1-1-2017

Selecting Sensors when Forecasting Temperature in Smart Buildings

Bruce Spencer
University of New Brunswick

Feras Al-Obeidat
Zayed University

Omar Alfandi
Zayed University

Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 109C (2017) 777–784
Procedia Computer Science
www.elsevier.com/locate/procedia

7th International Conference on Sustainable Energy Information Technology (SEIT 2017)

Selecting Sensors when Forecasting Temperature in Smart Buildings

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

Forecasts of temperature in a "smart" building, *i.e.* one that is outfitted with sensors, are computed from data gathered by these sensors. Model predictive controllers can use accurate temperature forecasts to save energy by optimally using Heating, Ventilation and Air Conditioners while achieving comfort. We report on experiments from such a house, in which we select different sets of sensors, build a temperature model from each set, and then compare the accuracy of these models. While a primary goal of this research area is to reduce costs by reducing energy consumption, in this paper, besides the cost of energy, we consider the cost of data collection and management. Each sensor employed in the forecast calculation incurs costs for installation and maintenance and an incremental cost for computation. Some sensors, however, may contribute little or no improvement to the forecast accuracy. We incrementally construct sets of sensors until we arrive at a set for which no superset produces a better forecast. Then we construct a successive series of subsets, such that forecast accuracy degrades slowly. As each sensor is removed, on the one hand, the forecast error increases, so the energy costs may increase for a given controller. On the other hand, the costs for installing sensors and for computing models are reduced. By considering this tradeoff over the the series of sets, an optimal set of sensors can be found to be used with that controller.

Peer-review under responsibility of the Conference Program Chairs.

Keywords: Home Sensor Network, Temperature Forecasting, Internet of Things, Feature Selection, Energy Efficiency, Model Predictive Control

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]. Pan *et al.*[3] point out that, due to thermal inertia, it is more efficient to maintain temperature in a room or building than to raise or lower the temperature. Accurate temperature forecasts can help reduce energy usage in buildings by using future values of temperature when deciding whether or not to activate the HVAC[4]. Moreno *et al.*[5] achieve estimated energy savings of 20% in a realistic situation based on the presence of persons in a room. Yuan *et al.*[6] achieve 20% savings while exploiting thermal inertia when assigning rooms for meetings by scheduling contiguous meetings in the same room.

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

1877-0509 © 2017 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.321

Model Predictive Controllers (MPC), 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[7]. The prediction horizon used in practice depends on how much data is needed by the HVAC controller to achieve acceptable comfort while reducing energy consumption.

While the costs savings may be significant, the overhead and operational costs associated with MPC may discourage adoption. These costs include the installation and maintenance of the sensing devices, a wireless sensor network, and the computational cost of modelling temperature as a function of the data generated by the sensors. To encourage wider adoption of MPC, in this paper we seek to reduce these associated costs. Specifically, we identify sensor data with little influence on forecast accuracy.

In the remainder of the paper we review the sensor data related to temperature forecasting reported for the house we study, we discuss the nature of the search, provide a best-first search procedure to select sensors, and compare the outcomes as we varying the history horizon, forecast horizon and the error metric. We report on related work from this data, and conclude with recommendations for using our results in both new and existing installations.

2. Background

2.1. Data from a Smart Home

The SML House[4] competed in the Solar Decathalon 2012 competition[8], using 88 sensors and 49 actuators. In this paper and in our previous work[9,10], we use a publicly available subset of this data[11], reporting values during March and April 2012 from 18 sensors every quarter-hour.

The sensors reported are as follows:

1. Wi – wind speed
2. Tw – twilight indictator
3. TP – predicted temperature
4. TL – living room temperature
5. TD – dining room temperature
6. T – external temperature
7. SW – sun on the west wall
8. SS – sun on the south wall
9. SE – sun on the east wall
10. Pcp – precipitation
11. P – sun irradiance measured by a pyranometer
12. LL – lights in the living room
13. LD – lights in the dining room
14. HL – humidity in the living room
15. HD – humidity in the dining room
16. H – external humidity
17. CL – carbon dioxide sensor in the living room
18. CD – carbon dioxide sensor in the dining room

