

# Robust sensor data collection over a long period using virtual sensing

Rachel Cardell-Oliver  
The University of Western Australia  
Perth, Australia  
rachel.cardell-oliver@uwa.edu.au

Chayan Sarkar  
SICS Swedish ICT  
Stockholm, Sweden  
chayan.sarkar@sics.se

## ABSTRACT

Virtual sensing is a technique in which sensor values in a sensor network are estimated some of the time instead of taking direct measurements. Virtual sensing conserves energy on the sensor nodes, and it can be used for recovering missing data due to node or network failure, and for estimating values where permanent sensor installation is infeasible. This paper considers the novel and challenging problem of designing an adaptive long range virtual sensing (LRVS) framework that is able to estimate values for days or months. The framework does not require any *a priori* modeling of generating functions for the sensor time series. A new correlation metric, called “nearest hour of the day neighbor”, finds similarity in multiple time series that is able to arbitrate the hidden contexts within the data. LRVS is evaluated on data from a building-monitoring sensor network with over 100 heterogeneous, noisy sensors from different suppliers.

## Keywords

virtual sensing, data collection, spatial correlation

## 1. INTRODUCTION

Estimating sensor data without taking direct measurements is a valuable technique for sensor network management because it can reduce the amount of data traffic significantly [11, 13]. However, existing methods are suitable for estimating sensor data for only a short time frame from a few minutes to a few hours. Thus, these techniques are inadequate for estimating sensor readings over a long time frame ranging from a full day to a few weeks. Moreover, estimation over a long range is not only beneficial for energy-efficient network management, but in a number of other circumstances.

**Missing data recovery:** In real-world deployments, sensors or communications may fail to report their readings for periods of days or weeks. A method for long range estimation

allows the missing data to be reconstructed within some margin of error.

**Energy-efficient operation:** Having accurate long range estimations allows certain sensors to remain dormant for long periods, so saving the energy used for measurement and transmission and also reducing congestion of communication bandwidth. As the Internet of Things allows sensor networks to be scaled up by an order or magnitude more nodes, there is a need to reduce the energy and bandwidth used for data gathering.

**Replacement for temporary sensors:** The expense or inconvenience of measuring a certain parameter may mean that measurements are only available temporarily. For example, iLTC [14] is an automated indoor lighting control system that requires light intensity measurement at various parts of the building on a continuous basis. Instead of deploying sensor nodes densely, the system uses temporary sensors and later those sensors are replaced with estimation models.

**Event detection by indirect sensing:** A sensor model trained using trusted, stable sensors can be used to identify deviations from expected behavior. For example, if the thermal performance of a smart home is measured when there are no occupants in the building, then the effect of human habitation can be identified as the difference between measured actual conditions (maybe only sporadically monitored) and the values estimated by the model trained in controlled conditions.

Ensuring high data accuracy for a long range estimation is difficult because sensor values are affected by seasonal and diurnal trends. Furthermore, sensors are affected by hidden contexts, either natural or man made. For example, local weather conditions or human activities such as artificial heating or cooling affect sensor values. In order to address the long range estimation problem in sensor networks, we propose a long range virtual sensing (LRVS) framework that harnesses cross-correlation amongst sensors. The basic idea behind our LRVS estimation mechanism is to estimate data for a node by utilizing the sensed values from a correlated neighbor. Cross-correlation is a promising approach because sensors in the same system will all be subject to the same seasonal, diurnal and hidden contexts. Though Sarkar *et al.* [12] proposed a cross correlation based estimation model, it finds correlated neighbors statistically without considering the short-term hidden patterns within the time series data. On the other hand, LRVS investigates various neighbor selection schemes in order to capture the patterns in the data from different sensor nodes.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

TSA '16 Workshop on Time Series Analytics and Applications, Hobart

© 2016 ACM. ISBN 123-4567-24-567/08/06.

DOI: 10.475/123-4

In this article we make the following contributions.

