: bspencer@unb.ca

DOI: 10.5383/ijtee.13.02.002

techniques give rise to usable forecasts with errors that are low and accuracy that is stable. Some of our methods involve a manual step, so automating or avoiding that is left for future work.

2. Datasets and Previous Work

Zamora-Martinez et al. [6], reporting on the SML house, provide 18 sensors over a four-week period in March and April of 2012. This data is publicly available [8]. The data is presented as a time series with 15-minute resolution to an online learning framework that forecasts 48 hours of forecasts, or 192 forecasts. Only two of the 18 sensors are used in this work: internal temperature and sun irradiance, and the hour is encoded as 24 Booleans. From an initially uninformed model, each new set of sensor readings is loaded and the model is improved. After 5 days of data, about 480 observations, the forecasts show good accuracy. There are two modelling technologies studied: an artificial neural network, which is trained using a variety of mechanisms that are variants of gradient descent, and also a Bayesian linear model, which performs the best most often. Instead of forecasting actual temperatures, they forecast the differences in temperature between periods. Accuracy is calculated for each of the methods over a number of experiments. Forecasts have MAE below 0.2 C occur about half of the time, the median error is 0.5 C, but can be up to 1 C.

In earlier work by the SML group [9], forecasts are computed for each minute over a three-hour forecast period, using five sensors along with an indicator of the hour of the data. Before modelling, the temperatures are replaced by the moving average over the previous 5 periods, and the resulting averages are normalized so that they have a mean of 0 and a standard deviation of 1. Therefore, they do not represent a number of degrees. The MAE over the three hour forecast horizon is 0.133, with a maximum of 1.98. Normalized RMSE is reported but not RMSE.

The data provided on the Smart* website [10] for Home A covers May, June and July 2012, and report either at specific times, or as energy demand is made, depending on the sensor. We accumulate all data into 15 minute intervals. Each day has intervals 0,...,95, where interval 0 occurs at 12 midnight and reports on readings from 11:45PM to midnight. The environmental data is averaged over all readings within the interval and reported at the end of that interval. Similarly, the load from each circuit within the home is averaged over the period. For the motion detector, we add up the number of motion events detected over the interval, and for the switch, we take the average over the interval of each power estimate. Power is computed by multiplying the maximum wattage of the switch by the proportion that it was dimmed. As far as we know, no previous temperature forecasting has been done with the Smart* data.

3. Temperature Forecasting with Linear Regression

Using terminology of Cheng et al. [11] , we perform multi-step-ahead direct forecasting in that we compute multiple forecasts into the future with a fixed set of data. Following terminology from Ben Taieb et al. [12] we use a pure direct forecast strategy to do this, by creating one forecast model for each possible future quarter-hour interval. We report forecast errors for each 15-minute interval and also for each forecast horizon, ending with the error over the full 12 hours.

We are not using an online learning approach, but instead we compute our models on a pre-collected set of recent readings called the training period, consisting of the first 2/3 of the available data and evaluate them on the test period, which is the remaining 1/3. The test period immediately follows the training period.

In linear regression, we are given a set of independent variables $x_1,\ldots,x_n$ and a dependent variable y of interest that we want to forecast as a function of the independent variables. Specifically, we seek parameters $\beta_0,\ldots,\beta_n$ so that $\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$ is a good approximation of y. When presented with a set of m instances of each x_i, called $x_{i,j}$ and the corresponding instances y_j, we select the β_i parameters so the root mean squared error (RMSE) function is minimized, where RMSE is

$$\sqrt{1/m \sum_{j=1}^{m} \left(\beta_0 + \sum_{i=1}^{n} \beta_i x_{i,j} - y_j\right)^2}$$

We also are interested in minimizing the mean absolute error (MAE), which is

$$1/m \sum_{j=1}^{m} \left|\beta_0 + \sum_{i=1}^{n} \beta_i x_{i,j} - y_j\right|$$

and the residual sum of squares (RSS) over the training data, which is

$$\sum_{j=1}^{m} \left(\beta_0 + \sum_{i=1}^{n} \beta_i x_{i,j} - y_j\right)^2$$

In temperature forecasting, the independent variables are recent readings from various sensors. Let the lag l vary across the intervals 0 through b into the past, and let k vary across the s sensors. Let $x_{k,t}$ be the t^{th} observation for the sensor k counting from the earliest observation in the data at $t = 1$. Let y_f be the independent variable, which is the internal temperature of the house at some future time. We let f represent the number of 15-minute periods into the future. For a fixed future period f , we seek the value of the coeffcient β_0 for the intercept of regression and s × b values for the coefficients $f_{k,t}$ of the k^{th} sensor at time t. We want to minimize the residual sum of squares

$$\sum_{t=b+1}^{m} \left(\beta_{f,0} + \sum_{l=0}^{b} \sum_{k=1}^{s} \beta_{f,k,t-l} x_{k,t-l} - y_{f+t}\right)^2$$

In this equation, t starts at $b+1$ because there are no lagged observations for the first b data points. We focus on forecasting the next 12 hours, so $f = 1, \ldots, 48$ periods.