2.2. Linear and Lasso Regression

Our forecasting methods are based on linear regression defined as follows. Given a set of independent variables $x_1, ..., x_n$ and a dependent variable y of interest that we want to forecast, we seek parameters $\beta_0, ..., \beta_n$ so that $\beta_0 + \Sigma_{i=1}^{n} \beta_i x_i$ 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 to minimize the residual sum of squares (RSS):

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

Lasso regression[12] minimizes RSS + $\lambda\ \Sigma_{j=1}^{m} |\beta_j|$ where λ is a tuning parameter that balances the emphasis between reducing error and using small β coefficients, Some β may reduce to zero, which deselects that variable x, thus endowing lasso regression with feature selection. For lasso regression, we use the R library `glmnet`[13,14,15].

2.3. Feature Selection

It may occur that too many independent variables, or features, confounds a forecast model. Irrelevant details overwhelm the modelling technique, which prevents it from computing an accurate forecast. Feature selection, the process of selecting specific features from which to build a model, is roughly divided into wrapper techniques, filter methods, and embedded methods. Wrapper techniques enumerate various combinations of features, and measure the accuracy of the resulting models, selecting that combination that exhibits the best error. Filter techniques measure the usefulness of features using computationally fast metrics. Embedded techniques identify useful features during the modelling process as a by-product. Lasso regression is an example.

In this paper we focus on wrapper technques that are guided by best-first provided by the R library `FSelector`[16] and embedded techniques, using lasso regression.

3. Models Using Lagged Sensor Readings

When creating a model from which to forecast temperatures, we provide multiple historical readings from each sensor. Given a history of b time periods, where readings are taken every quarter-hour, we provide $b + 1$ lagged readings from each of s sensors, which includes the current period at lag 0. Let $x_{k,t}$ be the t^{th} observation for sensor k counting from the first observation at time $t = 1$, as it appears in the training data. Let y_t be the internal temperature the house at time t. We are given observations over the m time periods in the training data. We create a linear a model for each future period f. We define the RSS as

$$\mathrm{RSS}(f) = \Sigma_{t=b+1}^{m} (\beta_{f,0} + \Sigma_{g=0}^{b} \Sigma_{k=1}^{s} \beta_{f,k,g} x_{k,t-g} - y_{f+t})^2$$

In this equation, t starts at b+1 because there are no observations for the lagged readings for the first b data points. Using lasso regression, we choose values for the coefficients $\beta_f = \{\beta_{f,0}\} \cup \{\beta_{f,k,g} \,|\, g = 0, \ldots, b,\ k = 1, \ldots, s\}$ where g identifies the lag and k identifies the sensor. The coefficients in β_f specify a model for each future interval f. We use two different forecast horizons; h is either 12 or 48 future time periods, i.e. 3 or 12 hours.

The coefficients are computed on the training data which is the first 2/3 of the data. Once they are computed, we switch over to using test data, which is the final 1/3 of the data. Thus x and y below refer to observations in the test data and m to the number of observations in the test data. We report the RMSE for each future interval f. In our experiments $f = 1, \ldots, 12$ for forecasts three hours into the future, and $f = 1, \ldots, 48$ for forecasts to 12 hours.

$$\mathrm{RMSE}(f) = \sqrt{1/(m-b)\, \Sigma_{t=b+1}^{m} (\beta_{f,0} + \Sigma_{g=0}^{b} \Sigma_{k=1}^{s} \beta_{f,k,g} x_{k,t-g} - y_{f+t})^2}$$

We report error metrics on all forecasts f over the forecast horizon h, including Mean RMSE $= 1/h\, \Sigma_{f=1}^{h} \mathrm{RMSE}(f)$ and Maximal RMSE = $max_f \mathrm{RMSE}(f)$.

4. Useful and Confounding Sensors

In our experiments we consider various sets of sensors and, from each, we measure the error from a forecast model based on the data from those sensors. To measure error, with the exception of the selected set of sensors, we hold all other factors fixed, including the training and test data, the size h of the forecast horizon and the number b of back observations. Thus the error from the model is a function only of the set of sensors.