- We propose an adaptive framework for estimating sensor values over long time periods (weeks to months) that does not require any *a priori* modeling of generating functions for the sensor time series.
- We propose a new correlation metric, called “nearest hour of the day neighbor”, to find similarity in multiple time series that is able to arbitrate the hidden contexts within the data.
- We evaluate our approach on data from a building-monitoring sensor network with over 100 sensors from different suppliers that has been running since August 2014. The results demonstrate that it is feasible to build robust estimators for long-time estimation of real-world data sets with noise, missing values, different types of sensed phenomenon, and varied measurement intervals.

The rest of the article is organized as follows. Section 2 discusses related work. Then we describe our proposed method in detail in Section 3. We provide extensive evaluation of our estimation mechanism in Section 4. Finally, we conclude the article in Section 5.

## 2. RELATED WORK

A popular approach for reducing the amount of data traffic in a sensor network is to use clustering and data aggregation techniques [2, 16]. The basic idea is that a node, selected as cluster-head, collects data from its neighboring (member) nodes and sends the accumulated data to the sink. The nodes use some aggregation technique to create a single data packet to transmit multiple data points [10]. This is particularly beneficial for a multihop networks, where intermediate nodes help to forward the data packets to the sink. As clustering induces fewer data packets within the network, it reduces the overall energy toll on the nodes.

On the other hand, time series analysis of sensed data enables sparse data point reporting. The sparse data points are complemented by accurately predicting the unreported data points. Sensor data prediction increases the operational efficiency even further than can be achieved by clustering and aggregation [8]. Time series analysis of sensor data reveals that often there is a strong correlation in the data [6]. Existing approaches have utilized this correlation to predict the data accurately.

Typically, environmental data such as temperature and humidity do not change abruptly within a short time span, and there is a strong correlation between the current value and past values. Capturing this autocorrelation in the sensor data, Santini *et al.* [11] presented a sensor data prediction mechanism that reduces the number of data transmissions, thus energy consumption significantly. The idea is to create and maintain an identical prediction model at the node and the sink. The node predicts the data and compares it with the actual measurement. Only if the predicted values drift significantly compared to the measured value (decided based on a pre-defined threshold) is the measurement transmitted. Otherwise, the sink predicts the data point.

Auto-correlation methods can achieve high accuracy when predicting readings a short time into the future (typically an hour or so). But their accuracy decreases sharply when

predicting values over a longer time scale (days to weeks). Using additional basis functions to model properties such as diurnal variation and spatial correlation across the nodes can improve accuracy as shown by Guestrin *et al.* [5]. Unlike the autocorrelation based technique, where data is transmitted when it changes significantly, here the updated model parameters are transmitted. Though these techniques reduce the number of data transmissions significantly, the nodes need to remain active and continuously monitor changes in sensed data.

Deshpande *et al.* [4] proposed a model driven data acquisition technique where some of the sensor nodes can be kept in dormant state where they are not required to sense. A similar technique is also proposed by Chu *et al.* [3] where, by exploiting the spatial correlation among the nodes, the data is predicted for the dormant nodes. However, a major drawback of these techniques is that they assume *a priori* and static correlation among the nodes.

To overcome this problem, Le *et al.* [9] proposed an adaptive model selection for time series prediction. Recently, the virtual sensing framework (VSF) [13] is proposed that is also not dependent on *a priori* knowledge about the correlation among the nodes. It learns the correlation pattern dynamically with low overhead. By exploiting the autocorrelation and cross correlation among the nodes, it can predict sensor data with high accuracy. It also adapts to changes in node-correlation over time.

Most of the existing works [7, 13, 15] mainly focus on energy-efficient network operation by reducing data transmission. As their goal is to predict within a short time range, the chance of out of bound prediction error is minimal. But these methods are inadequate for long range sensor data estimation. In contrast, we propose a long-range virtual sensor (LRVS) framework that can predict sensor readings over longer time periods.