There are s × (b + 1) predictors for each model, and there are s = 37 sensors in one of our data sets, and we use up to b = 8, so we deal with up to 37 × 9 = 333 predictors. We seek methods to simplify the model. One of the goals of this paper is recommend effective model reduction techniques for temperature forecasting.

3.1. Model Reduction Techniques

We conduct a variety of experiments where we vary the solver, the data file and the number of historical values from each sensor. The solvers are ordinary least squares regression, forward stepwise regression, principal component regression and partial least squares regression. In ordinary least squares, we use the lm method provided in R.

Our first model simplification technique is the variant of linear regression known as forward stepwise linear regression, first defined by Miller [13], using stopping rules defined by Bendel et al. [14]. We followed the presentation by Hastie et al. [15], and provided the leaps package [16], using the R method `regsubsets`. This method initially sets β_0 to the mean value of y and all other $\beta_i = 0$. Then it repeatedly selects a value for β_i so that the error function is reduced as much as possible among all such choices. Once a value of β_I is selected it is not changed further. After all such β_i are selected, stepwise regression halts with the model.

We also apply principal component and partial least squares regression, in which linear combinations of the original predictors are "repackaged" as separate components. Each component is defined in turn, until the number of components as specified by the hyperparameter is reached. In the case of principal components regression, the first component is the linear combination of all of the predictors that has maximal variation. The second and subsequent components maximize variation among components that are orthogonal to all previous components. Note that principal component regression does not pay attention to the values of independent variable. In the case of partial least squares, each coefficient of the first component is chosen to be proportional to the correlation of that predictor and the independent variable. The second and subsequent components are computed in the same way but instead of using the independent variable directly, they are based on the residual after the information in the previous components is removed. Note that partial least squares is supervised by the independent variable.

These two techniques reduce the problem of finding coefficients for all predictors to one of defining a smaller number of components and finding coefficients for each. In that sense, the number of dimensions is reduced. However, unlike the forward stepwise restriction, neither principal component nor partial least squares regression selects or filters out specific predictors, since each component is a linear combination of all predictors. Depending on the coefficients used, some predictors are left with little influence.

We use the `pcr` and `plsr` methods to perform principal component and partial least squares regression, respectively, specifying the validation option as CV. These methods are in the `pls` library [17,18] and described by a vignette [19].

3.2. Hyperparameter Training and Cross Validation

Once the data is prepared, we divide it into two periods, a contiguous (*hyperparameter) training period*, and an adjoining, later, contiguous *testing period*. The model is built using only the training period. As is typical, 2/3 of the data is used for training and 1/3 is used for testing. By insisting the separate sets are each contiguous, we can claim authenticity with the real-world setting where we are forecasting into a future represented by the training periods that has not yet occurred.

If the modelling technology has hyperparameters to train, first we train them and settle on appropriate values for the hyperparameters. Hyperparameters are selected from the training data. After the hyperparameter training is done, or if there are no hyperparameters, we build our final selected models directly from all of the training data.

We apply cross-validation during hyperparameter training. In cross-validation, first the training data is divided into ten randomly selected subsets called "folds". These subsets do not have to consist of contiguous data. Given a candidate hyperparameter value, we perform ten experiments. In each, one of the folds is considered the test fold and a model is trained on the other nine folds. We want to investigate the error arising from various values of the hyperparameters in order to select a value that will have the lowest error over the entire test set. How this is done depends the regression method. Ordinary linear regression has no hyperparameters, so that model is built directly from the entire training set.

In the case of foward stepwise regression, the hyperparameter is the number predictors that need coefficients. So for a fixed number of predictors, we compute ten models, and for each we measure the Bayesian Information Criteria (BIC)[20], defined as

$$1/n\ (\mathrm{RSS} + log(n)\, d\, \hat{\sigma})^2$$

where n is the number of observations, and d is the number of dimensions of the model, $\hat{\sigma}$ is an estimate of the standard deviation of the internal temperature. BIC penalizes more complex models, and thus balances model complexity against model error on the training data. A model with a smaller BIC is considered better, in that its error may be higher than minimal, but is less prone to over-fitting, and therefore should perform better on unseen data.