It will often occur that one sensor in a set of sensors is useful in that it provides predictive power. Let S be a set of sensors and let a and b be individual sensors. We say a is *useful* in S when the error from $S \setminus \{a\}$ is greater than the error from S. If a is useful in S then $a \in S$. It may also occur that two sensors each provide that same predictive power, for instance when they report similar information. In this case we can use either one. More precisely, we say

a and b are *interchangeable* in S when a and b are useful in S and the error from $S \setminus \{a\}$ is the same as the error from $S \setminus \{b\}$.

The definitions in this paper are relative to some tolerance, below which forecast error is insignificant. We do not define this tolerance here, but note that it will be determined by the model predictive controller as follows: If an increase in the forecast error does not affect the controller's ability to save energy, then that increase is below the tolerance. In this paper we speak informally and understand an error to be greater than another when the difference exceeds this tolerance, and likewise say that two errors are the same when their difference falls below this tolerance. In these experiments, since we are not measuring the performance of a controller, we take the tolerance to be 0.

Note that useful and interchangeable are defined with respect to a set of sensors. We may find that while a is useful in $S \setminus \{b\}$, a is not useful in $S \cup \{b\}$. For instance this will happen when a and b are interchangeable in S.

We may also observe that including a sensor in a model gives rise to a higher error. This can happen when the sensor leads us "down the garden path", for instance, when it appears to be correlated to the observed temperature in the training data, but oppositely correlated in the test data. We say that *a confounds S* when the error from $S \setminus \{a\}$ is lower than the error from $S \cup \{a\}$.

We may also observe sensors that together increase accuracy but individually do not. This can happen when the model uses an interaction between the sensors. Suppose the laundry is always done on Saturday and no other day, and starts when someone enters the laundry room on Saturday. Suppose one of the sensors reports the day of the week and another reports motion in the laundry room. Then the modeller may recognize a heating event – for the room heats up when the laundry is done – occurs when both sensors are activated. In this case, if the modeller associated a heating event just based on motion in the laundry room, regardless of the day of the week, it would be misled on the non-Saturdays, and the model's error would increase. Likewise it would be misled by associating a heating event with Saturday for those weeks where no laundry was done. Thus the laundry room motion sensor and the day of week sensor each individually confounds the model. However, together they improve the model. We say that two sensors a and b are *co-dependent* in S if individually each of a and b confound $S \setminus \{a, b\}$, but the error of $S \cup \{a, b\}$ is smaller than the error of $S \setminus \{a, b\}$.

We seek a set S^* of sensors that has minimal error among the power set of sensors. This implies all sensors in S^* are useful, and that all sensors not in S^* confound S^*. We say that a set of sensors *gracefully degrades* if we can remove one sensor at a time in some ordering, such that the error always increases. To guide the cost-benefit analysis, our goal is to create a gracefully degrading set of sensors. Given the tolerance for some controller, we advocate finding S^* and then removing any sensors while the reduction in error is below this tolerance.

5. Ordering Sensors by Influence

Our goal is to identify sensors that should be included in the model. Because we use lagged data in our model, each sensor provides many predictors in the regression, one for each quarter-hour of historical observations. For a given sensor, we may consider whether to include all of the predictors arising from this sensor, some of them, or none of them. This leads to a large search space. For instance, given one hour of lagged observations (plus the current observation) for each of 18 sensors, gives $5 \times 18 = 90$ predictors in the model. This gives rise to 2^{90} sets of predictors, which is clearly infeasible to search entirely. We also want to consider two hours hours of readings per sensor, but to avoid searching a space of 2^{162} sets of sensors, which would take us almost 10^{42} years to search if we could consider one set each second.

We rely on lasso regression, which selects features among the predictors in the regression. Since lasso feature selection is in place, we need only consider sensor selection so the search space is reduced from 2^{90} to 2^{18}, and is independent of the number of lagged readings per sensor. A complete search would take about 3 days if we could consider one set per second. We simplify it further by employing best-first search, which is a variant of bottom-up search that limits non-deterministic choices and is guided by a heuristic. Our heuristic prefers lower forecast error.