## 3. LRVS FRAMEWORK

In this section, we discuss our low error estimation mechanism for long durations, called LRVS. But before describing the time series modeling and estimation functions, let us describe the system model.

### 3.1 System model

LRVS does not assume any specific system model and it can be integrated with any sensor network. However, the estimation mechanism can only be applied where there is some spatial correlation in the sensed data. A sensor network is represented as a collection of sensors  $S = \{s_1 \dots s_n\}$  each recording values  $s_i(t) = v_{it}$  at times  $t$ . All the sensed data is reported to a central node, called the sink node. Though the sensor nodes are often battery operated embedded devices with limited processing capability, the sink node has sufficient resources or is connected to a resource-rich computer. LRVS creates virtual sensors at the sink, which contain functions to estimate some of the sensed data.

LRVS divides the sensor nodes into two node groups, dependent and independent, where every dependent node has an independent “correlated” neighbor. Our spatial estimation approach learns the best correlated neighbor and estimation model for every dependent node using training data. During the estimation process, the learned model is used to estimate the values of each dependent sensor. The process starts by selecting a training period  $P$  and an estimation

period  $E$  from the sensor network observations. Then a dependent sensor  $s_d$  and a set of potential explanatory (independent) sensors  $S_E \subseteq S - \{s_d\}$  is chosen. This step can be repeated for as many dependent sensors as necessary. Each sensor  $s_e \in S_E$  is considered in turn as a potential estimator for  $s_d$ . One or more estimator functions  $f_i$  for  $s_d$  is trained using values of  $s_e$  from the training period  $P$ . The goal of the estimator function is for  $f_i(t) \approx s_d(t)$  for all  $t \in P$  (see Section 3.2 for details). The sensor  $s_e$  and estimator function  $f_i$  is selected based on the best possible accuracy among the potential explanatory sensors. During the estimation period, data from  $s_e$  and the estimation function  $f_i$  is used to calculate estimated values for  $s_d$  (see Section 3.3 for details).

During operation, LRVS alternates between training and estimation periods. Each time a new estimator sensor or function may be chosen, ensuring that the algorithm adapts to the current context. In application scenarios where data is missing from a node or the node is replaced (temporarily or permanently), its correlated neighbor and associated estimation function is created based on previously collected data (training data). The mechanism can also be applied for energy efficient network operation by assigning as many nodes as possible to the dependent set and choosing the highest ratio of estimation to training days that achieves acceptable estimation accuracy. Though the behavior of LRVS is similar to VSF [13], the estimation functions are significantly different.

### 3.2 Estimation Functions

Four types of estimation function are considered in this article – nearest neighbor, nearest linear neighbor, nearest cubic neighbor, and nearest time of day neighbor.

**Nearest Neighbor (NN)** is a baseline approach. To select  $s_e \in S_E$  for a target sensor  $s_d$ , choose the  $s_e$  that minimizes the estimation error (difference) between  $s_d$  and  $s_e$  during the training period  $P$ . The NN estimator simply sets

$$f(t) = s_e(t).$$

**Nearest Linear Neighbor (NLN)** is based on the cross-correlation as in VSF [13]. Least squares linear regression is used to select the estimator sensor  $s_e$  and constants  $\alpha$  and  $\beta$  to minimize the estimation error. The nearest linear estimator is

$$f(t) = \alpha \cdot s_e(t) + \beta.$$

**Nearest Cubic Neighbor (NCN)** is based on the distributed regression method described in [5]. It extends the linear spatial neighbor approach by using a cubic function on the hour-by-hour data. The hour of day modeling captures the diurnal cycles of sensors. We use  $t.hr$  to denote the hour of day of time  $t$ . A least squares approach is used to fit the model as

$$f(t) = \beta \cdot s_e(t) + \alpha_3(t.hr)^3 + \alpha_2(t.hr)^2 + \alpha_1(t.hr) + \alpha_0.$$