For forward stepwise regression, each additional predictor necessarily reduces the error, so BIC reduces and then increases as the number of dimensions increases. We can identify the model size with minimal mean BIC. But since the BIC decreases slowly as it approaches the minimal as the predictors reduce the error by smaller and smaller amounts, we are interested the model size whose mean BIC is within one standard error of this smallest BIC. See Figure 1, where the BIC for each model size is shown for several example models. Each BIC is marked with a confidence interval spanning one standard error, based on the BIC's of the models of that size over the ten folds. The model with the lowest mean BIC is shown labeled green, and a smaller model whose BIC is larger by at most one standard error is labelled red. One can judge how well this heuristic works by examining how the BIC drops as the model size increases. We see reductions of about 7 predictors for the SML models and up to 29 for the Smart* models.

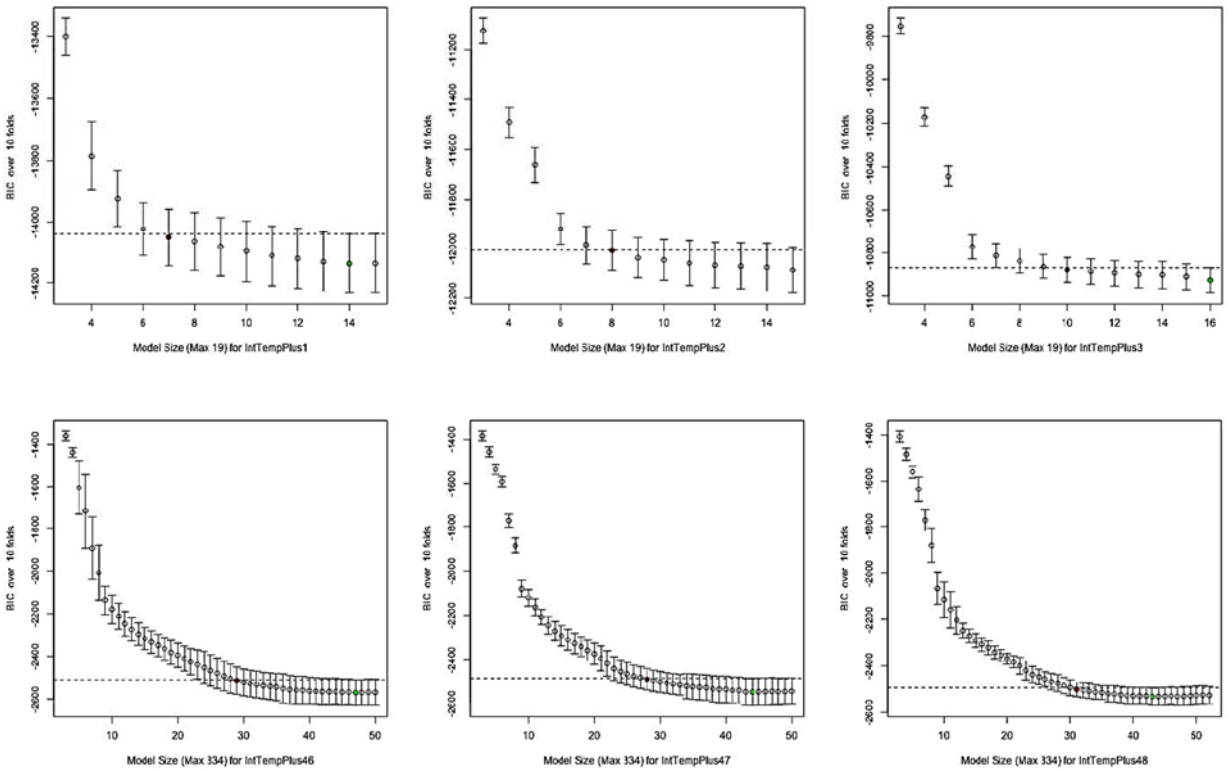

Fig. 1: In forward stepwise regression, selecting the number of predictors for the first three periods of the SML data with zero historical sensor values and the last three periods of the Smart data with 8 historical sensor values. These are our simplest and most complex models, respectively. The green dot represents the model size with minimal BIC and the red dot the smallest model size within one standard error of the minimal. We see reductions of 14-7, 16-8, and 16-10 for the three example SML models, and 47-29, 44-28, and 44-31 for three example Smart* models. Other periods and other sensor histories produce similar reductions. The largest reduction is 29 predictors.