Algorithm 1 Best-first search: Initially the empty set of sensors is made available. The model for this set simply predicts the mean temperature. The search proceeds by non-deterministically selecting one of the available sets S with relatively low error. Non-deterministically, a new sensor a is selected. If a is useful in $S \cup \{a\}$, then $S \cup \{a\}$ is made available for future consideration. The non-deterministic choices considered at each of these two choice points are guided by the heuristic, and limited by FSelector to about five choices. After all such selections are made, the search concludes. Among the sets that were considered, the set with minimal error, S', is taken as an estimate of S^*.

Because the heuristic guides the search toward the most promising parts of the search space, good estimates of S^* are expected. S' is confounded by all sensors not in S', so it is a local minimimum. However, the non-determinism is controlled, so the search space is not entirely explored, and S' is not guaranteed to be a global minimum.

Given the sets that were considered by best-first search, we use a second algorithm to generate a sequence of these sets with gradually increasing error.

Algorithm 2 Construct the sensor sequence: Let $S_1 = S'$, which is the set with lowest error. Let $i = 1$ and define S_{i+1} as the set with lowest error that is both a subset of S_i and a considered set. Proceed to increment i and compute the next set S until S_i is empty. Report the sequence of S's and the sequence of set differences between them. In most cases the set differences will be individual sensors.

In the next section we consider the effectiveness of this best-first search using the data of the SML house. There is no guarantee that Algorithm 1 will deliver the overall best set S^*, there is no guarantee that Algorithm 2 will generate the best sequence. However, Algorithm 2 delivers sequences that degrades gracefully, and therefore can be used to guide the cost-benefit analysis.

6. Experimental Results

We ran experiments using four readings per sensor, shown in Table 1, and again using eight readings per sensor, in Table 2. In each we varied the forecasting horizon to three and twelve hours into the future. We measured both maximal and mean RMSE over the forecast horizon.

Consider the example from Table 1(a), where we used four historical observations per sensor, generated three-hour forecasts, and measured maximal RMSE. Starting from the empty set, the search in Algorithm 1 consider sets up to about 10 sensors. Overall it considered 135 sets of sensors, which is a sharp reduction from the possible $2^{18} = 262,144$ sets. The minimal error occurs with nine sensors: Wi, Tw, TL, TD, T, SW, SE, CL, and CD, so this is our estimate of S^*. The other nine sensors: TP, SS. Pcp, P, LL, LD, HL, HD, and H, confounded this modeller. Using Algorithm 2, we progressively remove sensors from the set to increase the error gradually. The error increases only by 0.0021 °C if we ignore CL, the carbon dioxide sensor in the living room. Another small increase, 0.0028° C, occurs if we ignore the carbon dioxide sensor in the dining room.

We observe some trends in the results. All of our tabular results degrade gracefully. The errors in Table 2 are slightly smaller than those in Table 1. The better accuracy in Table 2 arises from the four additional historical values from each sensor. Longer forecast horizons lead to larger errors. Forecasts twelve hours into the future gives rise to errors about 4× to 5× larger than forecasts for the next three hours. The selected best sets of sensors is smaller when considering forecasts for the longer period, which suggests some factors have influence over the temperature for a brief period. Maximal RMSE is larger than mean RMSE, but within a factor of about two. The choice of error metric sometimes led to larger sets S^* of sensors and sometimes smaller.

No best model made use of H, HD, HL or LL, which are, respectively, the humidity externally, in the dining room and in the living room, and the lighting in the living room. Based on this analysis, we would not recommend installing these sensors in this house for the purpose of forecasting temperature.

7. Related Work

The SML team reports [17] accuracy when forecasting temperature differences over future quarter-hour intervals, using data from two of their 88 sensors: internal temperature and sun irradiance, as well as a time categorical variable. Using these sensors, forecasts for each quarter-hour over three hours were generated using a combination of forecast models based on ANNs. They used this selection of sensors in future work [4], which also forecasts temperature differences with a forecast horizon of 48 hours. They modeled temperature differences, while we modeled temperatures, so accuracy measures are not comparable.

While no research yet has been done to determine whether the same sensors will help with temperature forecasts and temperature difference forecasts, one might the same sensors to be useful for both. Therefore it is interesting to see if the same sensors helped in both cases. The SML team explored subsets of sensors, from among this set:

Table 1: Sensors selected by Best-First Search computing 3 hour and 12 hour forecasts, using Mean and Maximal RMSE, and based on 4 historical readings per sensor.