**Nearest Hour of Day Neighbor (NHN)** requires no *a priori* modeling of the generator functions for the sensor time series. It uses 24 points per day to characterize the distance between the target  $s_d$  and its estimator  $s_e$ . Let  $D(s_d, s_e, h)$  be the mean distance between sensors  $s_d$  and  $s_e$  at some hour of the day during the training period. That is,

$$D(s_d, s_e, h) = \text{mean}\{s_d(t) - s_e(t) \mid t \in P \wedge t.hr = h\},$$

where  $t.hr \in [0, 23]$  denotes the hour of day attribute of time  $t$ . The nearest hour of day estimator is

$$f(t) = s_e(t) + D(s_d, e_d, t.hr).$$

In this article, we use a model with 24 points, one for each hour of the day. But the approach generalises to use fewer or more time intervals to estimate the shape of a day of sensor readings.

### 3.3 Selection of Estimator Sensor

This section explains how the “best” estimator sensor is selected for each estimation method. For the first three estimators (NN, NLN, NCN), accuracy is measured by the well known Root Mean Squared Error (RMSE). The error is measured between the estimated value and the ground truth of the dependent sensor during time period  $P$ . To select the best estimator, chose the sensor  $s_e$  and estimation function  $f_i$  with the minimum RMSE using

$$RMSE(s_d, f_i, P) = \sqrt{\frac{\sum_{t \in P} (s_d(t) - f_i(t))^2}{\text{size}(P)}}.$$

The nearest hour of day (NHN) method selects the most stable estimator as the best. The standard deviation of the errors by hour of day is defined by

$$Sh(s_d, s_e, h, P) = \text{stdev}\{s_d(t) - s_e(t) \mid t \in P \wedge t.hr = h\}.$$

The mean standard deviation is used to select the most stable estimator sensor as

$$Err(s_d, s_e, P) = \text{mean}\{Sh(s_d, s_e, h, P) \mid h \in [0, 23]\}.$$

The best estimator  $s_e$  is selected as the one with the smallest variations in hourly estimates, i.e., the minimum  $Err(s_d, s_e, P)$ .

For comparing the estimates of different models, the R-squared goodness of fit model is used. The adjustment for model coefficients is not required here because, for our setting, the sample size (1000s of points) is much larger than the maximum number of model coefficients (maximum 24) and so the adjustment for model coefficients would be close to 0.

$$R^2(s_d, f, P) = 1 - \frac{\sum_{t \in P} (s_d(t) - f(t))^2}{\sum_{t \in P} (s_d(t) - \text{mean}(s_d))^2}.$$

## 4. EVALUATION

We evaluate LRVS using data from a building-monitoring sensor network. This network has been running continuously since August 2014 and comprises two buildings and their environs instrumented with over 100 sensors [1]. The heterogeneous network combines commercial sensors for temperature and relative humidity, bespoke sensors including temperature sensor profilers embedded through the walls of the buildings, temporary sensors that could be deployed only while the buildings were unoccupied, a local weather station and public data from the nearest Bureau of Meteorology weather station. Sensing intervals range from 5 minutes to 30 minutes. A subset of 32 sensors and 12 weeks (84 days) of data is selected for analysis. During these three months, from September to November 2014, the properties were monitored but unoccupied. The monitoring period include both summer and winter days (for Australia).