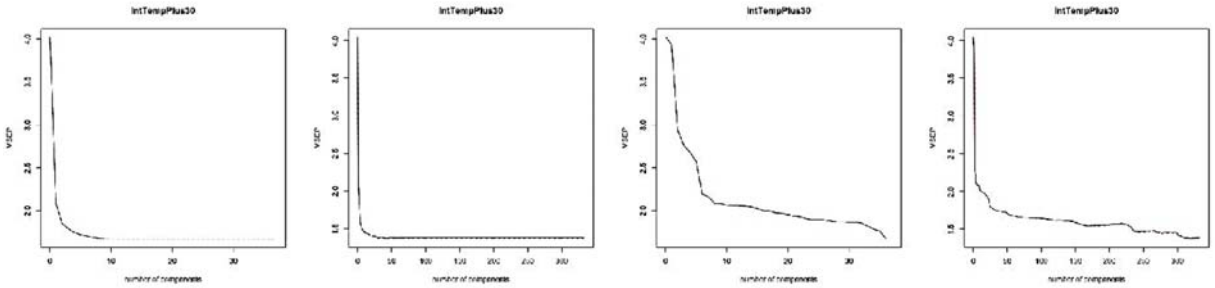

Fig. 2: Validation plots for SML data predicting period 30, at 7.5 hours into the future. From left to right, using partial least squares (PLS), with 0 historical readings, PLS with 8 historical readings, followed by principal components (PC) with 0 and 8 historical readings. Balancing the benefits of fewer components against those of lower error, we chose 4, 12, 6, 60 components, resp.

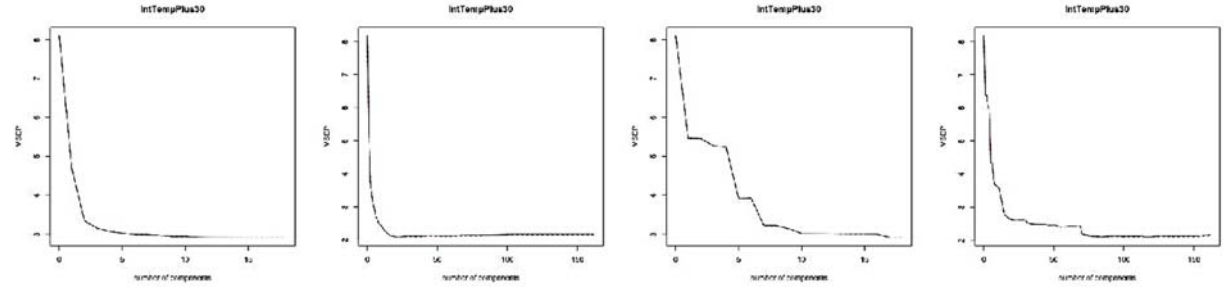

Fig. 3: Validation plots for Smart data predicting period 30, at 7.5 hours into the future for PLS with 0 history, PLS with 8 history, PC with 0 history, PC with 8 history. Balancing fewer components against lower error, we chose 4, 10, 34, and 310 components, resp.

While it is easy to calculate the number of dimensions of the model built by forward stepwise regression, it is less obvious how to do so for the two regression methods that define components: principal component analysis and partial least squares, since each of these uses all predictors for each component. We tried two strategies to help us decide how many components to include. First, the R library `plsdof` [21], described by Kraemer et al. [22] contains a method `pls.ic`, for computing information criteria. In addition to the BIC, it produces an integer `m.opt` with the recommended number of components to be used when doing partial least squares regression. Second, the `pls` library provides the method `validationplot`, which produces plots showing the measured prediction error over various numbers of components, and it uses cross validation. Our choice was whether to use the automatically generated number of components, or to review the plots manually. Our experience showed that `pls.ic` runs slowly when there are a large number of predictors. While it often generated consistent results, this was not always the case. We found the recommended number of components varied within each forecasting problem, even though the forecasting problems were similar. An area of future investigation for us is to become more comfortable with `pls.dof`.

See Figure 2 for some examples of using `validationplot` to select the number of components from the SML data set and Figure 3 for the Smart* data set. Unlike Figure 1, where it is easy to identify the appropriate model size, in Figures 2 and 3, it takes some judgment. One cannot always find a unique point indicating a relatively small model with almost minimal error. Consider the third plot in Figure 2. One might choose 5 components if preferring fewer components, or some value from 7-10 if preferring lower error. For this paper we reviewed the 960 plots= 48 futures × 5 history lengths × 2 forecasting methods × 2 data sets. We selected the number of components shown in Table 1.

We found that a higher number of components was needed more often by principal components than partial least squares, especially using the Smart* data. This aligns with intuition because there are up to 333 predictors in Smart* data, and as principal components is an unsupervised forecasting method, a large amount of data must be included in order for the method to find the right data.