(a) Best-First Search for the next 3 hours

Maximal RMSE	Sensors	Remove Next
0.4589	Wi+Tw+TL+TD+T+SW+SE+CL+CD	CL
0.461	Wi+Tw+TL+TD+T+SW+SE+CD	CD
0.4638	Wi+Tw+TL+TD+T+SW+SE	Wi
0.4662	Tw+TL+TD+T+SW+SE	TD
0.4951	Tw+TL+T+SW+SE	Tw
0.5219	TL+T+SW+SE	SW
0.5403	TL+T+SE	T
0.5886	TL+SE	SE
0.68	TL	

(b) Best-First Search for the next 12 hours

Maximal RMSE	Sensors	Remove Next
1.8598	Tw+TP+T+SW+LD	TP
1.9032	Tw+T+SW+LD	SW
1.975	Tw+T+LD	Tw
2.1399	T+LD	LD
2.3384	T	

(c) Best-First Search for the next 3 hours

Mean RMSE	Sensors	Remove Next
0.2404	Tw+TP+TL+TD+T+SW+SE+CL+CD	CL
0.2429	Tw+TP+TL+TD+T+SW+SE+CD	CD
0.2434	Tw+TP+TL+TD+T+SW+SE	Tw
0.2488	TP+TL+TD+T+SW+SE	T
0.2672	TP+TL+TD+SW+SE	SW
0.2827	TP+TL+TD+SE	TD
0.284	TP+TL+SE	TP
0.299	TL+SE	SE
0.3407	TL	

(d) Best-First Search for the next 12 hours

Mean RMSE	Sensors	Remove Next
1.1539	Tw+TP+TL+T+SW+LD+CL	TP
1.1566	Tw+TL+T+SW+LD+CL	CL
1.1598	Tw+TL+T+SW+LD	SW
1.1876	Tw+TL+T+LD	LD
1.2552	Tw+TL+T	T
1.3545	Tw+TL	Tw
1.4215	TL	

internal temperature (TD and TL), irradiance (P), internal humidity (HD and HL), and precipitation (PCP). Based on their results, a selection of three sensors gave the lowest errors: internal temperature, solar irradiance, and a time-categorical variable. Our results show some agreement: temperature was the most important, and humidity was not of any help. We found that PCP was of minor help. Unlike their result, we found that the pyranometer was not of any help. They seem not to have considered sensors that we found were helpful, including the sun on each wall, the CO_2 in the living and dining room, and twilight.

We found variability in the selections for different history and forecast lengths, and for different error metrics. We found that more sensors were helpful for short term forecasts, than for long term. Since the selections differed for forecasts of temperature over various horizons, perhaps the selections should also differ for forecasts of temperature *differences* over various horizons.

Feature extraction shares some similarities with feature selection. Feature extraction is the process of defining new features from existing ones, by selecting those features with good predictive accuracy, and repackaging them into linear combinations that are considered new features. Partial least squares and principal component analysis are two feature extraction techniques[18,19].

We used the same SML data for partial least squares and principle components[10]. Using four historinal readings per sensor, we found the RMSE forecast error for both methods to be about 0.7 for three-hour forecasts whereas the comparable mean RMSE values in this paper range from 0.24 to 0.34. Likewise for twelve-hour forecasts, the RMSE for the feature extraction methods was about 1.7 for twelve-hour forecasts, and ranged from 1.15 to 1.42 in this paper. The results were similar for eight historical readings per sensor. Thus, lasso regression and best-first search exhibit better forecast accuracy than these feature extraction methods for temperature forecasting.

Table 2: Sensors selected by Best-First Search computing 3 hour and 12 hour forecasts, using Mean and Maximal RMSE, and based on 8 historical readings per sensor.