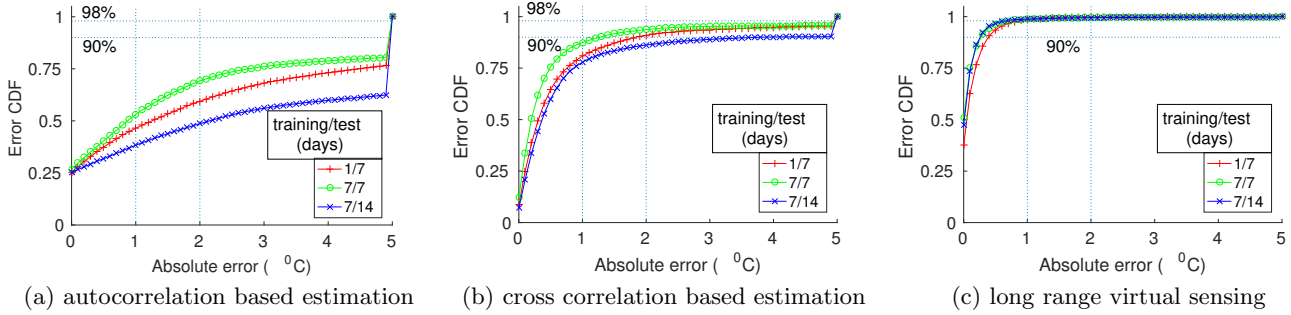


Figure 1: Estimation error comparison for various methods under different training and testing periods.

### 4.1 Estimation Accuracy

The primary goal of LRVS is to estimate sensor data accurately for a long time period. The existing method based on autocorrelation [11] performs poorly in case of long term estimation as evident from Figure 1a. The cumulative density function of estimation error across all the sensors shows that the error can be more than 2° C for up to 50% of the sensor readings. On the other hand, cross correlation based estimation can significantly improve the estimation accuracy [13]. Figure 1b shows that for up to 90% of the cases the error is within 1° C. However, as the testing period increases, the estimation accuracy decreases. Using the nearest hour of day neighbor based estimation, LRVS achieves a significantly higher accuracy as shown in Figure 1c. For up to 95% of the sensor readings LRVS achieves estimation error within 0.5° C.

### 4.2 Accuracy and Sensor Types

Figure 2 shows the results for the best estimator sensor for each of the four estimation methods on different groups of dependent sensors. There are 5 groups of dependent sensors measuring temperature and relative humidity: outside, within the wall structure, at ceiling level, head-height and relative humidity at ceiling level. It can be seen that for the wall, ceiling and head sensors, the RMSE is less than 1° C. for all four estimation methods. These sensor groups are similar to those used in previous studies: there are many sensors measuring the same phenomena (temperature) in similar situations. For this situation simply substituting the values of a similar sensor (i.e. NN) gives an estimate that is sufficiently accurate for many applications.

The building monitoring sensor network application has two sensor groups that are hard to estimate: outside temperature and relative humidity (RH). These types of sensors are hard to estimate for several reasons. Relative humidity has greater amplitude and its shape varies more from day to day than temperature. Outside temperature is hard to estimate because its amplitude is larger than any of the other temperature sensors. Another factor for estimating outdoor temperature and outdoor relative humidity may be that these sensors have the lowest sensing frequency, with one observation reported per 30 minutes.

Figure 3 shows estimation errors over time for one 14 day estimation period. The target sensor here is the hardest case of outdoor temperature. The best estimator sensor is chosen for each of the four estimator functions. The nearest hour of day (NHN) is the most accurate function in this

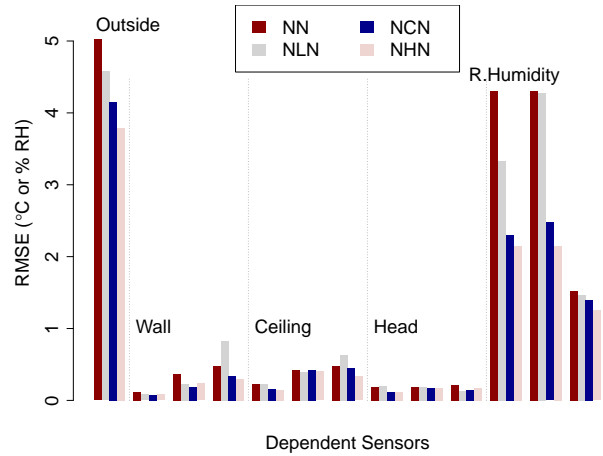


Figure 2: RMSE by estimation method for different types of sensors.