Table 1: Number of Components for Partial Least Squares and Principal Components Regression

Method	History	Intervals	Components
PLS	0	1-30	4
		31-48	5
	1	1-25	5
		26-48	20
	2	1-25	4
		26-48	15
	4	1-20	5
		21-48	10
	8	1-23	3
		24-48	10
PC	0	1-48	34
	1	1-14	44
		15-34	63
		35-48	70
	2	1-20	42
		21-27	85
		21-27	105
	4	1-15	35
		16-48	170
	8	1-5	40
		6-27	200
		28-48	310

(a) Number of components for Smart* Data

Method	History	Intervals	Components
PLS	0	1-19	3
		20-39	4
		40-48	7
	1	1-29	4
		30-39	18
		40-48	24
	2	1-29	4
		30-48	20
	4	1-24	4
		25-48	20
	8	1-23	4
		24-48	12
PC	0	1-9	7
		10-34	6
		35-48	10
	1	1-19	7
		20-29	21
		30-48	34
	2	1-19	9
		20-29	20
		30-48	45
	4	1-19	8
		20-29	20
		30-48	60
	8	1-19	10
		20-29	15
		30-48	60

(a) Number of components for SML Data

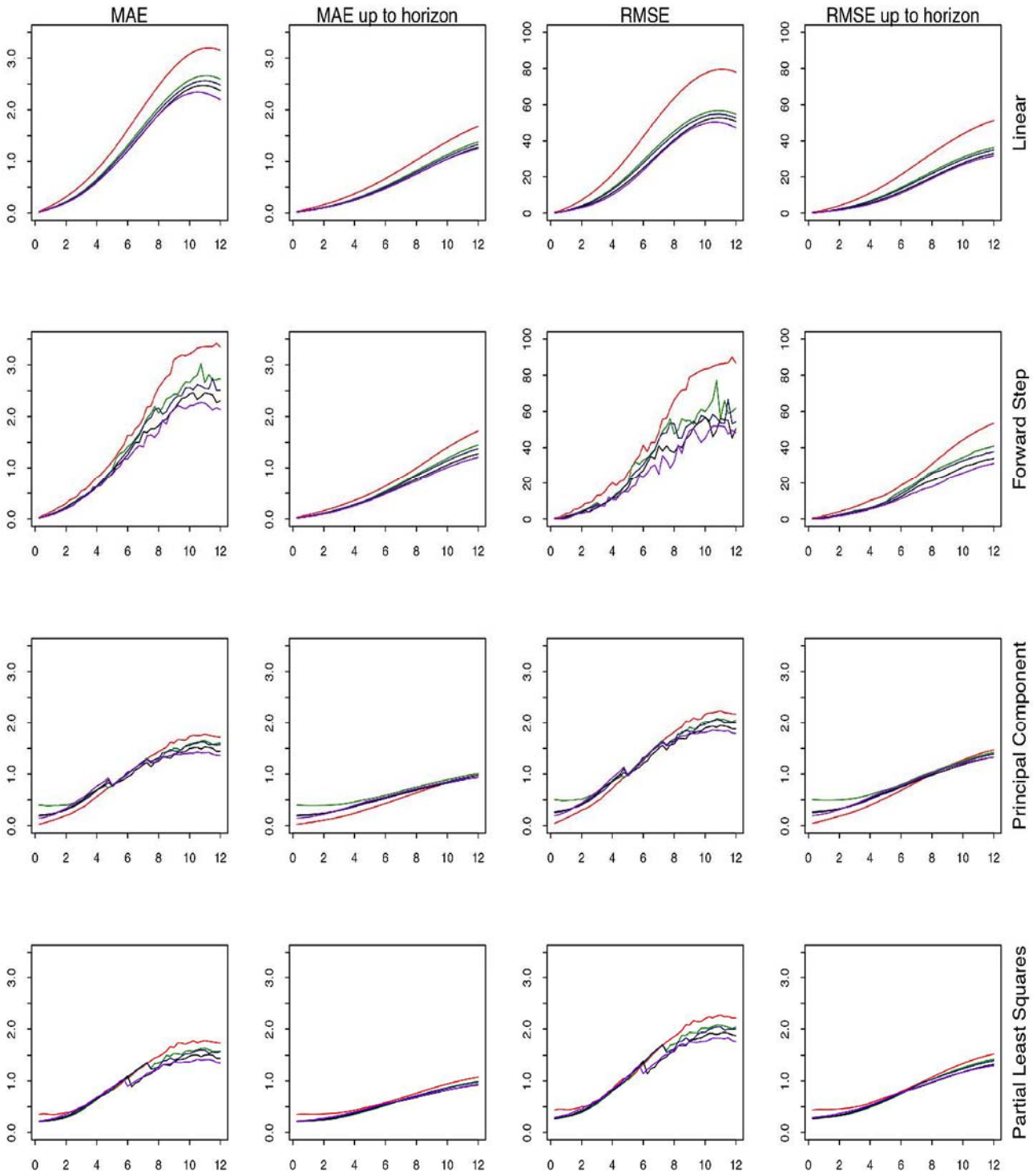

Fig. 4.: Forecast Errors at each hour for Linear, Forward Stepwise, Principal Components and Partial Least Squares Regressions on SML data. The history horizons are 0,1,2,4,8 and their plots are shown in red, green, blue, black and purple, resp