(a) Best-First Search for the next 3 hours

Maximal RMSE	Sensors	Remove Next
0.4559	Wi+Tw+TP+TL+TD+T+SW+SS+SE+Pcp+P+LD+CD	Pcp
0.4572	Wi+Tw+TP+TL+TD+T+SW+SS+SE+P+LD+CD	TP
0.4592	Wi+Tw+TL+TD+T+SW+SS+SE+P+LD+CD	CD
0.4643	Wi+Tw+TL+TD+T+SW+SS+SE+P+LD	P
0.4752	Wi+Tw+TL+TD+T+SW+SS+SE+LD	SW
0.4788	Wi+Tw+TL+TD+T+SS+SE+LD	Wi
0.4864	Tw+TL+TD+T+SS+SE+LD	SE
0.5076	Tw+TL+TD+T+SS+LD	SS
0.5326	Tw+TL+TD+T+LD	TD
0.545	Tw+TL+T+LD	LD
0.5731	Tw+TL+T	T
0.6207	Tw+TL	Tw
0.6681	TL	

(b) Best-First Search for the next 12 hours

Maximal RMSE	Sensors	Remove Next
1.8348	Tw+TL+TD+SW+CL	CL
1.8465	Tw+TL+TD+SW	TL
1.8637	Tw+TD+SW	SW
1.9678	Tw+TD	Tw
2.2423	TD	

(c) Best-First Search for the next 3 hours

Mean RMSE	Sensors	Remove Next
0.2499	Tw+TP+TL+TD+T+SW+SE+Pcp+CL+CD	CD
0.2509	Tw+TP+TL+TD+T+SW+SE+Pcp+CL	CL
0.2517	Tw+TP+TL+TD+T+SW+SE+Pcp	Pcp
0.254	Tw+TP+TL+TD+T+SW+SE	SE
0.2738	Tw+TP+TL+TD+T+SW	T
0.2867	Tw+TP+TL+TD+SW	Tw
0.3027	TP+TL+TD+SW	SW
0.3142	TP+TL+TD	TD
0.322	TP+TL	TP
0.3486	TL	

(d) Best-First Search for the next 12 hours

Mean RMSE	Sensors	Remove Next
1.1303	Tw+TL+TD+SW+CL+CD	CD
1.1367	Tw+TL+TD+SW+CL	CL
1.1477	Tw+TL+TD+SW	TD
1.169	Tw+TL+SW	SW
1.2395	Tw+TL	Tw
1.3838	TL	

8. Conclusion

A model predictive controller can achieve significant savings by using an accurate temperature forecast when determining whether or not to engage HVAC systems. Temperature forecasts are informed by sensor data. We propose a cost-benefit analysis that balances the cost arising from installation, operating and computation against the benefit of saving energy. A sensor's cost exceeds its benefit if it does not improve forecast accuracy by an amount sufficient to be useful to the controller.

The method we describe generates accurate temperature forecasts using lasso regression. It uses a best-first search technique to incrementally consider larger sets of sensors until no additional sensor improves the forecast accuracy. It then reduces this set by removing sensors incrementally and reporting the resulting sequence of forecast errors. If we assume that energy savings increase with forecast accuracy, this sequence of sets of sensors should help finding the optimal set of sensors.

Our system computes a gracefully degrading set of sensors for different situations, depending on the length of the forecast horizon, the number of historical observations, and whether the controller performs better with a lower mean error or a lower maximal error. Our findings indicate that the selection of sensors will be affected by these factors. In a new installation, we propose to temporarily install a large set of sensors, and to collect readings from these sensors over several weeks. Then it should be possible to determine which sensors to permanently install. Alternately, in an existing installation, the maintenance and computation costs may be reduced by removing sensors that are not providing benefit. The same ordering can guide this selection.

Our experiments show accuracy increases as more data is available for forecasting. Shorter term forecasts are more accurate than longer term forecasts, and derive benefit from more sensors than longer term forecasts.

While we have used lasso regression over lagged data as the underlying modelling technology, our technique based on best-first search can be applied to any underlying modelling technology.

Acknowledgements

The authors gratefully acknowledge the financial support of their organizations, especially Zayed University's RIF fund.