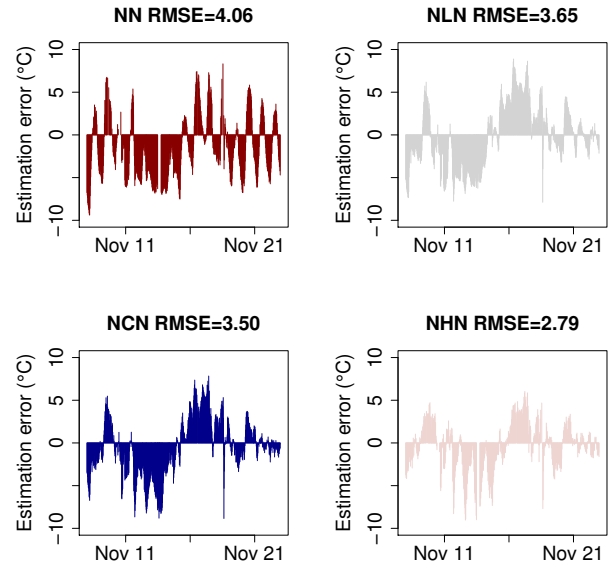


Figure 3: Time series of estimation errors for outdoor temperature.

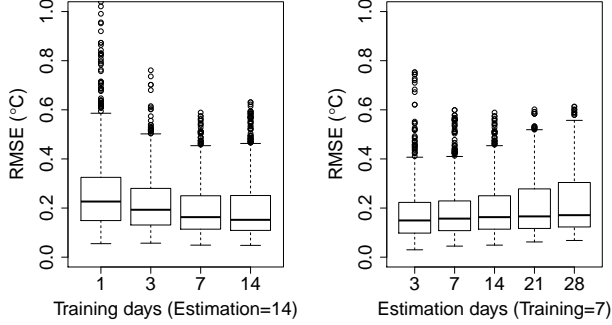


Figure 4: Accuracy by training and estimation periods.

case. Both the linear (NLN) and cubic (NCN) models show similar errors. Failure to account for time of day differences is seen in the diurnal cycle of the errors for NN.

### 4.3 Training and Estimation Periods

LRVS alternates between training and estimation periods. There is a trade-off to be made between the accuracy of the estimations and the energy-efficiency of the schedule. Short training periods and long estimation periods save the most energy, and are most flexible for recovering or replacing data. But long training periods and short estimation periods give the highest accuracy. Figure 4 shows the RMSE for a selection of training/estimation scenarios. Each box shows the range of RMSEs observed for the 27 sensors that could be estimated, each evaluated on 42 different windows from the dataset. The left hand figure compares different numbers of training days for a fixed estimation period of 14 days. As expected, as the training period is increased, the accuracy of estimation improves. This is likely because the effect of outlier readings is reduced by the availability of more normal data. In the right hand figure, the training period is fixed at 7 days and the accuracy for estimation periods from 3 to 28 days is shown. As the estimation period increases, the accuracy of estimations deteriorates slowly. These results suggest that a training period of 7 days and estimation period of 14 days provides a reasonable balance between energy-efficiency and accuracy.

### 4.4 Adaptive Training

One drawback of the existing algorithms is that they assume *a priori* and static correlation among the nodes. We address that problem by adaptively updating both the estimation method and estimating sensor over a series of time windows. In this section we investigate to what extent this strategy is necessary or beneficial.

The estimation algorithm was run with a 7/14 days training to estimation cycle over 4 windows for 32 target sensors. For each of the 32 target sensors, and each of the 4 windows we chose the best estimation method (NN, NLN, NCN, and NHN) and the best estimator sensor. This gave 128 (32 by 4) test cases, of which 117 could be estimated with acceptable  $RMSE < 5^\circ C$  or 5% humidity.