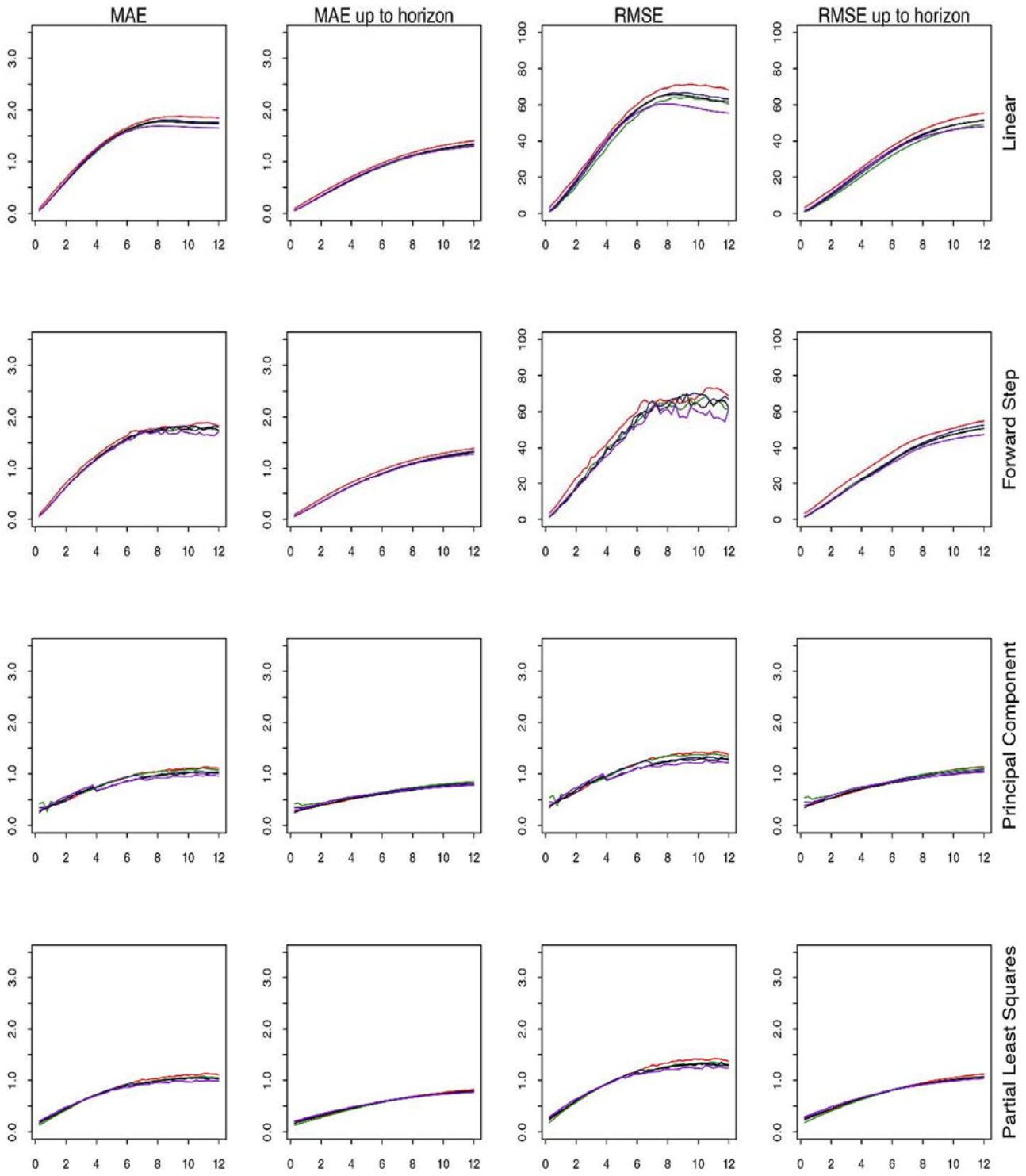

Fig. 5.: Forecast Errors at each hour for Linear, Forward Stepwise, Principal Components and Partial Least Squares Regressions on the Smart* data. The history horizons are 0,1,2,4,8 and their plots are shown in red, green, blue, black and purple, resp.