References

1. P. Morosan, R. Bourdais, D. Dumur, J. Buisson, Building temperature regulation using a distributed model predictive control, Energy and Buildings (2010) 14451452.
2. L. Pérez-Lombard, J. Ortiz, C. Pout, A review on buildings energy consumption information, Energy and Buildings 40 (2008) 394398.
3. D. Pan, D. W. Y. Yuan, X. Xu, Y. Peng, X. Peng, P.-J. Wan, Thermal inertia:towards an energy conservation room management system, in: A. Greenberg, K. Sohraby (Eds.), INFOCOM, IEEE, 2012, pp. 2606 – 2610.
4. F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, J. Pardo, On-line learning of indoor temperature forecasting models towards energy efficiency, Energy and Buildings 83 (2014) 162–172.
5. M. V. Moreno, M. A. Zamora, A. F. Skarmeta, User-centric smart buildings for energy sustainable smart cities, Transactions on Emerging Telecommunications Technologies 25 (1) (2014) 41–55. doi:10.1002/ett.2771.
URL `http://dx.doi.org/10.1002/ett.2771`
6. Y. Yuan, D. Pan, D. Wang, X. Xu, Y. Peng, X. Peng, P.-J. Wan, A study towards applying thermal inertia for energy conservation in rooms, ACM Trans. Sen. Netw. 10 (1) (2013) 7:1–7:25. doi:10.1145/2529050.
URL `http://doi.acm.org.proxy.hil.unb.ca/10.1145/2529050`
7. J. Álvarez, J. Redondo, E. Camponogara, J. Normey-Rico, M. Berenguel, P. Ortigosa, Optimizing building comfort temperature regulation via model predictive control, Energy and Buildings 57 (2013) 361 – 372.
8. United States Department of Energy, Solar Decathlon Europe Competition, `http://www.solardecathlon.gov` (2012).
9. B. Spencer, F. Al-Obeidat, Temperature forecasts with stable accuracy in a smart home, in: SEIT Sustainable Energy Information Technology Conference, 2016. doi:10.1016/j.procs.2016.04.160.
10. B. Spencer, F. Al-Obeidat, O. Alfandi, Short term forecasts of internal temperature with stable accuracy in smart homes, International Journal of Thermal and Environmental Engineering 13 (2) (2016) 81–89.
11. UCI, Sml2010 data set, `https://archive.ics.uci.edu/ml/datasets/SML2010`.
12. R. Tibshirani, Regression shrinkage and selection via the lasso: a retrospective, Journal of the Royal Statistical Society: Series B (Statistical Methodology) 58 (1) (1996) 267–288.
URL `http://www.jstor.org/stable/2346178`
13. J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for generalized linear models via coordinate descent, Journal of Statistical Software 33 (1) (2010) 1–22.
URL `http://www.jstatsoft.org/v33/i01/`
14. N. Simon, J. Friedman, T. Hastie, R. Tibshirani, Regularization paths for cox's proportional hazards model via coordinate descent, Journal of Statistical Software 39 (5) (2011) 1–13.
URL `http://www.jstatsoft.org/v39/i05/`
15. J. Friedman, T. Hastie, N. Simon, R. Tibshirani, Package `glmnet`: Lasso and Elastic-Net Regularized Generalized Linear Models Ver 2.0-, `https://cran.r-project.org/web/packages/glmnet/glmnet.pdf` (March 15 2016).
16. P. Romanski, L. Kotthoff, Package FSelector Selecting Attributes. R package version 0.21, https://cran.r-project.org/web/packages/FSelector/FSelector.pdf (August 2016).
17. F. Zamora-Martínez, P. Romeu, P. Botella-Rocamora, J. Pardo, Towards energy efficiency: Forecasting indoor temperature via multivariate analysis, Energies 6 (9) (2013) 4639. doi:10.3390/en6094639.
URL `http://www.mdpi.com/1996-1073/6/9/4639`
18. R. W. Bjørn-Helge Mevik, The `pls` Package: Principal Component and Partial Least Squares Regression in R, Journal of Statistical Software 18 (2) (2007) 1–24. doi:10.18637/jss.v018.i02.
19. B.-H. Mevik, R. Wehrens, K. H. Liland, pls: Partial Least Squares and Principal Component Regression, r package version 2.5-0 (2015).
URL `https://CRAN.R-project.org/package=pls`