Table 1 shows how the choice of best estimator varied. Although NCN and NHN were most often selected as the best method, NH and NLN were the best estimator for a few cases. That is, all four methods were needed to obtain the

Table 1: Adaptive choice of estimation method and sensor.

Best Method	NN	NLN	NCN	NHN
Times selected from 117 cases	2	6	49	60
Estimator sensors	0/4	1/4	2/4	3/4
Times occurred from 32 sensors	1	14	10	7

Table 2: Comparison of training data selection methods for estimating the outdoor temperature sensor.

Strategy	RMSE ( $^\circ C$ )	$R^2$
Combined training days (56)	3.7	0.665
Selected training days ( $4 \times 14$ )	2.8, 3.0, 3.0, 4.1	0.784, 0.813, 0.866, 0.013

best accuracy. Only 14 sensors selected the same estimator sensor in each of their four test windows. No target had 4 different estimator sensors, but 2 or 3 different sensors were required in many cases. For example, for one ceiling sensor during 4 training windows the best estimation methods were NCN, NHN, NN, NCN and the best estimator sensors were 42, 30, 38 and 30 (respectively). These results demonstrate that the adaptive strategy of LRVS is beneficial and that both estimation method and sensor should be updated adaptively over time.

### 4.5 Replacement and Indirect Sensing

So far we have repeated the training and estimation windows comprising consecutive days with a new model being calculated for each training window. This approach is ideal for the scenarios of energy saving and recovering missing data. However, the scenarios of replacing temporary sensors and indirect sensing requires accuracy to be maintained for much longer time periods of months or years. In this section we consider the problem of how to select a stable estimator for these challenging scenarios.

The previous sections demonstrated that the choice of sensor and method for estimation should change over time, and that 7/14 days is the best cycle for training and estimation days. So how should the algorithm adapt in the situation when the days that need to be estimated are far away in time from the training days? We considered two possible strategies for this situation. One strategy is to train a model using all the available data for training, so that the trained method is optimised over as many situations as possible. Then use the learnt model to estimate values for the full estimation period. Alternatively, we can adapt the 7/14 scenario, by selecting 7 training days from the available data, learning an estimation model, and then applying that model to estimate 14 days selected from the estimation period. This process is repeated until all the days in the estimation period are covered. The 7 training days should be selected to be as similar as possible to the 14 days that are to be estimated.

Table 2 compares the effectiveness of these two strategies. For this experiment we used a sample of 56 days from the September to November 2014 data used so far as the training data. An additional 56 days selected from March 2015 to October 2015 were used to test estimation for the (hard to

estimate) outdoor temperature sensor. For strategy one, using all 56 training days together, the estimation RMSE is 3.7° C. But for strategy two, when the estimation period is partitioned into four 14-day windows and each is trained with the most similar 7 days from the training data, the RMSE improves to 3.2° C. The mean R squared goodness of fit statistic ( $R^2$ ) is slightly lower for the second strategy, because one of the four training windows has a very poor R squared. Overall, these results suggest that, if suitable training windows are selected, then the LRVS framework can also be used successfully for the challenging scenario of replacement or indirect sensing.

## 5. CONCLUSION

We have proposed an adaptive long range virtual sensing (LRVS) framework for estimating sensor values over long time periods of days to weeks. LRVS does not require *a priori* modeling of generating functions. A new correlation metric, called “nearest hour of the day neighbor”, was able to find similarity in multiple time series and to arbitrate the hidden contexts within the data. The framework was evaluated on data from a building-monitoring sensor network. The results demonstrate that it is feasible to build robust estimators for long-time estimates for real-world data sets with noise, missing values, different types of sensed phenomenon, and different measurement intervals. LRVS allows trade-offs to be made to achieve fit for purpose estimations for different applications. For example, for some applications the simple nearest neighbour estimation strategy gives sufficient accuracy, whilst others would require more accurate modelling with “nearest hour of the day neighbor” (NHN). Moreover, we showed that the best choice of estimator sensor and method changes over time, and that LRVS is able to adapt to these changes.