3.3. Computing Forecasts and Reporting Errors

Once the hyperparameters are selected, we build a model using all of the training data to set the coefficients for this candidate best model, and ran this model on the training data. We report on the error that occurs between the model's prediction and the observed temperature for each of these forecasts. We are interested both in the mean absolute error (MAE), which tells the average absolute difference between forecasted and observed data, and in the root mean square error (RMSE). MAE is a convenient measure since it is a physical quantity that lay people can relate to, e.g. "being 1 degree off, on average". It is possible to have a low MAE but a high RMSE, by having most individual forecast errors low, but some very high. Therefore, RMSE is a better indicator of consistent accuracy. We set all our methods to achieve minimal RMSE, and note that a low MAE is achieved as a by-product.

4. Results and Discussion

Our experiments run the four solvers, five different history horizons and 12 hours of forecasts on the SML data with results shown in Figure 4 and on the Smart* data shown in Figure 5. The first solver is ordinary linear regression, considered the baseline above which all other solvers should achieve. Note that the four plots in the top right corner of both Figures 4 and 5 have a vertical scale of 100, owing to the high RMSE for both the linear regression and forward stepwise regression. All other plots have vertical scale of 3.

The first and third columns report on the error at each interval while the second and fourth report on the error when the forecast window includes all forecasts up to the horizon. For instance, with SML data using 8 lags for each sensor, the partial least squares MAE for the final interval at the end of the 12th hour is 1.35 C, the MAE over the entire forecast period is 0.92 C.

Forward stepwise regression does not appear to offer much of an improvement over linear regression, while principal components and partial least squares are much more accurate, especially considering the RMSE. It is also interesting to note how little improvement arises from including the historical sensor readings. For principal components it appears to sometimes be a hindrance, as the red line representing no historical readings is lower than others for about the first six hours of forecasts on SML data.

In comparison with previous work on the SML data set, we note that results published by the SML group include 2012 and 2014 [9,6]. As noted, the 2014 results cannot be directly compared because the temperatures are replaced by their moving average and then normalized. The 2014 results also cannot be directly compared as the forecast temperature differences between adjacent time intervals, rather than raw temperatures. We ran an experiment to test our method on forecasting temperature differences using principal component regression with no historical sensor readings, without carefully tuning the number of components, but instead selecting again the number of components tuned for forecasting direct temperatures. Thus it was not optimized for this problem. We achieved MAE over the 12 hours of 0.063, with a maximal MAE of 0.08. Their forecasts for these temperature differences over 192 intervals are based on two sensors and have a median MAE of 0.5 and maximal MAE of about 1. They do not report RMSE. Their forecast accuracy does not appear to be stable. Our RMSE over the 12 hours is 0.09 with a maximal RMSE of any one forecast at 0.10 indicating our accuracy is stable for forecasting temperature differences.

5. Conclusions and Future Work

Our goal is to generate accurate forecasts of internal temperature in homes with access to data from a variety of sensors. Temperature forecasts can be used in conjunction with a temperature control unit that decides to apply heating or cooling depending on the current temperature and the forecasted temperature, rather than just the current temperature alone. Such controllers have been shown to reduce energy consumption by HVAC systems by 20% [5], which itself is estimated to be 20% of all energy used [1,2]. Thus potential savings may approach 4% of all energy produced.

We compare four variants of linear regression, including forward stepwise regression, principal component regression and partial least squares regression. We guide model selection by ten fold cross-validation to reduce overfitting.

For the SML house using partial least squares regression models trained on just over 19 days of data with 9 readings each from 18 sensors, we achieve MAE at 0.93 C over a 12 hour forecast horizon with maximal per-interval MAE of 1.4 C, and RMSE at 1.30 over the horizon with maximal per-interval RMSE at 1.76.

For the Smart house using principal component regression models trained on 60 days of data with 9 readings from each of 37 sensors, we achieve MAE at 0.78 C over a 12 hour forecast horizon with maximal per-interval MAE of 1.03 C and RMSE at 1.03 over the interval with maximal per-interval RMSE of 1.3.

Because forecast models based only on immediate data are almost as accurate as those using a history from each sensor, we achieve results almost as accurate using no history. Thus we can restart forecasts immediately after any sensor failure, and so energy savings improvements are not delayed. An online learning strategy that takes five days to train may need an interim strategy to avoid propagating that gap to the stream of forecasts.

Considering the tradeoff between forecast horizon and forecast accuracy, forecasts further into the future have lower accuracy. Forecasting over a 2-hour horizon in the SML house, the MAE is 0.26 C and the RMSE is 0.31. Over the same 2-hour horizon in the Smart house the MAE is 0.42 C and the RMSE is 0.58. In both cases the error rises almost linearly up from the 2- to the 12-hour forecast errors. Thus, better forecasts can be made if a shorter forecast horizon is needed.

While we believe the forecast are usable, due to their low error and stable accuracy, we recognize that the manual process of deciding how many components to include in the partial least squares and principal component regression is time consuming and will address this in future work.

References

[1] P. Morosan, R. Bourdais, D. Dumur, and J. Buisson, "Building temperature regulation using a distributed model predictive control," Energy and Buildings, p. 1445-1452, 2010.

[2] L. Perez´-Lombard, J. Ortiz, and C. Pout, "A review on buildings energy consumption information," Energy and Buildings, vol. 40, p. 394398, 2008.

[3] D. Pan, D. W. Y. Yuan, X. Xu, Y. Peng, X. Peng, and P.-J. Wan, "Thermal inertia:towards an energy conservation room management system," in INFOCOM (A. Greenberg and K. Sohraby, eds.), pp. 2606 – 2610, IEEE, 2012.

[4] J. Alvarez, J. Redondo, E. Camponogara, J. Normey-Rico, M. Berenguel, and P. Ortigosa, "Optimizing building comfort temperature regula-tion via model predictive control," Energy and Buildings, vol. 57, pp. 361 – 372, 2013.

[5] M. V. Moreno, J. Santa, M. A. Zamora, and A. F. Skarmeta, "Context-aware energy e ciency in smart buildings," in 17th International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI 2013), December 2013.

[6] F. Zamora-Mart´ınez, P. Romeu, P. Botella-Rocamora, and J. Pardo, "On-line learning of indoor temperature forecasting models towards energy e ciency," Energy and Buildings, vol. 83, pp. 162–172, November 2014.

[7] S. Barker, A. Mishra, D. Irwin, E. Cecchet, P. Shenoy, and J. Albrecht, "Smart : An open data set and tools for enabling research in sustainable homes," in Proceedings of the 2012 Workshop on Data Mining Applications in Sustainability (SustKDD 2012), August 2012.

[8] UCI, "Sml2010 data set." https://archive.ics.uci.edu/ml/datasets/SML2010.

[9] F. Zamora-Mart´ınez, P. Romeu, J. Pardo, and D. Tormo, "Some empirical evaluations of a temperature forecasting module based on artificial neural networks for a domotic home environment," in International Conference on Knowledge Discover and Information Retrieval, 2012.

[10] "Smart*: Optimizing energy consumption in smart homes." http://lass.cs.umass.edu/projects/smart/.

[11] H. Cheng, P. Tan, J. Gao, and J. Scripps, "Multistep-ahead time series prediction," Lecture Notes in Computer Science 3918, p. 76577, 2006.

[12] S. Ben. Taieb, G. Bontempi, A. Atiya, and A. Sorjamaa, "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition," Expert Systems with Applications, vol. 39, pp. 7067–83, 2012.

[13] A. Miller, Subset Selection in Regression. Chapman & Hall, second ed., 2002.

[14] R. B. Bendel and A. A. Afifi, "Comparison of stopping rules in forward stepwise regression," Journal of the American Statistical Association, vol. 72, no. 357, pp. 46–53, 1977.

[15] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction, Second Edition. Springer, 2011.

[16] T. Lumley, "Package leaps: Regression Subset Selection Ver 2.9." https://cran.r-project.org/web/packages/leaps/leaps. pdf/, February 15 2015.

[17] R. W. Bjørn-Helge Mevik and K. H. Liland. https://cran.r-project.org/web/packages/pls/pls.pdf/, August 22 2015.

[18] R. W. Bjørn-Helge Mevik, "The pls Package: Principal Component and Partial Least Squares Regression in R," Journal of Statistical Software, vol. 18, no. 2, pp. 1–24, 2007.

[19] B.-H. Mevik and R. Wehrens. https://cran.r-project.org/web/packages/pls/vignettes/pls-manual.pdf/, August 22 2015.

[20] A. A. Neath and J. E. Cavanaugh, "The bayesian information criterion: background, derivation, and applications," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 4, p. 199203, March/April 2012.

[21] N. Kraemer and M. Braun. https://cran.r-project.org/web/packages/pls/plsdof.pdf/, February 20 2015.

[22] N. Kramer¨ and M. Sugiyama, "The degrees of freedom of partial least squares regression," Journal of the American Statistical Association, vol. 106, no. 494, pp. 697–705, 2011.