Areas for future work include evaluating the LRVS framework on a wider range of sensor types and measured phenomena and consideration of additional estimation methods. For example, the nearest hour of day method could be generalised to use fewer or more time intervals for estimation.

## 6. ACKNOWLEDGMENTS

The authors acknowledge the support of the Australian Research Council (ARC Linkage Grant LP140100375), the Western Australian Department of Housing and Go8-DAAD.

## 7. REFERENCES

- [1] C. Beckett, D. Ciancio, C. Huebner, and R. Cardell-Oliver. Sustainable and affordable rammed earth houses in Kalgoorlie, Western Australia: Development of thermal monitoring techniques. In *Australasian Structural Engineering Conference, July 9-11, Auckland, NZ, July 2014*.
- [2] C.-T. Cheng, K. T. Chi, and F. C. Lau. A clustering algorithm for wireless sensor networks based on social insect colonies. *IEEE sensors journal*, 11(3):711–721, 2011.
- [3] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong. Approximate data collection in sensor networks using probabilistic models. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 48–48. IEEE, 2006.
- [4] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, pages 588–599. VLDB Endowment, 2004.
- [5] C. Guestrin, P. Bodik, R. Thibaux, M. Paskin, and S. Madden. Distributed regression: an efficient framework for modeling sensor network data. In *Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on*, pages 1–10. IEEE, 2004.
- [6] H. Gupta, V. Navda, S. Das, and V. Chowdhary. Efficient gathering of correlated data in sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 4(1):4, 2008.
- [7] Y. Huang, W. Yu, C. Osewold, and A. Garcia-Ortiz. Analysis of pkf: A communication cost reduction scheme for wireless sensor networks. *IEEE Transactions on Wireless Communications*, 15(2):843–856, 2016.
- [8] H. Jiang, S. Jin, and C. Wang. Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 22(6):1064–1071, 2011.
- [9] Y.-A. Le Borgne, S. Santini, and G. Bontempi. Adaptive model selection for time series prediction in wireless sensor networks. *Signal Processing*, 87(12):3010–3020, 2007.
- [10] J. Lu, F. Valois, M. Dohler, and M.-Y. Wu. Optimized data aggregation in wsns using adaptive arma. In *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SensorComm), Venice, Italy*, pages 115–120, 2010.
- [11] S. Santini and K. Romer. An adaptive strategy for quality-based data reduction in wireless sensor networks. In *Proceedings of the 3rd international conference on networked sensing systems (INSS 2006)*, pages 29–36, 2006.
- [12] C. Sarkar, V. S. Rao, and R. V. Prasad. No-sense: Sense with dormant sensors. In *Communications (NCC), 2014 Twentieth National Conference on*, pages 1–6. IEEE, 2014.
- [13] C. Sarkar, V. S. Rao, R. V. Prasad, S. N. Das, S. Misra, and A. Vasilakos. Vsf: An energy-efficient sensing framework using virtual sensors. *IEEE Sensors Journal*, 16(12):5046–5059, 2016.
- [14] C. Sarkar, A. Uttama Nambi SN, and R. Venkatesha Prasad. iltc: Achieving individual comfort in shared spaces. In *Proceedings of the International Conference on Embedded Wireless Systems and Networks (EWSN)*. ACM, 2016.
- [15] S. K. Soni, N. Chand, and D. P. Singh. Reducing the data transmission in WSNs using time series prediction model. In *Signal Processing, Computing and Control (ISPCC), 2012 IEEE International Conference on*, pages 1–5. IEEE, 2012.
- [16] O. Younis and S. Fahmy. Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on mobile computing*, 3(4):366–379, 2004.